Burroughs

System Software Operation Guide

B 90/B 900I B 1800/B 1900I CP 9500 Computer Management Systems (CMS)

(Relative to Release 3.05)

This manual replaces all previous editions of form 2032801

Copyright © 1985, Burroughs Machines Limited, Hounslow, England
Copyright © 1985, Burroughs Corporation, Detroit, Michigan 48232

Distribution Code SB

Priced Item
Printed in U.S.A.
March 1985

2032801

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 it used.

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

Correspondence regarding this publication should be forwarded using the Remarks form at the back of the manual, or may be addressed directly to Corporate Product Information, Burroughs Machines Limited, Cumbernauld, Glasgow, G68 0BN, Scotland.

LIST OF EFFECTIVE PAGES

Page	Issue
iii	Original
iv	Blank
v thru xxv	Original
xxvi	Blank
1-1 thru 1-4	Original
2-1 thru 2-14	Original
3-1 thru 3-67	Original
3-68	Blank
4-1 thru 4-263	Original
4-264	Blank
5-1 thru 5-19	Original
5-20	Blank
6-1 thru 6-33	Original
6-34	Blank
7-1 thru 7-86	Original
8-1 thru 8-115	Original
8-116	Blank
9-1 thru 9-115	Original
9-116	Blank
10-1 thru 10-57	Original
10-58	Blank
A-1 thru A-5	Original
A-6	Blank
B-1 thru B-31	Original
B-32	Blank
C-1 thru C-30	Original
D-1	Original
D-2	Blank

TABLE OF CONTENTS

Section	Title	Page
1	INTRODUCTION	1-1
	THE CMS CONCEPT	1-1
	BASIC	1-3
	SOFTWARE RELEASE LEVELS	1-3
	Software Patches	1-3
	SOFTWARE SUPPORT	1-3
	TO THE READER	1-4
2	BASIC CMS OPERATION	2-1
	INTRODUCTION	2-1
	PERIPHERALS	2-1
	SYSTEM AND USER DISKS	2-2
	DISK FORMAT	2-2
	Disk Initialization	2-2
	Disk Files	2-3
	Disk File Names	2-3
	Disk File Group Names	2-4
	Disk Directory	2-4
	Indexed Files	2-5
	Dual Pack Files	2-6
	MAGNETIC TAPE FILE NAMES	2-6
	PRINTER FILES	2-7
	OTHER PERIPHERALS	2-8
	PROGRAMS	2-8
	Executing Programs	2-8
	INTRINSICS	2-9
	MIX NUMBERS	2-11
	OUTPUT MESSAGES	2-11
	RAILROAD DIAGRAMS	2-12
3	CMS COMMON INTRINSICS	3-1
	INTRODUCTION	3-1
	SYNTAX RULES USED IN THIS SECTION	3-1
	Definition Of Syntactic Variables Used	3-1
	Output Message Rules	3-2
	AD (ASSIGN DEVICE)	3-3
	AD Output Messages	3-3
	AP (ADD A PACK ON SYSTEM USING PSEUDO-PACK, B 900 ONLY)	3-5
	AP Output Messages	3-5
	AX (ACCEPT A MESSAGE FOR A PROGRAM)	3-7
	AX Output Messages	3-7
	CL (CLEAR DEVICE)	3-9
	CL Output Messages	3-9
	CTL (GAIN/RELINQUISH SYSTEM SPO, NOT IMPLEMENTED ON B 1000 SYSTEMS)	3-11
	CTL Output Messages	3-11
	DB (DIRECT TO BACKUP)	3-13
	DB Output Messages	3-14

TABLE OF CONTENTS (Cont.)'

Section	Title	Page
3	DC (DATA COMMUNICATIONS OPERATOR INPUT)	3-15
	DC Output Messages	3-15
	DM (DUMP PROGRAM, ONLY IMPLEMENTED ON B 1000	3-16
	DM Output Messages	3-16
	DP (DISCONTINUE AND DUMP)	3-18
	DP Output Messages	3-18
	DS (DISCONTINUE PROGRAM)	3-20
	DS Output Messages	3-20
	DT (SET SYSTEM DATE AND TIME)	3-21
	DT Output Messages	3-22
	EX (EXECUTE A PROGRAM)	3-23
	EX Output Messages	3-24
	FD (FORM DEFINE)	3-25
	FD Output Messages	3-26
	GO (RESTART A STOPPED PROGRAM)	3-28
	GO Output Messages	3-28
	HELP (DETAILED TASK INFORMATION ON B 90 ONLY)	3-29
	HELP Output Messages	3-30
	LG (START SYSTEM MESSAGE LOGGING)	3-31
	LG Output Messages	3-31
	LS (STOP SYSTEM MESSAGE LOGGING)	3-32
	LS Output Messages	3-32
	LT (LOAD TRANSLATION TABLE, ONLY IMPLEMENTED ON B 90)	3-33
	LT Output Messages	3-33
	MX (DISPLAY CURRENT MIX)	3-35
	MX Output Messages	3-39
	ND (NEW DENSITY, ONLY IMPLEMENTED ON B 90)	3-40
	ND Output Messages	3-41
	OL (REQUEST STATUS INFORMATION OF PERIPHERALS	3-42
	OL Output Messages	3-43
	PG (PURGE TAPE)	3-44
	PG Output Messages	3-44
	PO (POWER OFF A DISK)	3-46
	PO Output Messages	3-46
	PR (ASSIGN PROGRAM PRIORITY, NOT IMPLEMENTED ON B 90 SYSTEMS)	3-49
	PR Output Messages	3-49
	RD (RESERVE DISK DRIVE)	3-50
	RD Output Messages	3-50
	RL (RELABEL DISK/PACK)	3-51
	RL Output Messages	3-51
	RY (READY DEVICE)	3-53
	RY Output Messages	3-53
	SF SUBSTITUTE FILE (NOT IMPLEMENTED ON B 90)	3-55
	SF Output Messages	3-55
	SM (MULTI-MCS DATA COMMUNICATIONS SEND MESSAGE)	3-57
	SM Output Messages:	3-57
	SN (ASSIGN A SERIAL NUMBER TO A TAPE)	3-58
	SN Output Messages	3-58

TABLE OF CONTENTS

Section	Title	Page
3	SO PRBK (SET PRINTER BACKUP ON/OFF)	3-60
	SO PRBK Output Messages	3-61
	ST (TEMPORARILY SUSPEND A RUNNING PROGRAM)	3-62
	ST Output Messages	3-62
	SV (SAVE PERIPHERAL)	3-63
	SV Output Messages	3-63
	TO PRBK (TEST PRINTER BACKUP OPTION)	3-64
	TO Output Messages	3-64
	VF (VERTICAL FORMAT ON PRINTER, NOT IMPLEMENTED ON B 1000)	3-65
	VF Output Messages	3-66
4	CMS-COMMON UTILITIES	4-1
	INTRODUCTION	4-1
	STAR FILES	4-1
	Examples	4-3
	SYS-SUPERUTL	4-3
	SYSLANGUAGE	4-4
	PSEUDO-PACKS	4-5
	LOGGING IN THE 3.05 RELEASE	4-6
	Types Of Log File	4-7
	Types Of Logging	4-7
	CYCLIC System Message Logging	4-8
	NON-CYCLIC System Message Logging	4-8
	Intrinsics And Utilities Linked With Logging	4-8
	Suggested Logging Method	4-9
	Log File Sizes	4-10
	ARCHIVING USING UPDATE FACILITY	4-11
	COMMON UTILITY OUTPUT MESSAGES	4-13
	ADD (ADD FROM LIBRARY TAPE TO DISK)	4-15
	(LD – TAPE UTILITY)	4-15
	Examples	4-15
	Output Messages	4-16
	AMEND (DISK FILE AMENDING – APPLICABLE TO B 90 ONLY)	4-17
	Starting Byte For Modification	4-19
	Output Messages	4-21
	CH (CHANGE FILE NAME)	4-22
	Examples:	4-22
	Output Messages	4-22
	CHECKADUMP (COMPARE LIBRARY TAPE WITH DISK)	4-24
	Examples	4-24
	Output Messages	4-25
	CHECK.DISK (CHECK ALL SECTORS OF A DISK)	4-27
	Output Messages	4-27
	COMPARE (COMPARE FILES)	4-31
	Examples	4-31
	Format Of Printer Output	4-32
	R Option	4-33
	Additional Capabilities	4-34

TABLE OF CONTENTS (Cont.)

Section	Title	Page
4	Non-Disk Devices	4-35
	Record And Block Sizes	4-35
	Limitations	4-36
	Output Messages	4-36
	COPY (FILE COPY)	4-39
	Copying Keyfiles	4-39
	Additional Capabilities	4-42
	Non-Disk Devices	4-42
	Unlabelled Tapes	4-42
	Record And Block Sizes	4-43
	File Size	4-44
	Single Area	4-44
	Crunching Files	4-44
	Extending Disk Files	4-45
	EBCDIC	4-45
	Free And Shared	4-45
	Recover	4-45
	Selected File Copy	4-45
	Selected Index File Copy	4-46
	Save Factor	4-46
	Copying Files To And From Remote CMS And Non-CMS Systems	4-46
	Output Messages	4-47
	CP (COMPUTE)	4-51
	Output Messages	4-51
	CREATE (CREATE DISK FILE – B 90 ONLY)	4-53
	Output Messages	4-56
	DA (DISK ANALYSIS)	4-58
	Analysis Of Standard CMS Disks	4-58
	Disk Mode	4-58
	File Mode	4-61
	Analysis Of Non-CMS Standard Disks	4-61
	General Notes	4-63
	Families	4-63
	Pseudo Disk Tags	4-63
	Abbreviations	4-64
	Output Messages	4-64
	DSKUTL	4-67
	Bootstrap Table For IN (B 90 And B 900 Only)	4-68
	DSKUTL – <COLDSTART> Option B 90 Only	4-70
	Format 1: IN (B 90 And B 900 Only)	4-70
	Operation	4-71
	Pseudo Pack Identification Table Construction	4-73
	Format 2: RF	4-74
	Operation	4-75
	Format 3: LIST	4-77
	Operation	4-77
	Format 4: COPY (B 90/B 900 Only)	4-78
	Operation	4-78

TABLE OF CONTENTS (Cont.)

Section	Title	Page
4	Format 5: BOOT (B 90/B 900 Only)	4-80
	DSKUTL Reformat And Initialize Parameters – Via Accepts	4-82
	DSKUTL Output Messages	4-82
	DUMP (DUMP FILES FROM DISK TO A LIBRARY TAPE)	4-88
	(LD – TAPE LIBRARY UTILITY)	4-88
	Output Messages	4-89
	DUMPADISK (DISK DUMP)	4-90
	Format:	4-91
	Format 1	4-91
	Format 2	4-91
	Format 3	4-92
	Formats 2 And 3	4-92
	Format 1: PRINT.DIR	4-92
	Operation	4-92
	Format 2: DUMP or UNLOAD	4-93
	Operation	4-93
	Format 3: ADD or LOAD	4-93
	Operation	4-94
	Error Recovery	4-94
	Output Messages	4-94
	DUMPANALYSE	4-100
	Obtaining the Dumpfile	4-100
	Initiating the Analysis	4-100
	Initiating Interactive Analysis	4-100
	B 90 Systems	4-101
	B 1000 Systems	4-101
	Initiating Non-Interactive Analysis	4-102
	Analysis Common to All Program Languages – Interactive/ Non-Interactive Analysis	4-104
	Program Parameter Block Information	4-104
	File Information	4-105
	Data Segment and Unformatted Dump	4-108
	COBOL Dependent Dump Analysis	4-108
	Dump-Related Dollar Options	4-108
	COBOL Segment Allocation	4-109
	COBOL S-Interpreter Work Area	4-110
	Control Stack	4-110
	COP Table (Current Operand Table)	4-110
	Namefile Table	4-111
	RPG Dependent Dump Analysis	4-111
	Dump-Related Source Constructs	4-111
	Object Program Organization	4-111
	Further Information	4-112
	MPL Dependent Dump Analysis	4-112
	MPL S-Registers	4-112
	Formatted Code and Data Stack	4-114
	Volatile Registers	4-114
	Descriptor Analysis	4-114
	Relation to MPL Compilation Listings	4-115
	Location of Program Failure Point	4-115

TABLE OF CONTENTS (Cont.)

Section	Title	Page
4	Description of Namefile for Use in Source Related Dump Analysis	4-116
	Output Messages	4-116
	ECMA.LD (Load/Dump of ECMA Tape Files – B 90/B 900 Only)	4-119
	Basic Initiation	4-119
	Format (Disk to Tape copy)	4-119
	Format (Tape to Disk copy)	4-119
	Compact Initiation	4-120
	Format (Disk to Tape copy)	4-120
	Format (Tape to Disk copy)	4-120
	Output Messages	4-121
	FILEUTL (FILE MAINTENANCE UTILITY)	4-123
	Screen Formats and Functions	4-123
	Key to Screen Formats	4-124
	Cursor Management	4-124
	Screen 1 – Select	4-127
	Screen 2 – Create	4-128
	Screen 3 – Update	4-129
	Screen 4 – Display	4-130
	Screen 5 – Select Record Sections	4-131
	Screen 6 – Record Input/Edit	4-132
	Screen 7 – Copy Records from Another File	4-134
	Screen 8 – Record Display	4-135
	Screen 9 – Specify Keys	4-136
	Screen 10 – Find Information	4-137
	Screen 11 – Help 1	4-138
	Screen 12 – Help 2	4-139
	Screen 13 – Help 3	4-140
	Screen Messages	4-140
	File Error Messages	4-141
	Remove Function Message	1-141
	Copy Messages	4-142
	Copy Error Messages	4-142
	Find Command Messages	4-142
	No Disk Space Message	4-143
	Input or Output Parity on a File Being Created or Modified	4-143
	FL (Display File Attributes on Self-Scan – B 90 Only)	4-144
	Output Format	4-144
	Output Messages	4-146
	FS (File Squash)	4-147
	Output Messages	4-148
	ICMD	4-149
	Examples	4-151
	Output Messages	4-151
	IR (Initiate Log Recall)	4-153
	Output Messages	4-153
	KA (Analyze Disk Space Assignment)	4-154
	Examples:	4-154
	Output Format	4-155
	Display of Bootstraps	4-155
	Output Messages	4-156

Section	Title	Page
4	KEY.CHECK	4-157
	Output Format	4-158
	Examples	4-158
	Output Messages	4-158
	KX (Disk Allocation Information)	4-161
	Output Messages	4-162
	LB (Look Back in Log)	4-163
	Output Messages	4-163
	LD – TAPE LIBRARY UTILITY (ADD, LOAD, DUMP, UNLOAD)	4-164
	"UPDATED" Bit Options	4-164
	Output Messages from LD – ADD/DUMP/LOAD/UNLOAD	4-165
	LF (Look Forward in Log)	4-169
	Output Messages	4-169
	LIST (File List)	4-170
	Examples:	4-170
	Additional Capabilities	4-170
	Non Disk Files	4-170
	Unlabelled Tapes	4-172
	Record and Block Sizes	4-172
	Selected File List	4-173
	Selected Indexed File List	4-173
	Output Messages	4-174
	LOAD (LD – TAPE LIBRARY UTILITY)	4-176
	Output Messages	4-177
	LOAD.VFU (Load Vertical Format Unit – B 90/B 900 Only)	4-178
	Define	4-178
	Format	4-179
	List	4-179
	Delete	4-179
	Limitations	4-179
	Output Messages	4-180
	LR (List Directory)	4-182
	Format:	4-182
	Examples:	4-182
	Output Format	4-183
	Output Messages	4-184
	MODIFY	4-186
	Format	4-186
	Examples	4-187
	Interactive Mode	4-187
	Console Operation	4-187
	Datacomm Operation	4-188
	File Attributes – Interactive/Non-interactive Analysis	4-189
	Output Messages	4-189
	PB (List Printer Backup Files)	4-194
	PB Initiating Message Parameters	4-195
	DATACOMM PB	4-197
	Examples	4-198
	PB Messages	4-200
	Files Requiring Special Forms	4-200

TABLE OF CONTENTS (Cont.)

Section	Title	Page
4	Output Messages	4-200
	PD (Print Disk Directory)	4-204
	Output Messages	4-204
	PL (Print Log Files)	4-206
	Format	4-206
	Examples	4-208
	Output Messages	4-208
	PPID (PSEUDO PACK IDENTIFIER DISPLAY – B 900 Only)	4-211
	Output Messages	4-211
	RM (Remove Files from Disk)	4-212
	Format:	4-212
	Examples:	4-212
	Example:	4-212
	Output Messages	4-213
	SCR (System Confidence Routine)	4-214
	Operating Instructions	4-215
	Prompt 1:	4-215
	Prompt 2:	4-217
	Prompt 3:	4-227
	Prompt 4:	4-228
	Prompt 5:	4-230
	Execution Details	4-232
	DCR (Disk Confidence Routine)	4-234
	CPU.IO	4-235
	MT.IO	4-237
	OTHER.IO	4-238
	SQ (Squash Disk)	4-240
	Examples:	4-241
	Output Messages	4-242
	General Guidelines	4-245
	SYCOPY (Copy Library Tapes)	4-246
	CPY	4-246
	CMP	4-246
	CCM	4-247
	MRG	4-247
	TEACH	4-248
	Output Messages	4-248
	TAPELR (List Library Tape Directory)	4-249
	Format	4-249
	Examples	4-249
	Output Format	4-249
	Output Messages	4-250
	TAPEPD (Print Name of a Library Tape)	4-251
	Output Messages	4-251
	TL (Transfer Logs)	4-252
	Output Messages	4-253
	UNLOAD (LD – TAPE LIBRARY UTILITY)	4-254
	Output Messages	4-255
	UPDATE (Disk File Update – B 90 Only)	4-256
	Output Messages:	4-259

Section	Title	Page
4	WL (What Log File)	4-260
	Format:	4-260
	Output Messages	4-261
	XD (Delete Bad Disk Sectors)	4-262
	Warnings	4-262
	Output Messages	4-262
5	THE SORT/MERGE	5-1
	INTRODUCTION	5-1
	GENERAL FEATURES	5-1
	INVOKING THE SORT	5-2
	THE SORT LANGUAGE	5-3
	The File Statement	5-3
	The Key Statement	5-4
	The User-Option Statement	5-5
	INTERACTIVE SORT	5-8
	FUNCTIONAL DESCRIPTION	5-10
	Regular Record Sort	5-10
	Inplace Record Sort	5-11
	Keyfile Creation	5-11
	Tagfile Creation	5-11
	Merge	5-13
	Details Of Sort Keys	5-13
	ALPHA (Or UA)	5-13
	NUMERIC (Or UN)	5-13
	SA	5-14
	SN	5-16
	SSA	5-16
	Deleted Records	5-16
	Output Messages	5-16
	0-99	5-17
	170-200	5-17
6	COMPILATION FACILITIES	6-1
	INTRODUCTION	6-1
	DETAILS OF "CO.MASTER" FOR RELEASE 3.5	6-1
	The Startup Message	6-1
	TO INITIATE A SINGLE COMPILATION	6-4
	General Parameters	6-5
	<disk-id-1>	6-5
	<disk-id-2>	6-5
	<program-id>	6-5
	<disk-id-3>	6-5
	<compiler-name>	6-5
	<compiler-option>	6-6
	<print-option>	6-6
	<message-statement>	6-6
	<file-statement>	6-7
	File Modification Parameters	6-7
	PATCHFILE	6-7
	SOURCEIN	6-8

TABLE OF CONTENTS

Section	Title	Page
6	SOURCEOUT	6-8
	WORKFILE	6-8
	PRINTOUT	6-9
	CONTROLFILE	6-10
	ICFILE	6-10
	MASTERFILE	6-10
	Errors	6-11
	DP.NAMES (COBOL And RPG Only)	6-11
	Description Of The Attributes	6-11
	CO VERSION 2	6-16
	To Interrogate The Status Of Compilations	6-16
	TO RESTART AN ABORTED COMPILATION	6-19
	TO CLEAR AN ABORTED COMPILATION	6-21
	COMPILER DOLLAR OPTIONS FOR CO	6-23
	RPG Compiler	6-23
	MPLII Compiler	6-23
	COBOL* Compiler	6-23
	Stand-Alone OPTLIST	6-23
	COBOLXREF*	6-24
	Dollar Option Precedence	6-24
	CO ERROR MESSAGES	6-25
	CO OTHER MESSAGES	6-31
	ZIP FAILURES	6-32
	RESERVED WORDS	6-33
7	CMS SYSTEM SOFTWARE OUTPUT MESSAGES	7-1
	INTRODUCTION	7-1
	SYNTAX RULES USED IN THIS SECTION	7-1
	Definition Of Syntactic Variables Used	7-1
	SUBSECTION A	7-3
	OUTPUT MESSAGES FOR B 90/B 900/CP 9500 SYSTEMS	7-3
	Events 1-5	7-3
	Software Information	7-3
	Events 6, 22-27, 29-37, 40-41, 69, 79-81, 84, 87, 89-90	7-5
	Invalid Requests On Class A Or B Communicate To MCP (1)	7-5
	Events 10-19, 28, 39, 42, 82, 85, 88	7-9
	Software Suspensions	7-9
	Events 20-21	7-12
	Invalid Requests On Class A Or B Communicate To MCP (2)	7-12
	Events 43-48	7-13
	Device Errors	7-13
	Events 49-68, 76-78	7-14
	Loader Detected Failures	7-14
	Events 70-75	7-17
	Run Structure Problems	7-17
	Event 83	7-18
	Unimplemented Communicates	7-18
	Event 91	7-19
	Message Dictionary Section	7-19

TABLE OF CONTENTS (Cont.)

Section	Title	Page
7	Events 100-169	7-20
	Interpreter Detected Errors	7-20
	Events 100-105 MPL/COBOL Interpreter Detected Errors	7-20
	Events 110-132 MPL Interpreter Detected Errors	7-22
	Events 140-169 COBOL Interpreter Detected Errors	7-24
	Events 170-199	7-26
	Sort/Merge Information	7-26
	Events 200-349	7-29
	Data Communications Errors	7-29
	Events 200-229, 231-237	7-30
	Datacomm Handler Detected Errors (For B 90/B 900/CP 9500)	7-30
	Events 230, 245-254	7-36
	Datacomm Loader Detected Errors	7-36
	B 80/B 90 Dependent Datacomm Messages	7-38
	Event 255	7-38
	Events 256-263	7-39
	Event 269	7-41
	B 900/CP 9500 Dependent Datacomm Messages	7-42
	Events 300-305	7-42
	Events 350-499	7-43
	Implementation Dependent System Messages	7-43
	Events 350-369 – Common Messages	7-43
	Events 370-399 – B 90 Dependent Messages	7-45
	Events 430-469, 630-634 – B 900 System Dependent Messages	7-46
	Events 564-567 – B 90/B 900 SDI Disk Messages	7-48
	SUBSECTION B	7-50
	OUTPUT MESSAGES FOR B 1800/B 1900 SYSTEMS	7-50
	Events 1-6	7-50
	Software Information	7-50
	Events 7-9	7-51
	Device Messages	7-51
	Events 10-19, 27-29, 39, 42, 69, 81-82, 85 And 88	7-52
	Software Suspensions	7-52
	Events 20-26, 30-38, 40, 41, 79, 80, 84, 87, 89, 90	7-55
	Invalid Requests On Class A Or B Communicate To MCP	7-55
	Events 43-49	7-59
	Device Errors	7-59
	Events 50-68, 76	7-60
	Loader Detected Failures	7-60
	Events 70-75	7-63
	Run Structure Errors	7-63
	Events 100-169	7-64
	Interpreter Detected Errors	7-64
	MPL Interpreter Detected Errors	7-65
	COBOL/RPG Interpreter Detected Errors	7-67
	Events 170-199	7-70
	Sort/Merge Information	7-70
	Events 200-349	7-74
	Data Communications Errors On B 1000 Systems	7-74

TABLE OF CONTENTS (Cont.)

Section	Title	Page
7	Events 200-247	7-75
	Events 249-254	7-81
	Events 272-274	7-82
	Events 282-284	7-83
	Events 285-286	7-84
	Events 350-499	7-85
	Implementation Dependent System Messages	7-85
	Events 350-369	7-85
	Events 470-499	7-85
8	B 90 DEPENDENT SYSTEM SOFTWARE	8-1
	INTRODUCTION	8-1
	POWER ON	8-1
	CMS BOOTSTRAP MODE	8-5
	Forcing System Initialization	8-5
	COLDSTART (B 90 ONLY)	8-6
	Restrictions On Use	8-6
	Creation Of A Coldstart Tape	8-6
	Creation Of A Coldstart Disk	8-7
	Outline Of COLDSTART Utility	8-7
	Initialization Parameters For 211 Fixed Disk	8-9
	Error Messages	8-9
	STAND-ALONE UTILITIES (SAU)	8-12
	Loading Stand-Alone Utilities	8-12
	SAU From A Console	8-12
	SAU From A Terminal	8-12
	SAU Header Message	8-12
	SAU Functions	8-13
	Terminal Configuration For SAU	8-13
	Common SAU Output Messages	8-14
	Disk I/O Errors During SAU	8-15
	A Note On Dual Pack Files	8-15
	CH (CHANGE DISK FILE NAME)	8-17
	Output Messages	8-17
	CI (INITIALIZE A DISK WITH COLDSTART BOOTSTRAP)	8-19
	Output Messages	8-19
	CLEAN (CLEAN BSM DRIVE READ/WRITE HEADS)	8-21
	Output Messages	8-21
	COMPARE (COMPARE TWO DISK FILES)	8-22
	Output Messages	8-22
	COPY (COPY FILES DISK TO DISK)	8-24
	Output Messages	8-24
	Dual Pack Files	8-26
	DISCOPY (DUPLICATE A BSMII DISK)	8-27
	Output Messages	8-27
	FE (INITIALIZE MTR DISK)	8-28
	Output Messages	8-28
	IN (INITIALIZE A DISK WITH WARMSTART BOOTSTRAP)	8-30
	Output Messages	8-30

TABLE OF CONTENTS (Cont.)

Section	Title	Page
8	LD (LOAD FILES FROM CASSETTE TO DISK)	8-32
	Output Messages	8-32
	LS (LIST FILE SIZES)	8-34
	Output Messages	8-34
	OL (PRINT STATUS OF DRIVES)	8-35
	Output Messages	8-35
	PDX (PRINT DISK DIRECTORIES)	8-36
	Output Messages	8-36
	PO (POWER OFF)	8-37
	RL (RELABEL A DISK)	8-38
	Output Messages	8-38
	Warning	8-38
	RM (REMOVE DISK FILES)	8-39
	Output Messages	8-39
	Dual Pack Files	8-39
	WS (WARM START)	8-41
	Operating Procedure	8-41
	LOADING THE MASTER CONTROL PROGRAM (MCP)	8-42
	Console And SELF-SCAN SPO Device	8-42
	Terminal SPO Device	8-43
	BASIC OPERATION UNDER MCP CONTROL	8-44
	D-Lights (Bank 5)	8-44
	MCP States	8-44
	Automatic Volume Recognition (AVR)	8-45
	Console Keyboard Under MCP Control	8-45
	Interrupting The MCP	8-46
	POWER OFF	8-47
	ROM SCANNING ALGORITHM FOR ROM LEVEL 4	8-48
	ROM SCANNING ALGORITHM FOR LEVEL 5 ROMS	8-49
	Loading Coldstart Bootstrap – PK1	8-49
	Loading Warmstart Bootstrap – PK2	8-49
	ROM SCANNING ALGORITHM FOR LEVEL 6 ROMS	8-51
	Data Communications Requirements	8-51
	Keyboard Input And Output	8-52
	ROM Start Up Cycle	8-52
	COLDSTART BOOTSTRAP ALGORITHM	8-54
	WARMSTART SCANNING ALGORITHM	8-55
	MEMORY DUMP TO MAGNETIC TAPE	8-56
	MEMORY DUMP TO DISK	8-57
	SYSTEM LOAD ERRORS	8-59
	Diagnosis Of Disk Errors At System Load Time	8-60
	Caelus Device Primary Status	8-60
	Caelus Disk Secondary Status	8-61
	SDI Host Controller Status	8-61
	SDI Device Status	8-61
	Magnetic Tape Device	8-62
	ERRORS UNDER MCP CONTROL	8-63
	ERRORS DURING WARMSTART OF A CONSOLELESS SYSTEM	8-67
	B 90 DEPENDENT UTILITIES	8-69

TABLE OF CONTENTS (Cont.)

Section	Title	Page
8	BUILDTRANS (BUILD TRANSLATION TABLE)	8-70
	Limitations	8-70
	Operating Instructions	8-70
	Mode 0	8-71
	Mode 1	8-71
	Mode 2 (Not Implemented On B 90)	8-71
	Shift Codes	8-71
	Workfile Creation	8-71
	Initiation And Working Mode	8-72
	Functions	8-72
	GET	8-72
	MAKE	8-72
	INCLUDE	8-72
	LIST	8-73
	BUILD	8-73
	EDIT	8-74
	DELETE	8-74
	LET	8-74
	REMOVE	8-75
	RECOVER	8-75
	BYE	8-75
	BUILDTRANS Message Format	8-75
	SYSTRANS Format	8-78
	CONFIGURER (Configure B 90 System)	8-80
	Operating Instructions	8-81
	Output Messages	8-85
	GEN.DUMPFL (Create Empty B 90 Memory Dump File)	8-87
	Output Messages	8-88
	INSTALL (Disk File Copy from System to Fixed Disk)	8-89
	Output Messages	8-90
	PATCHMAKER (Patching B 90 Machine-Code Object Program Files)	8-92
	Patching Procedure	8-93
	Sample Flash	8-93
	Operating Instructions	8-95
	Patchmaker Error Messages	8-96
	PMB90 (Analyze B 90 Memory Dumps)	8-99
	Starting the Utility	8-99
	AX	8-99
	GET < filename >	8-99
	ON < diskname >	8-100
	ON TAPE	8-100
	MERGE	8-100
	Using the Utility	8-100
	Syntax of Input Specifications	8-101
	Initiating PMB90	8-109
	Using the MERGE Option	8-109
	Early Merge Termination	8-110
	B 90 DEPENDENT INTRINSICS	8-112
	GT (General Trace)	8-113

Section	Title	Page
8	Format of Diagnostic Printout	8-115
	ENHANCEMENTS TO LOGGING	8-116
	Variations to PL and IR caused by MTS	8-116
9	B 900/CP 9500 DEPENDENT SYSTEM SOFTWARE	9-1
	INTRODUCTION	9-1
	CONTROL PANEL	9-1
	Data Communications Transmit And Receive Indicators	9-1
	Data Communications Group Select Switch	9-1
	OPERATOR ATTENDED/UNATTENDED	9-1
	SYSTEM START-UP	9-3
	Physical Power-On	9-3
	WARMSTART	9-4
	System With SPO	9-4
	Warmstart Variations	9-5
	Coping With Problems	9-6
	COLDSTART	9-6
	COLDSTART Initiation	9-6
	Procedure	9-6
	COLDSTART Attended Mode – SPO Present	9-7
	BC (Binary Copy)	9-8
	Procedure	9-8
	SPO – Attended Operation	9-8
	?DS	9-11
	FE	9-12
	IN (Initializing Disks)	9-13
	Procedure	9-13
	Initializing Fixed Disks	9-14
	LD (Load)	9-15
	Procedure	9-15
	OL (On Line)	9-16
	RF (Reformat)	9-17
	Procedure	9-17
	RL (Relabel)	9-19
	Procedure	9-19
	RP (Replace)	9-20
	WS (Warmstart)	9-21
	COLDSTART Attended Mode – No SPO Present	9-22
	COLDSTART – Operator Unattended Mode	9-25
	INITIALIZE, LOAD, WARMSTART	9-26
	REPLACE, WARMSTART	9-27
	POWER OFF	9-28
	TAKING SYSTEM DUMPS	9-29
	Operating System Dumps Created Automatically	9-29
	Using GT MD	9-30
	Taking A ROM-Created Dump	9-31
	Multiple Disk Operation	9-31
	Analyzing Dumps	9-32
	TAKING PROGRAM DUMPS	9-33
	FUNCTIONAL DESCRIPTION OF SYSTEM STARTUP	9-34

TABLE OF CONTENTS (Cont.)

Section	Title	Page
9	B 900/Cp 9500 Bootstraps	9-35
	WARMSTART FUNCTIONAL DESCRIPTION	9-40
	Default Configuration	9-40
	Buffer Memory Defaults	9-40
	Task Processor Defaults	9-41
	Data Comm Processor Defaults	9-41
	COLDSTART FUNCTIONAL DESCRIPTION	9-46
	FE (Initialize MTR Disk)	9-46
	Increasing An Existing Fixed Disk Assemblage	9-47
	COLDSTART Disk Selection	9-48
	COLDSTART Default Assignments	9-48
	COLDSTART Messages	9-49
	COLDSTART Error Displays	9-49
	DUMPS FUNCTIONAL DESCRIPTION	9-55
	Read Only Memory (ROM) Dump Routine	9-55
	Disk Format	9-55
	ROM Dump Messages	9-56
	ROM Dump Display Reference Table	9-57
	SPECIAL CONSIDERATIONS	9-59
	Pseudo-Pack Concepts	9-59
	Data Communications	9-60
	Enhanced Terminal SPO	9-60
	Tape Concepts	9-63
	Tape Commands	9-63
	Disk Pack Concepts	9-63
	Disk Pack Used To Store System Software	9-64
	Testing Recording Surface Of Disk Devices	9-64
	Backup Operating System And Disk Processors	9-66
	UTILITIES AND INTRINSICS	9-67
	ANALYZER (User Program Dump Analyzer)	9-67
	Output Messages	9-67
	AP (Assign Pseudo-Pack)*	9-69
	Output Messages	9-70
	AT (At Host)	9-71
	Output Messages	9-71
	BINARY COPY (Duplicate Disk Media)	9-72
	Special Considerations	9-73
	Output Messages	9-74
	CA (Candidate)	9-76
	Output Messages	9-76
	CONFIGURER (System Configuration Utility)	9-77
	PRINT	9-78
	Initiation	9-78
	MAKE	9-80
	MODIFY	9-80
	DISPLAY	9-85
	DUMPFILE.UTL (System Dump File Segmenter)	9-86
	File Partitioning	9-86
	File Reconstruction	9-87
	Output Messages	9-87

Section	Title	Page
9	EX (Execute)	9-89
	Output Message	9-89
	FPP (Field Patch Program)	9-90
	Output Messages	9-92
	Valid Mnemonics for Patches	9-93
	GT CF (Get System Configuration)	9-94
	GT MD (Get Full Memory Dump)	9-95
	GT MX (Get Mix)	9-96
	Display Mix with Task Processor Assignment	9-96
	HN (Hostname)	9-97
	Output Message	9-97
	NPC900 and NPC900P	9-98
	Syntax	9-98
	Error Messages	9-99
	NW (Network)	9-101
	Output Messages	9-101
	PO (Power Off)	9-102
	Output Messages	9-102
	Option 1	9-102
	Option 2	9-102
	RL (Relabel)	9-103
	Output Messages	9-103
	RLD (Release Level Display)	9-105
	Output Message	9-106
	ROMANALYZER (ROM Dump File Analyzer)	9-107
	ROMANALYZER Output	9-107
	Page Descriptor Format	9-108
	Format of Hex Dump	9-108
	Output Messages	9-108
	ROMCONVERT (ROM Dump File Cpnverter)	9-110
	Output Messages	9-111
	SM (Multi-MCS Data Comm)	9-112
	SYSANALYZER (System Dump Analyzer)	9-113
	SYSANALYZER Error Messages	9-115
10	B 1800/B 1900 DEPENDENT SYSTEM SOFTWARE	10-1
	CMS INTRINSICS	10-1
	CMS UTILITIES	10-2
	CONFIGURER	10-2
	Syntax	10-2
	Defaults	10-6
	Update Messages	10-6
	Error Messages	10-7
	Warning Messages	10-8
	DC.INFO	10-9
	Syntax	10-9
	Error Messages	10-9
	Completion Message	10-9
	DCP.ANALYZER	10-10
	How to Take a DCP Dump	10-10

Section	Title	Page
10	How to Analyze a DCP Dump	10-10
	Syntax	10-10
	Commands and Formatters	10-11
	DISKDUMP	10-16
	DSKDSK	10-16
	DSKMTP	10-17
	MTPDSK	10-17
	AND COMPARE	10-17
	Unlabelled Open	10-17
	Program Initiation	10-17
	BOJ Messages	10-18
	Error Handling and Error Messages	10-18
	Bad Sector Count in Label	10-18
	Comparison Errors	10-18
	Recoverable Input Errors	10-18
	Bad Sectors Errors	10-19
	Fatal Errors	10-19
	Program Warnings	10-19
	End Of Job Messages	10-20
	Warnings	10-20
	Approximate Duration of the Process	10-20
	LT (Load Line Printer Train)	10-21
	Output Messages	10-22
	Error Messages	10-22
	Teach Option	10-22
	MEM.ANALYZER	10-23
	NPC1.B1000	10-27
	Introduction	10-27
	Use of the NPC1.B1000	10-27
	Syntax	10-27
	NPC34.B1000	10-28
	Introduction	10-28
	Use of the NPC34.B1000	10-28
	Syntax	10-28
	PATCH.MAKER	10-29
	Error Messages Displayed	10-29
	Example	10-30
	Operating Instructions for Patching the MCP File	10-30
	Two System Packs are available (SYS1 and SYS2)	10-31
	One System Pack Only is Available	10-31
	Example	10-31
	Operating Instructions for Patching the Interpreters or NPC34.B1000.	10-32
	Example:	10-32
	RETRIEVE	10-33
	Syntax	10-33
	Messages	10-33
	SDA (Only Relevent to B 1990 Systems)	10-35
	Syntax	10-35
	Completion Message	10-35
	STAND-ALONE UTILITIES	10-36

Section	Title	Page
10	Creation of Cassettes	10-36
	Syntax	10-36
	Operating Instructions	10-36
	Initiation of the Stand-Alone Utilities	10-36
	For All Systems Except B 1825/30, B 1990	10-36
	B 1825/30 System	10-37
	B 1990 System	10-37
	All Systems	10-37
	Cassette Error	10-37
	COLD.START	10-38
	Operation	10-38
	Disk Pack and Cartridge Units	10-38
	Magnetic Tape Units	10-38
	Error Messages	10-40
	CLEAR.START	10-42
	Function	10-42
	Selection of the MCP File	10-42
	Channel Selection	10-42
	Channel Identification	10-42
	DISK INITIALIZER (Pack.Init)	10-44
	General Information	10-44
	Operating Instructions	10-44
	Cassette Loading	10-44
	Card Format	10-44
	SPO Input	10-45
	Time Needed	10-46
	Error Messages	10-46
	Available Table Size – Example	10-46
	Maximum Number of Removed Sectors	10-47
	MEMORY.DUMP	10-48
	Operating Instructions	10-48
	Loading Instructions	10-48
	How to Trace a System Loop	10-48
	List of Possible Halts (in L Register)	10-49
	Dump Failures	10-50
	SYSTEM HALTS DOCUMENTATION	10-51
	System Halts	10-51
	CLEARSTART Halts	10-51
	MCP Halts	10-52
	System Dependent Fetch Values	10-54
	Physical I/O	10-54
	Logical I/O	10-56
	Data Communicaton Errors	10-56
APPENDIX		
A	Glossary of Technical Terms	A-1
B	Complete Railroad Diagrams	B-1
C	Example of Printed Utility Output	C-1
D	Related Documentation	D-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
1-1	CMS Portability	1-1
2-1	Physical Disk Structure	2-3
2-2	Disk Directory Structure	2-5
2-3	Indexed Files	2-6
2-4	Dual Pack Files	2-7
2-5	Sample SPO List	2-13
2-6	Railroad Diagram Sample 1	2-13
2-7	Railroad Diagram Sample 2	2-13
4-1	Fixed Disk Directory Structure	4-5
4-2	Railroad Chart for COMPARE Utility	4-34
4-3	Railroad Chart for Copy Utility (Sheet 1 of 2)	4-40
4-3	Railroad Chart for Copy Utility (Sheet 2 of 2)	4-41
4-4	Routing Diagram for FILEUTL	4-125
4-5	Railroad Chart for LIST Utility	4-171
4-6	OTHER.IO Sample Printer Output	4-239
5-1	Regular Record Sort	5-10
5-2	Keyfile Creation	5-12
5-3	Tagfile Creation	5-12
5-4	File Merge	5-14
5-5	Multiple Key Sort	5-15
6-1	Operation of CO Utility	6-3
8-1	B 90 Warmstart ROM Level 4	8-3
8-2	B 90 Coldstart and Warmstart ROM Level 5	8-4
8-3	Outline of COLDSTART Utility	8-8
8-4	D-Light Display Interpretation.	8-44
9-1	Control Panel	9-2
9-2	SPO	9-3
9-3	Block Diagram of Operating Steps in Daily Running of B 900/CP 9500	9-4
9-4	COLDSTART Switch Settings	9-26
9-5	Sample SYSCONFIG File	9-79

LIST OF TABLES

Table	Title	Page
2-1	Command Mnemonics	2-10
5-1	Sign Convention for Signed 8-bit Numeric Fields	5-15
5-2	Sign Convention for Signed 4-Bit Numeric Fields	5-16
5-3	Sign Convention for Separate Sign Character with 8-bit Alphanumeric Fields	5-16
6-1	Valid CMS Devices	6-12
6-2	Default Settings for Parameters Modifiable Through CO	6-14
6-3	Error Messages From CO Error Messages	6-25
6-4	CO Reserved Words	6-33
8-1	Consoleless System Error Codes	8-67
9-1	COLDSTART Functions	9-7
9-2	Error Messages of Disk Selection Phase of Binary Copy	9-9
9-3	Error Messages Displayed During Copy Phase of Binary Copy	9-9
9-4	Error Messages of Verification Phase of Binary Copy	9-10
9-5	Keypad Attended Operation – Binary Copy	9-22
9-6	Keypad Attended Operation – FE Initialize	9-22
9-7	Keypad Attended Operation – INITIALIZE	9-23
9-8	Keypad Attended Operation – LOAD	9-23
9-9	Keypad Attended Operation – REFORMAT	9-24
9-10	Keypad Attended Operation – REPLACE	9-24
9-11	Keypad Attended Operation – WARMSTART	9-25
9-12	Automatic Dump Feature Messages	9-30
9-13	Bank A Hex Displays for Startup Programs	9-34
9-14	TRACK ZERO BOOT Error Displays	9-36
9-15	SYSBOOTSTRAP Error Displays	9-37
9-16	General Description of WARMSTART Messages	9-42
9-17	Disk Attribute Default Values for IN, FE and RF Functions	9-48
9-18	COLDSTART Error Displays	9-49
9-19	COLDSTART Functions	9-53
9-20	Bank D Error Codes	9-56
9-21	ROM Dump Display Reference Table	9-57
9-22	Bank E Display for ROM Dump	9-58
9-23	SCL Commands and Issuing SPO	9-62
9-24	Pack Initialization Procedure	9-64
9-25	Structure of SYSCONFIG File	9-77
9-26	Default Values	9-82
9-27	Translation Table Names	9-83

SECTION 1
INTRODUCTION

THE CMS CONCEPT

CMS (Computer Management System) software is a powerful set of software items designed to operate on a number of different hardware products.

To the user of an individual hardware product running CMS software, there is a well-defined operator interface and set of programming languages. The importance of CMS is that the same user may use a different hardware product running CMS software, and with the same languages. This portability eliminates major operator retraining between different CMS products. It also allows freedom of interchange of programs between hardware products, limited only by availability of hardware features. For example, a program may be developed and compiled on one system, and run on another. Also, because the compilers are also programs, there is portability of compilers between hardware systems as well. Data files are similarly transferable from one system to another. This portability is achieved by building on the "soft machine" concept. Refer to figure 1-1.

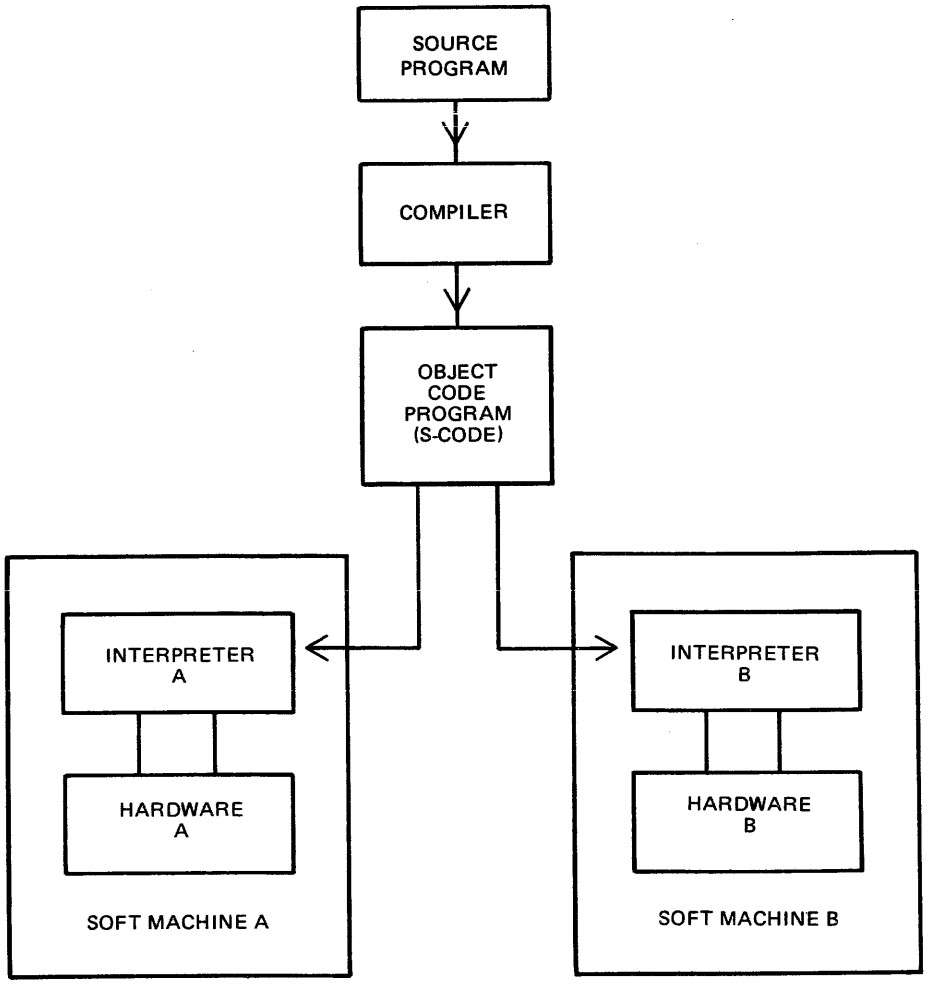


Figure 1-1. CMS Portability

The programmer writes a program in a high-level language. The CMS programming languages are:

COBOL
RPG (including RPGII)
MPL (CMS Message Processing Language)
NDL (Network Definition Language)

This program is written in "source code". This is then input to one of the CMS compilers which converts it to "object-code" or "S-code". This is the executable program. The "S-code" is similar in design to the "machine code" of earlier generations of computer.

In earlier generations of computer this "S-code" would be executed by hard-wired instructions. With the advent of fast micro-processor computers, however, it is possible to build a set of micro-instructions which interprets each "S-code" and executes it. The set of micro-instructions is therefore called an "interpreter". The combination of interpreter and micro-processor hardware is sometimes termed a "soft machine".

As the "S-code" is independent of any particular hardware, it is possible (and has been achieved in CMS) to build several soft machines which will execute an "object program" in a similar manner. Hence the CMS object programs are portable across the different CMS machines.

These machines include:

B 90
B 900
B 1800
B 1900

There are different CMS interpreters on each system. For example, on the B 90 the interpreters are:

BILINTERPX
COBOLINTX
NDL.INTERPX

BILINTERPX is used to execute programs written in MPL and in BIL (an implementation language used for compiler-writing which is so similar to MPL that they share the same S-code format). COBOLINTX is used to execute programs written in COBOL and RPG (these two languages share the same S-code format). NDL.INTERPX is used to interpret data communication controller programs written in NDL.

Certain common features needed in all programs (such as the handling of peripheral devices) have been collected together into a Master Control Program (MCP). The MCP is a micro-code program and is therefore specifically written for each hardware product. Thus there is a B 90 MCP, B 900 MCP, B 1800 MCP and a B 1900 MCP. The MCP also controls the operator interface (which is standard across the CMS range) and maintains overall control of the system, providing complete resource management including multi-programming, I/O device handling and memory management.

CMS software also provides a number of utility programs. As these are written in MPL, they also are portable across the CMS range, limited only by hardware feature availability.

To cover the complete features of each CMS product line, certain aspects of the software are written for a specific product. These additional features include important operational characteristics, and are described in sections 8 through 10. Sections 2 through 7 of this manual cover items which are applicable to any CMS product.

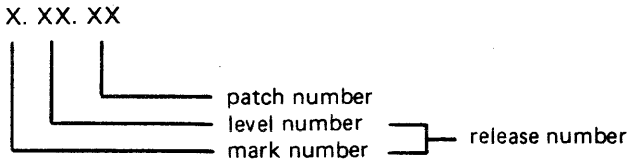
BASIC

In addition to the languages mentioned above, CMS also supports BASIC.

BASIC is a directly interpretive language, so there is no object code which has to be generated by a compiler. BASIC can be used from any terminal on a CMS system, including the ODT, provided that the system has been configured to allow this. For more information refer to the CMS BASIC Language Reference Manual, form number 1155819.

SOFTWARE RELEASE LEVELS

Each item on a CMS software release is identified by a three-part number, as follows:



The mark and level numbers constitute the release number. For example, the COBOL compiler 3.05.08 is the COBOL compiler included in the 3.05 release of system software, with patch number 08.

Software items from different releases should not be used together. For example, an interpreter from release 3.05 should not be used with an MCP from release 3.04.

This book describes system software relative to the 3.05 release.

Software Patches

Within a particular release, patches to individual items may be issued. For example, an MCP identified by 3.05.02 contains certain improvements over an MCP identified by 3.05.01. A patch always increases the patch number. It is always advisable to use the highest patch versions within any one release. All system software items within a given release (mark and level numbers) may be used together, regardless of the patch number, unless explicitly stated otherwise at the time of release of the item.

Certain items may be patched by the user. The details are machine-dependent and are described in the relevant section (8 through 10).

SOFTWARE SUPPORT

Throughout this book, suggestions are made for corrective action where possible, following a particular output message or symptom of failure. Sometimes the phrase "request technical assistance" has been used. This should be interpreted as a recommendation to contact your immediately higher support level if you are not sure of what to do or do not feel justified in attempting further action without competent advice.

All problems with the system should be recorded. This is for two purposes: to report the problem; and to avoid similar problems in the future. The report should contain the date and time and list the systems. As a minimum, it is recommended that the SPO hard-copy printout or SPO log is kept for future reference.

TO THE READER

This book is written as reference material. It is a guide to be consulted during operation of any CMS machine.

This book explains how to start and stop the system software. As this is normally hardware-dependent, the relevant section (8 through 10) should be consulted.

Once the system software has started (that is, the system is under MCP control), the operator may interface with the MCP via the SPO (Supervisory Printout) device in order to execute programs. The type of device may vary with the hardware product, but input and output messages are standardized.

Section 2 of this book explains some general terms which should be understood in order to make full use of the CMS features. It explains how to cause programs to be executed. This section also explains how to read the diagrams used throughout the book to describe the format of input messages and other details.

Details of input messages are given, in alphabetical order, in sections 3 and 4. The items in section 4 are utility programs which are executed in the same manner as other programs. The items in section 3 are embedded features in the MCP. Refer to section 2 for a fuller explanation.

Sections 5 and 6 describe the sort/merge feature and the compilation feature respectively, and will be of special interest to programmers. Section 5 includes a functional description of the sort/merge feature.

Section 7 lists the messages which may be output to the SPO by the system software during execution of the system. As each message is identified on the SPO by a number, reference to this book can be made by this number.

For other items such as hardware and system software failures, refer to the particular hardware section section (8 through 10) for details.

SECTION 2
BASIC CMS OPERATION

INTRODUCTION

All CMS operation has two basic principles: it is disk-based; and operator communication is with the MCP by a SPO device. Other peripherals may be present, depending on the configuration. This section introduces some basic principles which should be understood by all CMS operators. The material in this section is common to all CMS products. Other details that are machine-dependent are given in the relevant section.

PERIPHERALS

Each peripheral is referenced by a three-character abbreviation, where the first two characters give the type of peripheral and the third character refers to the particular peripheral by the letter A, B, and so on. For example, LP is the abbreviation for a line printer, so the first line printer is referred to as LPA, and the second is LPB.

The peripheral types are listed here:

AC – console with any output device
AM – any multi-function card unit
AP – any (serial or line) printer
AR – any card reader
AT – any magnetic tape
CP – any card punch
CT – cassette tape
DC – data communications controller
DF – fixed disk
DI – industry-compatible mini-disk (ICMD)
DK – disk cartridge
DM – Burroughs super mini disk (BSMD and BSMDII and 5.25 inch floppy disk)
DP – disk pack
KB – keyboard
LP – line printer
MT – magnetic tape (reel)
M8 – 80-column multi-function card unit
M9 – 96-column multi-function card unit
PC – console with serial printer
P8 – 80-column card punch
P9 – 96-column card punch
R8 – 80-column card reader
R9 – 96-column card reader
RS – Reader Sorter
RT – Real Time Clock
SC – (console with) SELF-SCAN® device
SD – screen display
SP – serial printer (on console)

If the configuration contains more than one device of the same type, the designation (A, B and so on) depends on the location of the peripheral controller in the hardware.

SELF-SCAN is a registered trademark of Burroughs Corporation.

The three-character references are used in all operator communication with the MCP (refer to section 3).

SYSTEM AND USER DISKS

The MCP resides on a disk unit. At warmstart time (when the system is started up and the MCP begins to function) the MCP notes the disk containing the executing MCP code. This is called the "system disk".

During operation there is only one system disk. Other disks may contain a copy of the MCP code, but only the disk from which the MCP is running is the system disk.

All other disks on the system during machine operation are called "user disks".

There is one restriction on the portability of system disks between different CMS products. A system disk may not be taken to a different CMS product and used there as a system disk. It may, however, be used on the second system as a user disk. It may also be used on the first system as a user disk. User disks may always be interchanged between different systems.

DISK FORMAT

A disk consists of one or more platters, one or both surfaces of which may be used to record data. The recording area of disks is divided into the following physical items:

Track:

> An area of one surface of a disk which is at the same distance from the center of the disk. The entire track can be accessed without moving the position of the read/write head.

Sector:

> The basic unit of disk address, size 180 bytes on all Burroughs disks, and 128 bytes on ICMD. A physical read or write uses a complete sector. There are several additional bytes in each sector, used only by the hardware and not accessible to user programs. The sector is also called a "segment".

Cylinder:

> If there is more than one surface, each track at the same distance from the center makes a cylinder. The entire cylinder may be accessed without altering the position of the read/write heads.

Figure 2-1 illustrates these terms.

Disk Initialization

Each disk must be initialized before use on a CMS machine. Initialization creates correct sector addresses throughout the disk recording surface, then writes certain data in the low-address part of the disk. The first sector is numbered sector zero, and the first track is numbered track zero. A disk with a bad track 0 (zero) cannot be initialized. The method of initializing the disk is machine-dependent (refer to the appropriate section).

Sector zero contains the disk label. This includes the name of the disk, or "disk-id". Every disk has a disk-name. This disk-name can be from one to seven characters, using the set A to Z, 0 to 9 and the dot (".") and hyphen ("-").

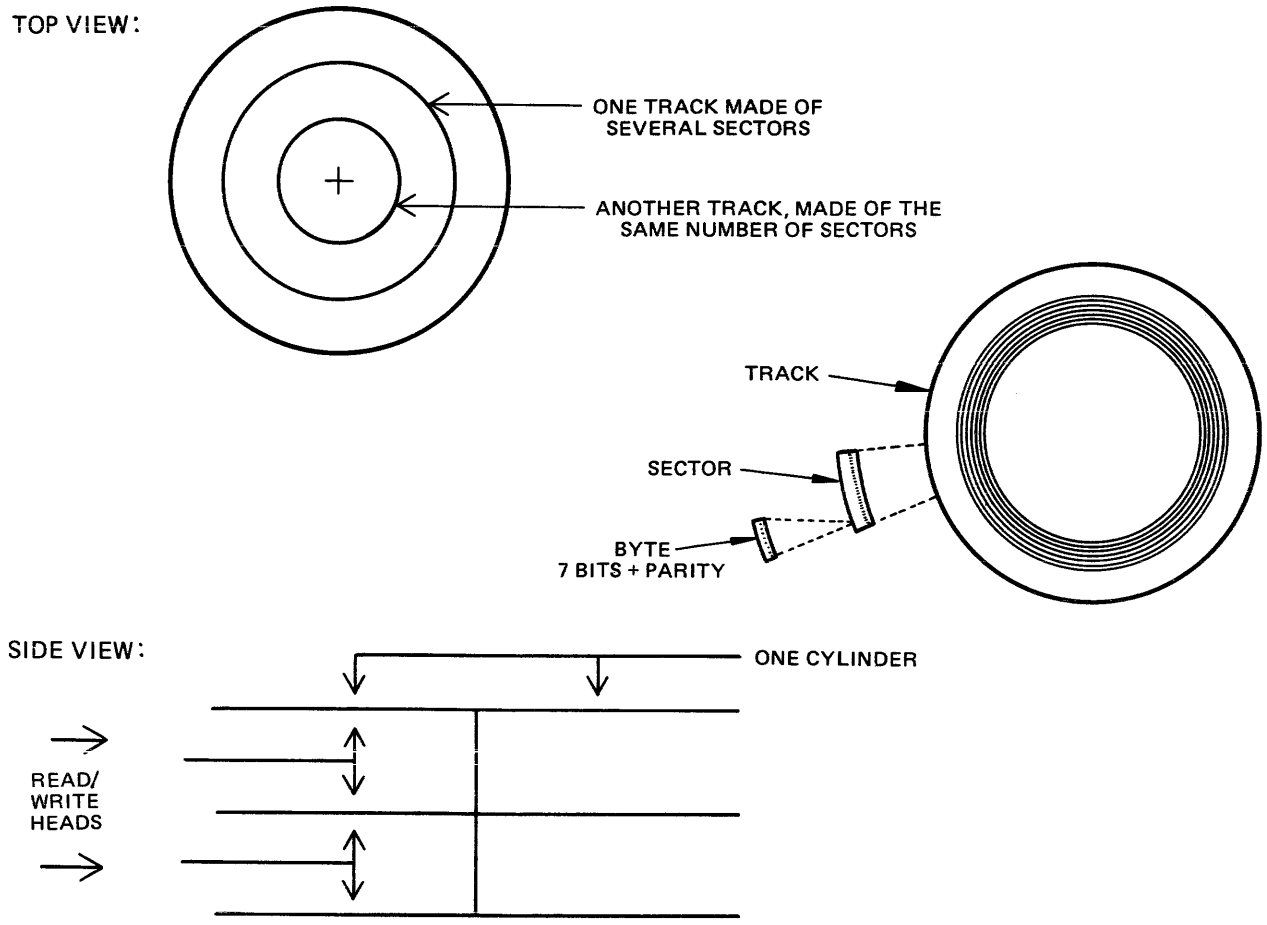


Figure 2-1. Physical Disk Structure

Disk Files

Information is stored on a disk in a "disk file". There may be many files on one disk. Each file is referenced by a "file name". A file name can be from one to twelve characters, using the set A to Z, 0 to 9, and the dot and hyphen. Each disk contains a directory of the files on that disk. This directory is accessed by utilities such as KA and PD (see section 4).

Information can be of different types: normal data, accessed by programs; special data, accessed by the MCP; and programs themselves. The MCP is itself a program, and so are other "system files" such as the interpreters. System files have special restrictions in that a control is placed on their removal (see RM, section 4).

Disk File Names

On any system, every disk file (whether data or a program) is accessed by a two part reference, as follows:

disk-name/file-name

For example, the disk file M101A/REP200 is a file with a file-name REP200 to be found on the disk with a disk-name M101A.

It is not necessary to give the name of the system disk when referring to files residing on the system disk. Alternatively, a disk-name of 0000000 by convention refers to the system disk. For example, the

disk file REP200 or 0000000/REP200 is a file with a file-name REP200 to be found on the system disk.

It is not allowed to have two disks of the same disk-name in use at the same time. It is not allowed to have two files of the same file-name on the same disk. However, it is quite permissible for two different disks to contain a file with the same file-name. For example, the files M100A/REP200 and M101A/REP200 refer to two different disk files (although one may be a copy or update of the other).

Disk File Group Names

In many utilities (see section 4) it is convenient to refer to groups of files, depending on common starting characters of their file-names.

All files on a disk may be referenced by the equals symbol ("="). For example, the reference M101A/= refers to all files on the disk with the disk-name M101A.

All files beginning with, say, the characters REP may be referenced by REP=. For example, the reference M101A/REP= refers to all files on disk M101A with file-names of REP200, REPA, REP678P, and so on.

In general, a group-name consists of an equals symbol ("=") optionally preceded by up to 11 symbols which are the first part of the file-names of each of the files in the group.

Example:

Consider a disk M101A containing files with file-names:

PR200, REP100, REP200, REP250, RQ510, CRCOPY

Then the following group-names refer to the files indicated:

M101A/=
PR200, REP100, REP200, REP250, RQ510, CRCOPY
M101A/REP=
REP100, REP200, REP250
M101A/R=
REP100, REP200, REP250, RQ510

Disk Directory

The disk directory is a table on every CMS-initialized disk which enables the MCP to locate any disk file by name. Full details of the directory layout are given in the CMS MCP manual.

The directory is a fixed size determined at disk initialization time, based on the maximum number of files to be placed on the disk. An attempt to create more files than there are entries in the directory will give an appropriate MCP run-time error message.

The directory consists of three parts:

the name-list
the disk file headers for each file
the available table

The relationship between these parts are given in figure 2-2. The name-list is a list, by file-name, of each file existing on that disk. A search through this name-list will reveal if a file is present or not:

if present, the name-list entry points to the disk file header for the file. This is a table giving the location of each part of actual data in the file (the file may be divided into up to sixteen separate physical areas on the disk). In the figure only one area is indicated. The available table is a list of the disk areas not in use by a file. When a new disk file is created, an available space is found from this table and an entry made in the name-list, then the space is used to write the file information. When a disk file is removed, its entry is deleted from the name-list and the areas specified in the disk file header are entered in the available table.

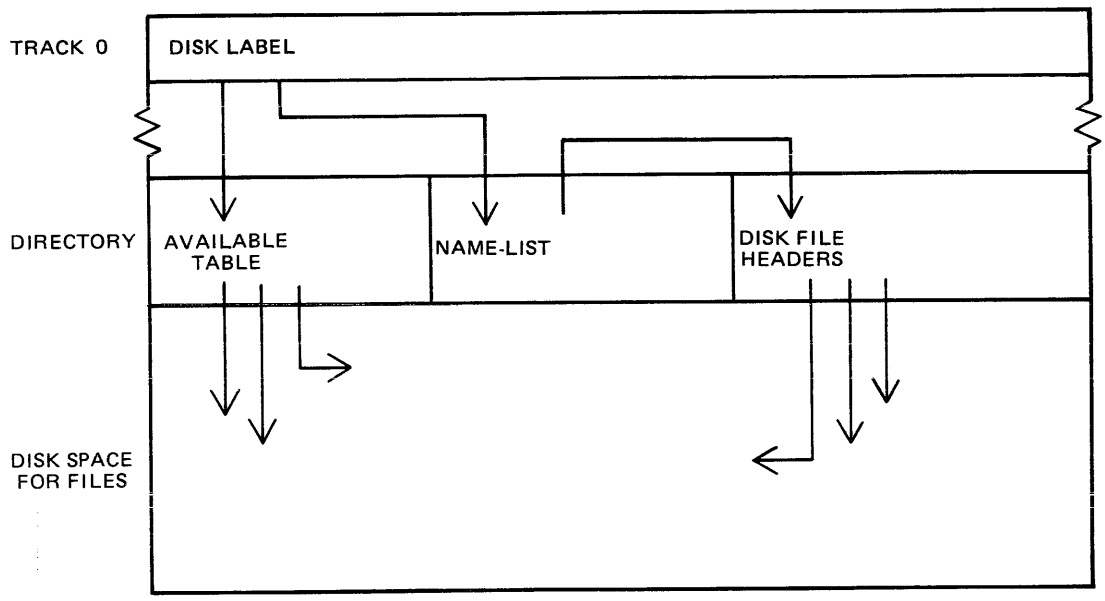


Figure 2-2. Disk Directory Structure

If there is insufficient space on a disk to allocate new disk file areas, COPY gets suspended by the MCP, waiting on available space. The operator may remove a file (see RM) to make more space available. The KA utility (see section 4) and KX function provide information on the available space on a disk. The B 90 Stand-Alone utility LS will also give the available space on a disk whose files have been listed by an LS <disk name>/=.

As a simplification, it may be stated that when a disk is initialized the directory is rebuilt with no entries, indicating that the entire disk space is available apart from the directory itself. In fact, any bad areas on the disk are marked in the directory so that they cannot be allocated to files (see also the XD utility); also, there is a special entry called "SYSMEM" which enables certain programs such as PD and RM (which access the directory) to operate successfully.

Indexed Files

Indexed files are in fact a pair of files, the "key file" and the "data file". They may reside on the same or separate disks. Each file in the pair has a separate entry in the disk directory of the disk on which it resides. A special table at the beginning of the key file (the "key file parameter block") gives, among other information, the disk-name and file-name of the associated data file. See figure 2-3 for a diagram of the relationships between the two files.

The purpose of indexed files is to simplify access to data in the data file by using a set of keys (such as account number) in each record of the data file. These keys are placed in the key file. A key file may be created by the SORT utility and intrinsic (see section 5, where examples are given).

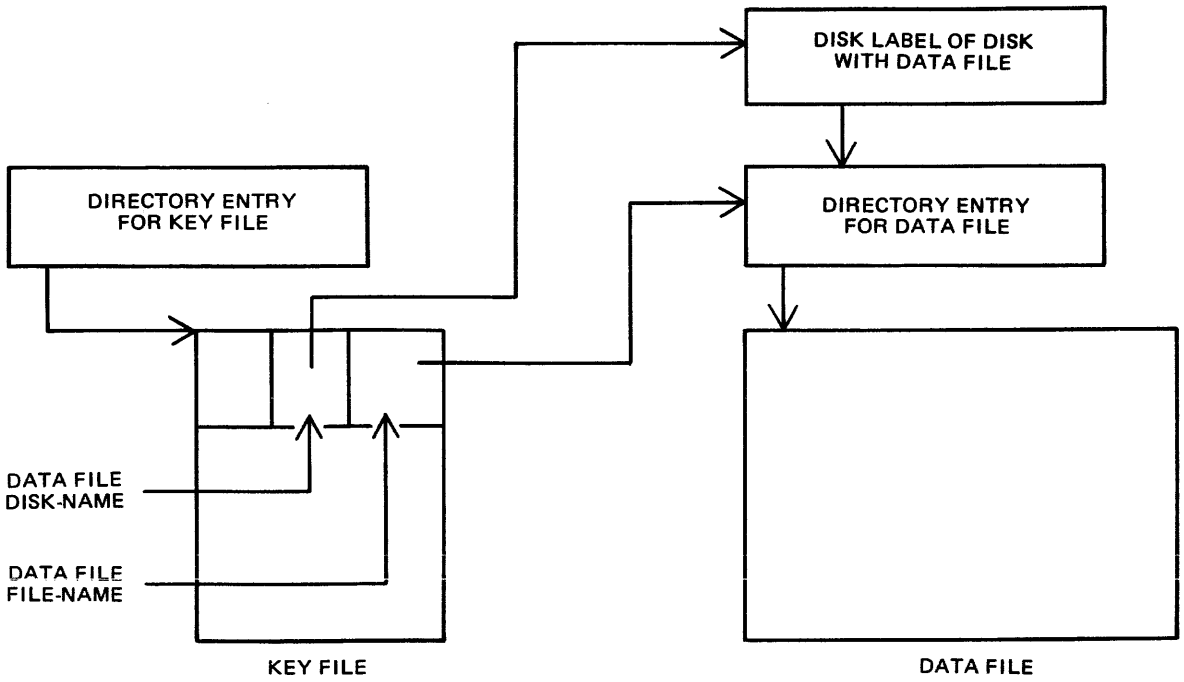


Figure 2-3. Indexed Files

Special consideration must be given to copying indexed files, due to the link between the key file and data file. This is especially true when copying from one disk to another. Details are given in each relevant section (see COPY utility, section 4; also the machine-dependent copy facilities).

Dual Pack Files

As mentioned before, a disk file may be divided into up to sixteen separate areas. If these areas are located on two separate disks, the file is known as a "dual pack file".

There is an entry in the directories of both disks for a dual pack file, together with the disk-name of the other disk. Each disk directory has a copy of the disk file header for this file, but the table of locations for each file area also indicates if the area is located on "this" disk or the "other" disk. This is shown diagrammatically for a file with four areas in figure 2-4. In most applications it is necessary for both disks of a dual-pack file to be on-line at the same time.

NOTE

No new dual-pack files will be created. An attempt to do so will produce an error message. If an old dual-pack file is accessed, a warning message will be issued. A future release will prevent access to old dual-pack files.

MAGNETIC TAPE FILE NAMES

Note: this includes tape cassette.

A tape may be used to store data either on one file (a "single-file tape") or as a "multifile" tape. Each file is separated by a tape mark. Additionally, each file normally has a beginning and an ending label. A multifile tape has also a special beginning ("volume") label.

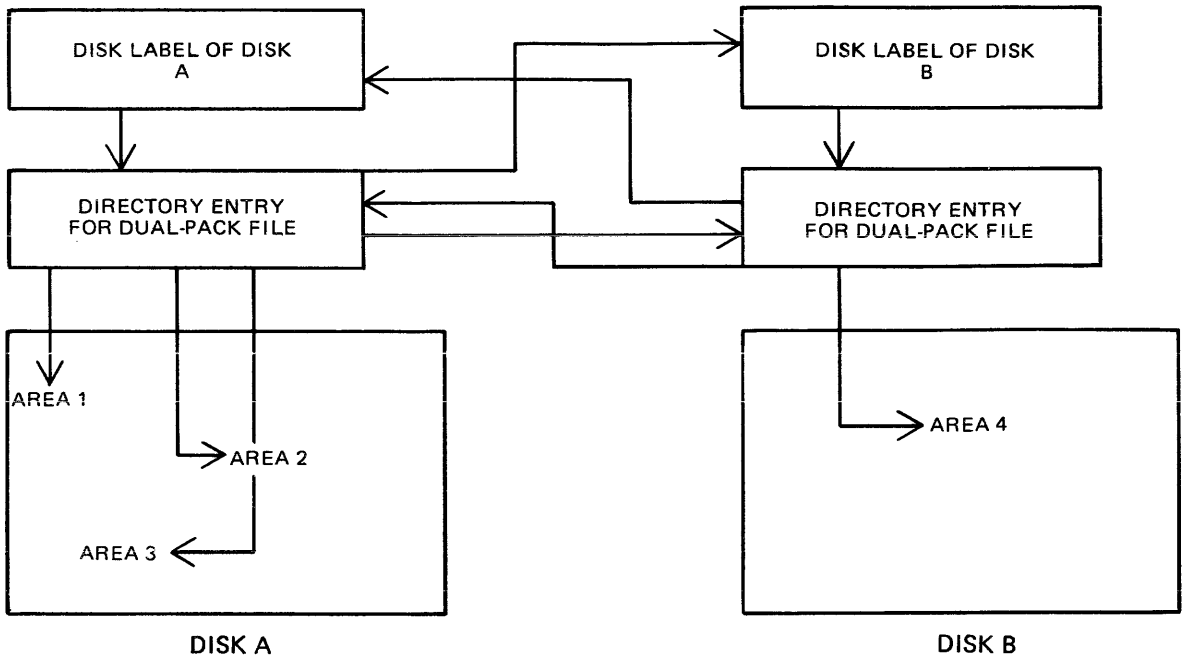


Figure 2-4. Dual Pack Files

On loading a tape, the MCP reads the first label to determine the tape name. Tape file names are in two parts:

multifile-name/file-name

For a single-file tape, the multifile-name will be "0000000". The format of the multifile-name is the same as for the disk-name of a disk file.

The COPY utility (section 4) produces a single-file tape when copying to tape. The backup utility and the LD utility (section 4) always produces multifile tapes called "library tapes". Library tapes are referenced by the multifile-name: there is a standard convention for labelling all the files on a library tape. For full details of tape formats, refer to the CMS MCP Reference manual.

Tapes (multi-file or single-file) may be unlabelled. Such tapes must always be accessed via the AD intrinsic (section 3) because there is no label that the MCP can recognize when the tape is loaded. Tapes containing labels that are non-standard are also treated as multifile unlabelled tapes.

PRINTER FILES

There are two types of printer: a wide line printer and a console printer, depending on available hardware. The console printer is also known as a "serial printer". These hardware devices are also referred to as "files" and are given file-names of up to seven characters. When the file is opened and closed, an identifying print line is given to indicate the name of the file. The identifying print line can be suppressed be setting the NO.LABEL option in the file parameter block (see MODIFY utility, section 4). This file-name is also used in MCP messages. Refer to the CMS MCP Reference manual for full details.

It is possible to designate a file type of "any printer". Such a file will be written to a wide line printer if this peripheral is available. If not available, this file will be written to the console printer if available.

If there is no console printer either, the MCP will display a "NO FILE" or "DEVICE REQUIRED" message, or if the printer backup option is on, it will direct the file to printer backup.

OTHER PERIPHERALS

All peripherals are treated as files for input, output or a combination of input/output, depending on the hardware type. The use of any peripheral device is governed by the file-name of up to seven characters, which will appear in any related MCP messages. Refer to the OL intrinsic (section 3) for other details.

PROGRAMS

An executable program is information stored on disk as a disk file. It is referenced in the same way as any data file: that is, through the disk-name and file-name (or just the file-name if the program resides on the system disk). The rules for the program name are the same as for any disk file-name.

A "utility" is a program provided for general use by all CMS operators, for house-keeping and other general purposes. For example, the LD utility enables operators to load and dump disk files from the disk to magnetic tape for backup purposes.

Executing Programs

In order to execute a program, part or all of the information in the disk file must be brought into memory and placed under control of the MCP. This is called "program load", and takes a certain interval of time.

Programs may be loaded and executed by merely providing the name of program file to the MCP. If so desired, the keyword "EX" may be placed before the program name. For example, suppose one wishes to execute a program that resides on a disk PR200A in a file called DCS. Either the input:

```
EX PR200A/DCS
```

or just

```
PR200A/DCS
```

will cause the program to be loaded and executed.

Depending on the system, a BOJ (beginning-of-job) message may be displayed by the MCP after the program has been loaded, and an EOJ (end-of-job) message may be displayed by the MCP at the end of the program. The display of these messages may be turned on or off for individual programs by the MODIFY utility (see section 4).

Failures may occur when attempting to load a program. For example, the requested program may not be on disk. A list of load failure messages is given in section 7.

Many programs enable the operator to enter further information after the program name. This is known as an "initiating message" and the contents are entirely dependent on the program. Nearly all the utilities in section 4 allow further information, the format of which is given in the description of each utility program. For example, the input:

```
COPY REP202 TO RPTAPE
```

consists of the command to load and execute the program called "COPY" (found on the system disk in this example), followed by the information "REP202 TO RPTAPE" which is passed to the program. There are two types of error which can be made: either there is a load failure (because, for example, the COPY program is not on the system disk), when the MCP would issue an appropriate message; or the following information is an incorrect format for the program, when the program itself would issue a message. In the former case, the MCP message is described in section 7. In the latter case, the output message is described under each utility.

Note that if the utility resides on, say, the disk PR2, the input message would be:

```
PR2/COPY REP202 TO RPTAPE
```

or

```
EX PR2/COPY REP202 TO RPTAPE
```

In section 4 this additional information is omitted in the interest of clarity. It is, however, common for utilities to reside on a disk other than the system disk, in which case the disk-name must be provided.

It is also possible for programs to be automatically executed by another program. In this case, the first program is said to "zip" the second program. No operator input is used in this case, but the BOJ message may be displayed for the zipped program.

INTRINSICS

There is an important type of operator input that does not involve a command to execute programs or utilities. These messages are calls on "intrinsics" which are part of the MCP. Those intrinsics which are common to all CMS machines are described in section 3. Other intrinsics are given in the relevant machine-dependent section.

Because an intrinsic is part of the MCP, there is no separate program corresponding to the name of the intrinsic. Therefore the keyword "EX" is not allowed in a call on an intrinsic, neither can a user disk-name be specified. There is no program load time because the MCP is already executing. For example, the input:

```
RY DMA
```

is a request to the MCP to ready (RY) the disk peripheral designated by DMA. This input message to the MCP must not be preceded by the keyword "EX".

Table 2-1 is a list of intrinsics and reserved commands which cannot be used as programs or disk names.

MIX NUMBERS

As a program is loaded, the MCP assigns it a number from its table of executing tasks. This is the "mix-number" and is used in any messages output by MCP relating to this task. The mix-number is also used in all messages input by the operator for this task. Some input messages also require the corresponding program name as well as the mix-number. The MX intrinsic (see section 3) may be used to determine the current mix of tasks.

The allocation of mix-numbers is dependent on the CMS product. Refer to the corresponding section for more details.

Table 2-1. Command Mnemonics

Command Mnemonic	Type of Command	Function	Restrictions (See Notes)
AD	SCL	Assign Device	3
AP	SCL	Add Pack	3
AX	SCL	Accept	4
CH	Superutility	Change	
CL	SCL	Clear	4
CO	Utility	Compile Utility	
CP	Utility	Compute	
CTL	SCL	Gain controlling SPO	
DA	Utility	Analyse disk	
DB	SCL	Direct to Backup	4
DC	SCL	Data Communications	
DD	Utility	Disk Dump	
DM	SCL	Dump	4
DP	SCL	Discontinue and Dump	4
DS	SCL	Discontinue	4
DT	SCL	Date	5
EX	SCL	Execute	
FD	SCL	Forms Define	3
FL	Utility	Display Directory	
FS	Utility	File Squash	
GO	SCL	Go	4
GT	SCL	General Trace	3
IR	Superutility	Initiate Recall	
KA	Utility	Disk Analyser	
KX	Superutility	Interrogate filesizes	
LB	Superutility	Continue recall	From earlier messages
LF	Superutility	Continue Recall	From recent messages
LG	SCL	Log Go	3
LR	Utility	List Regions	For disk only
LS	SCL	Log Stop	3
LT	SCL	Load Translation table	3
MX	SCL	Mix	
ND	SCL	New Density	3
OL	SCL	On line	
PB	Utility	Print Backup	
PD	Superutility	Print Directory	
PG	SCL	Purge Device	3
PL	Utility	Print Log	
PO	SCL	Power Off	3
QZ	Superutility	Create new log files	2
PR	SCL	Priority	3
RD	SCL	Reserve Device	3
RL	SCL	Relabel Pack	3
RM	Superutility	Remove	

Table 2-1. Command Mnemonics
(continued)

Command Mnemonic	Type of Command	Function	Restrictions (See Notes)
RY	SCL	Ready	3
SF	SCL	Substitute file	4 – not for printer backup files
SM	SCL	Multi-MCS data communications	4
SN	SCL	Serial Number	3
SO	SCL	Set PRBK Option	3
SQ	Utility	Squash Disk	
ST	SCL	Stop	4
SV	SCL	Save	3
TL	Utility	Transfer Log	
TO	SCL	Test Options	
VF	SCL	Vertical Format	3
WL	Utility	What log	
XD	Utility	Delete bad sectors	For disk only
ZQ	Superutility	Create new log files	1

Note 1: One of the Superutility functions; zipped only by warmstart and LG.

Note 2: Only zipped by Superutility.

Note 3: Only valid from the system SPO.

Note 4: Only valid from task's originating SPO (or from system SPO).

Note 5: Date/time may only be changed by the system operator, but can be interrogated by any operator.

OUTPUT MESSAGES

As mentioned earlier, messages may be output on the SPO either by the MCP and other system software or by the program. It is important to distinguish between the two types of output messages in order to look up the message in the appropriate place.

Messages output by the MCP are of two kinds: short responses to intrinsics, and longer descriptions of any event to be brought to the attention of the operator. The short descriptions are self-explanatory. For example, the input message:

```
OL LPA
```

(an intrinsic to inquire of the status of line printer LPA) may result in the response:

```
LPA READY
```

Similarly, the short message:

LPA NOT READY

will be displayed if LPA is stopped by the operator or through any fault. The longer descriptions are always referenced by an "event number" enclosed in brackets. The format of these messages is given in section 7, and operators should be generally able to recognize that such a message has been output by the MCP.

For example, the message:

10/LR PROGRAM SUSPENSION <14>
ASSISTANCE REQUIRED FOR PRINTER FILE LRPRINT
PRINTER NOT CURRENTLY AVAILABLE

indicates an MCP message with event number 14. Enter HELP 10 or refer to section 7 for information on possible causes and suggested actions to take.

Messages with event numbers may also be output by other parts of the system software such as interpreters and the sort-intrinsic, although the overall format is similar. After recognizing the event number, reference should be made to section 7 (or section 5 for sort-related messages).

Messages output by all other programs are known as "displays" and may be preceded by the keyword "DISP". Note, however, that utility programs may display messages without this preceding keyword.

All messages output by the utility programs described in this manual are listed under the respective utility. For example, messages displayed by COPY utility are listed under the COPY utility. Messages may additionally be displayed by the MCP for events related to the execution of the COPY, DSKUTL, LD and SCR utilities (for example, if the COPY utility is instructed to copy a file which does not exist, a "<file-name> NOT FOUND" message will be output), but these MCP messages will always be distinguished by the event number.

Messages displayed by other programs are not discussed in this manual. Reference must be made to the appropriate manual or operating instructions for that program.

Figure 2-5 illustrates a sample SPO list giving a mixture of messages described in this section. Note in this example that the utility programs LIST and LR do not give rise to BOJ and EOJ or DISP messages. The user program PROGA shows all three messages. These messages may be turned on for utilities by using the MODIFY utility (section 4).

RAILROAD DIAGRAMS

Most of the descriptions of input messages in this book are given as simple railroad diagrams with corresponding descriptive text and examples. See figure 2-6.

To form valid input, follow the railroad "track" from left to right or in the direction of the arrows. A junction in the track indicates that alternative paths may be followed. Items enclosed in angled brackets ("<" and ">") must be replaced with actual values, as before. Each item not enclosed in angled brackets is included as it is found. Spaces are added where necessary, as in format diagrams.

There are two other features available in railroad diagrams to make possible the exact specification of any input message. These are illustrated in figure 2-7.

```
input command to run LIST      →  LIST FRED
input command to run PROGA     →  PROGA
MCP message for PROGA BOJ      →    02/PROGRA commencing
input command to run LR        →  LR =
display information of         →   02/PROGA DISP: PROGRAM A
PROGA                             VERSION 35
MCP output message event 14    →   03 LR PROGRAM suspension<14>
                               →   Assistance required for
                                   printer file LRPRINT
                               →   Printer not currently
                                   available.
input request OL intrinsic     →  OL LPA
MCP Response to OL message     →   LPA LISTPRT in use by 01/list
input request to MX            →  MX
intrinsic
MCP response to MX message     →
                               →  NMBR prog-name  PR status
                               →  01/List          B executing
                               →  02/PROGA         A executing
                               →  03/LR        B suspended waiting
                                                on device required
MCP message for PROGA EOJ      →  02/PROGA TERMINATED
input request to Help          →  Help 3
intrinsic (B 90 ONLY)
MCP response to Help           →     Printer requested by 3/LR in
message                              used by 01/LIST
                                     * wait until printer device
                                       LPA free
                                     * enter "DB 3" to create
                                       printer backup file
                                   End Help
input request ST intrinsic     →  ST 1
MCP response to ST message     →     01/LIST stopped.
```

Figure 2-5. Sample SPO List

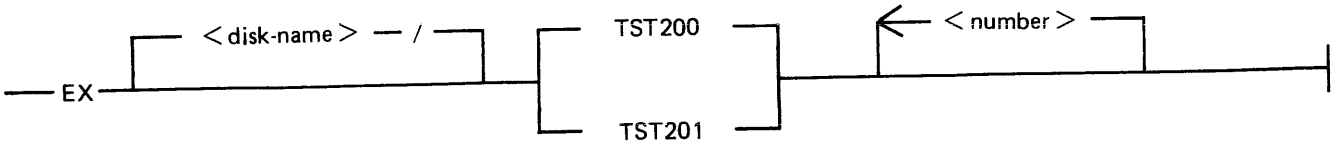


Figure 2-6. Railroad Diagram Sample 1

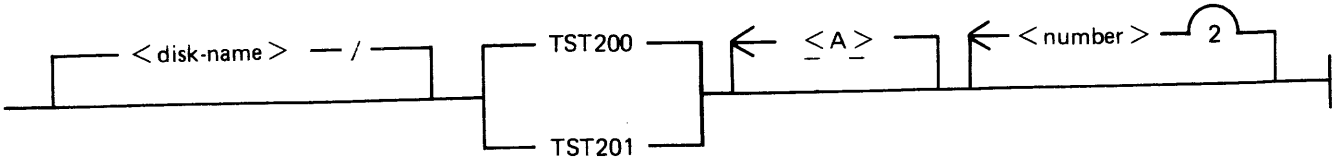


Figure 2-7. Railroad Diagram Sample

Firstly, the maximum number of times around a loop may be controlled by including the number in the track of the loop. In the example, it is possible to omit the <number>, or to include either one or two values of <number>. Secondly, if angled brackets are to be included as part of the message, these must be underlined. In the example, there is an optional part of the message which consists of the three characters "<A>". The following messages would then be valid:

```
EX PR2/TST200
EX PR2/TST200 27
EX PR2/TST201 27 56
EX PR2/TST201 <A>
EX PR2/TST200 <A> 56
```

but the following would be invalid:

```
EX PR2/TST200 27 56 243
EX PR2/TST201 A
EX PR2/TST201 A 73
```

Note also that if a number under a loop is preceded by an asterisk ("*"), then that loop must be included in the syntax at least the number of times specified. For example, if the loop included the characters "*1", then the loop must be included at least once.

SECTION 3
CMS COMMON INTRINSICS

INTRODUCTION

This section describes, in alphabetical order, those input commands which are embedded in ("intrinsic to") the MCP and which are common to all CMS products.

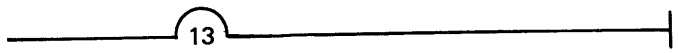
It is not valid to precede these messages with "EX", because the intrinsics are not separate programs to be loaded and executed. The intrinsics cannot be executed from a user disk, because they are part of the MCP which is on the system disk.

The response to these intrinsics may vary slightly between CMS products, due to different hardware being used. Where applicable, the variations have been noted in the text. The maximum number of characters input to the MCP via SPO (System Command Language, SCL) is 255 (including spaces).

SYNTAX RULES USED IN THIS SECTION

A syntax diagram is constructed of words formed of upper- and lower-case letters, arrows, special characters and digits. The basic rule is that any path traced along the forward directions of the arrows will produce a syntactically valid statement. All words formed of upper-case letters (except ASCII) and all special characters (for example, commas, colons, hyphens, slashes) in the diagram must appear in the message as shown (acute parentheses are special characters when they are underlined). All words formed of lower-case letters enclosed in acute parentheses are syntactic variables representing user-supplied information.

Any "bridge" over a number, such as



may be traversed a maximum number of times specified by the digit (thirteen times in this example).

Definition Of Syntactic Variables Used

mix	mix number.
packid	disk name or pack identifier. From 1 to 7 characters. The packid must not be "0000000".
mfid	multiple file identification. This may be packid or the name of a tape or cassette. From 1 to 7 characters.
fileid	file identification. The name of the file. From 1 to 12 characters.
progname	program name. From 1 to 12 characters.

Packid, mfid, fileid, progname may be constructed from the upper case alphabetic characters (A-Z), the numeric characters (0-9), the special characters hyphen (-) and point (.) and trailing spaces. If less than the maximum field size is input, the character string is padded with spaces on the right to complete the field size.

peripheral	a three character mnemonic. The first two characters describe the hardware device type and the final character describes the unit within that type. Units are specified by characters (A, B...) for example, DKC - disk cartridge, unit C.

device specifier	either a three character mnemonic specifying a particular device type and unit, as for peripheral, or a two character device type specifying the device family. For example, DK – all disk cartridges.
faulty input	The whole of the input message up to and including the first element detected to be erroneous. This forms part of many output messages.
device status	The current status of the device. This will take the form of the response to an OL intrinsic described in this section.
serial no	The six digit serial number of a disk, or the five character serial number of a tape.
reel	The tape reel number in a multi reel tape.
number	An integer number.
priority	The priority of a program (A, B or C).

All fields must be separated by one or more spaces.

Output Message Rules

All output messages are indented by three character positions to distinguish them from input messages. The three character positions are not available to user programs but are reserved for the system to provide additional information about the message.

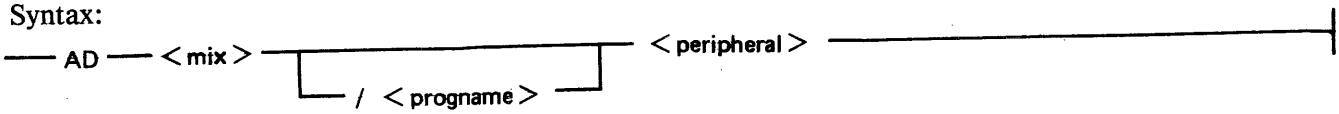
If the system breaks a line and inserts a CR/LF as a result of line overflow, the continuation line will have three dots in the first three character positions.

Messages which have been recalled from a log file are indicated by a "/" in the first character position of each line of the message, the exception being LR when the display option is specified.

On printer devices, each message is terminated by a carriage return to the start of a new line.

AD (Assign Device)

Syntax:



This intrinsic allows the operator to assign a peripheral to a program which requires a particular device.

Examples

Example 1:

LR utility requires a printer:

```
spo input     LR =
spo output    03/LR PROGRAM SUSPENSION <14>
spo output    ASSISTANCE REQUIRED FOR PRINTER FILE LRPRINT
spo output    PRINTER NOT CURRENTLY AVAILABLE
spo input     HELP 3
spo output    PRINTER REQUESTED BY 03/LR SAVED
spo output    *ENTER "RY LPA" TO MAKE PRINTER READY
spo output    *ENTER "AD 03 LPA" TO ASSIGN PRINTER
spo output    *ENTER "DB 03" TO CREATE PRINTER BACKUP FILE
spo output    END HELP
spo input     AD 3 LPA
spo output    END LR
```

The operator has assigned LPA to mix number 3.

Example 2:

The LIST utility requires an unlabelled tape:

```
spo input     LIST TAPE1 MTP NO.LABEL
spo output    04/LIST PROGRAM SUSPENSION <28>
spo output    ASSISTANCE REQUIRED FOR TAPE FILE SPURIOUS/TAPE1 001
spo output    USE "AD" TO ASSIGN DEVICE
spo input     AD 04/LIST MTA
```

AD Output Messages

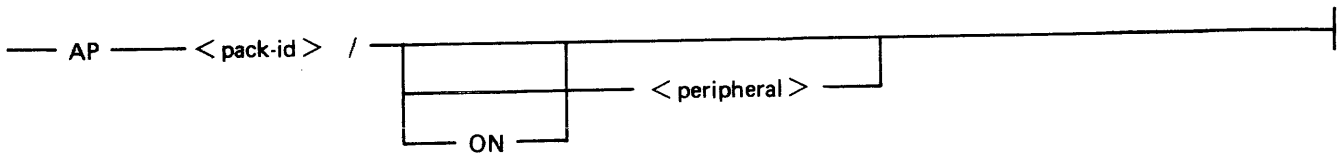
Message	Possible Causes	Suggested Action
AD faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS.	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program requiring device assignment. Re-enter assignment.
AD faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program requiring device

(continued)

Message	Possible Causes	Suggested Action
NUMBER. ENTER "MX" FOR MORE DETAILS		assignment. Re-enter assignment.
mix/progname NOT IN SUITABLE STATE TO ACCEPT "AD". ENTER "MX" FOR MORE DETAILS.	Specified program was not suspended waiting for a device assignment.	Check with MX for mix number of suspended program and re-enter.
AD faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS.	The peripheral is not attached to the system. (Not produced on B 90).	Check with OL for required peripheral and re-enter.
"AD" IGNORED – peripheral TEMPORARILY UNAVAILABLE FOR USE BY mix/progname. ENTER "OL" AND RETRY "AD" WHEN DEVICE AVAILABLE.	1. The peripheral is in use by another program. 2. The peripheral is physically not ready. 3. The peripheral is not saved.	1. Assign another device or wait until the device is free. 2. Use RY to ready the device or physically ready it. 3. Use SV to save the peripheral. Re-enter assignment.
DEVICE SPECIFIED IN "AD" COMMAND NOT SUITABLE FOR mix/progname. ENTER "MX" FOR PROGRAM STATUS AND REFER TO SOG.	The device assigned does not meet the requirements of the program requiring the device.	Check operating instructions for the program.
"AD" IGNORED – UNLABELLED DISK DRIVE peripheral FOR USE BY mix/progname MUST BE RESERVED FIRST. USE "RD" AND TRY AGAIN.	The drive was not reserved using the RD intrinsic.	Use RD to reserve the required drive and re-enter assignment.
"AD" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter AD from the controlling SPO device.
"AD mix/progname" IGNORED – DUAL PACK FILE CREATION IS NO LONGER PERMITTED	It is no longer possible to create a dual pack file.	Do not use dual pack files

AP (Add a Pack on system using Pseudo-pack, B 900 only)

Syntax:



This intrinsic is only applicable to pseudo-packs and causes an entry for the new packid to be added in the pseudo-pack identifier table (PPIT).

When a peripheral is specified, the pseudo-pack that is added is restricted to the specified device. Otherwise a non-restricted pseudo-pack will be added. This command can be used to resolve the NO-PACK condition. All physical units that have packids listed in the PPIT must be physically ready for the command to be accepted.

Example:

The following will add a new packid to the system:

```
spo input       AP PACK01/
spo output      AP PACK01/  ADDED TO SYSTEM
```

AP Output Messages

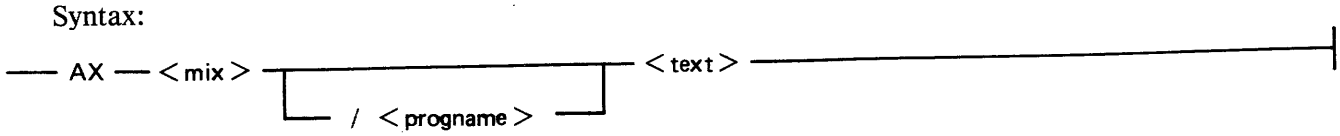
Message	Possible Cause	Suggested Action
AP faulty input INVALID – DISK NAME INCORRECTLY FORMED. REFER TO SOG FOR ALLOWABLE FORMAT	The packid specified is not valid.	Check input and re-enter AP.
AP faulty input IGNORED – CANNOT CREATE PSEUDO-PACK ON SYSTEM WITHOUT FIXED DISKS	There were no fixed disk units on the system when the AP intrinsic was entered.	Ensure that the fixed disk unit is connected to the system and is ready.
AP faulty input IGNORED – PSEUDO-PACKS CAN RESIDE ONLY ON FIXED DISK UNITS	The peripheral specified is not a fixed disk.	Check input and re-enter AP.
AP faulty input IGNORED – ALL DISKS BELONGING TO FIXED DISK ASSEMBLAGE MUST BE ON LINE BEFORE COMMAND ISSUED. ENSURE DISKS ARE POWERED ON	A physical unit from the PPIT is not ready.	Ready all physical units named in the PPIT before re-entering AP.

(continued)

Message	Possible Cause	Suggested Action
"AP" IGNORED – CANNOT ADD ANY MORE PSEUDO-PACKS TO SYSTEM. AN EXISTING PSEUDO-PACK MUST BE USED	The pseudo-pack identifier table is full.	None.
"AP" IGNORED – packid/ IS ALREADY ONLINE	There are two packs on the system with the same name.	Power off one of the packs then re-enter AP.
AP faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The specified peripheral is not on the system.	Ensure that the required peripheral is connected and ready before re-entering AP.
AP faulty input IGNORED – HARDWARE ERROR ON peripheral. IF DISK, RUN "CHECK.DISK". IF TAPE, REPLACE MEDIA	A hardware error prevented the intrinsic being executed.	Seek technical assistance.

AX (Accept a message for a program)

Syntax:



This intrinsic allows the operator to communicate with a program which is suspended waiting for an "AX" command.

The MCP will prompt the operator for input by printing "mix-number/program-name REQUIRES DATA, USE "AX" on the SPO (SCL) device.

Any input in the text field that exceeds the size specified by the program waiting on the accept will be truncated. Consult the operating instructions for the program for valid "text" responses.

NOTE

"text" begins at the first non-space character following mix-number(/program-name). Also a null response is a single space in the text field. A space is required between the mix number and (program-name).

Example:

The program BM001 displays a message asking for a file name to be entered. The operator responds with the appropriate text, in this case ARSCHG, by the AX message.

```
spo input      BM001
spo output     01/BM001 COMMENCING
spo output     01/BM001 DISP: ENTER BM202 FILE NAME
spo output     01/BM001 REQUIRES DATA, USE "AX"
spo input      AX 1/BM001 ARSCHG
```

AX Output Messages

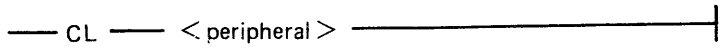
Message	Possible Causes	Suggested Action
mix/progname "string" REQUIRES DATA. USE "AX"	Program is suspended waiting for an accept.	Enter accept.
AX faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program requiring accept. Re-enter accept.
AX faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program requiring accept. Re-enter accept.
mix/progname NOT IN SUITABLE STATE TO ACCEPT "AX". ENTER "MX" FOR MORE DETAILS	Specified program not waiting for an "accept".	Use MX to find mix number of program requiring accept. Re-enter accept.

(continued)

Message	Possible Causes	Suggested Action
"AX" IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The program was not executed from this SPO device.	If this is the correct mix/prog-name then enter the accept from the SPO device that initiated the program or the controlling SPO device.

CL (Clear Device)

Syntax:



This intrinsic is valid for all devices attached to the system except disks. It is used by the operator to clear the status of a peripheral. It can only be used when the program using the device has been DS'ed or DP'ed and is waiting to be cleared to allow the program to terminate. This restriction does not apply to B 1000.

For example, if the line printer jams during the printing of a report and the program producing the report was DS'ed, the program will not terminate (go to End of Job) until the line printer is made ready or cleared using the CL intrinsic.

Examples:

```
spo input CL LPA
spo input CL MTA
```

CL Output Messages

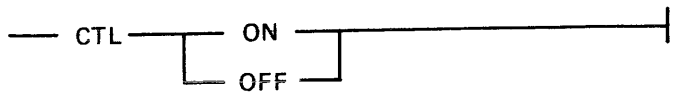
Message	Possible Causes	Suggested Action
CL faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS	The specified peripheral is illegal.	Check input and clear the peripheral that was in use by the DS'ed/DP'ed program. Re-enter clear.
mix/progname NOT IN SUITABLE STATE TO ACCEPT "CL". ENTER "MX" FOR MORE DETAILS	The program has not been DS'ed or DP'ed. (Not produced on B 90.)	DS or DP the program that is causing the trouble and clear the device if required.
CL faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The peripheral specified is not attached to the system.	Use OL to find the peripheral that the program is attached to. Re-enter clear.
"CL" INVALID – DEVICE peripheral CANNOT BE CLEARED UNLESS IT IS IN USE BY A PROGRAM WHICH HAS BEEN TERMINATED BY THE OPERATOR. ENTER "OL" AND "MX" FOR MORE DETAILS	The peripheral specified is not in use by the waiting program.	Use OL to find the peripheral that the program is attached to. Re-enter clear.

(continued)

Message	Possible Causes	Suggested Action
"CL" IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The program was not executed from this SPO device.	Ensure that the SPO device used is the one that initiated the program, or use the SYSTEM SPO device.

CTL (Gain/Relinquish System SPO, not implemented on B 1000 systems)

Syntax:



This intrinsic allows the operator from a terminal SPO to ask for SYSTEM SPO status by issuing "CTL ON". Certain SCL commands (see table 2-1) can only be issued from the system SPO and only it receives certain system messages. The system SPO also has the unique ability to interact with tasks that have been originated from other SPOs. In response to a successful CTL ON command, the MCP replies with the following message:

```
COMPUTER MANAGEMENT SYSTEM (CMS)
MCP VERSION 03.05.+ +  + + + +

"CTL" ON COMPLETE - SYSTEM SPO STATUS ACQUIRED
```

Also if the date has not been set:

```
IF ACCURATE DATE/TIME IS REQUIRED,
UPDATE USING 'DT'
```

CTL Output Messages

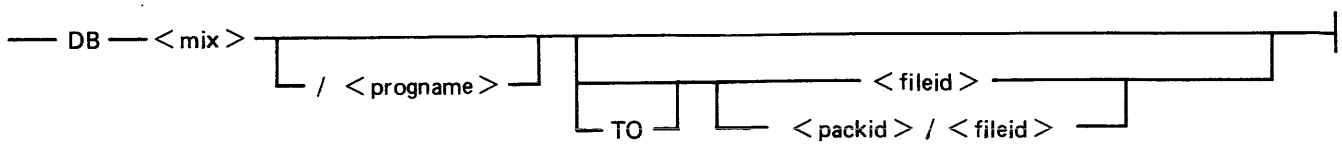
Message	Possible Causes	Suggested Action
CTL faulty input IGNORED - TERMINAL NOT IN SUITABLE STATE TO BECOME SYSTEM SPO. REFER TO SOG FOR MORE DETAILS	1. The station or line is not ready.	1. Ready station or line then re-enter CTL.
	2. The station is disabled input or output.	2. None.
	3. The MCS owning the station is not running.	3. None.
'CTL ON' IGNORED - THIS ALREADY IS THE SYSTEM SPO	This terminal is already the SYSTEM SPO.	None.
'CTL ON' IGNORED - ANOTHER SPO CURRENTLY HAS SYSTEM SPO STATUS	A SYSTEM SPO already exists.	None.
SYSTEM SPO STATUS REMOVED	1. A successful CTL OFF has been performed. 2. The local SPO has taken SYSTEM SPO status. 3. The line or station has gone not ready.	

(continued)

Message	Possible Causes	Suggested Action
'CTL OFF' IGNORED – THIS IS NOT THE SYSTEM SPO	This terminal did not have SYSTEM SPO status.	None.

DB (Direct To Backup)

Syntax:



where

packid	This is the name of the disk or pack on which the printer backup file will be placed. If no disk name is specified, the printer backup designate disk will be used.
fileid	This is the file name which will be given to the printer backup file. If no file name is given, a file name of the form PBnnnnn (where nnnnn is a five digit progression number in the range 00001 to 65535) will be used.

This intrinsic allows the operator to direct the printer output of a program, which is waiting for a physical printer, to a printer backup file if there is a printer availability problem. DB overrides both the option selected by the program, via its File Parameter Block, and the system options in force, due either to the intrinsic SO or the default contained in SYSCONFIG file (see CONFIGURER utility.)

Examples:

Example 1.

The program WRITER is waiting on a printer:

spo output	22/WRITER <28> PROGRAM SUSPENSION <14> ASSISTANCE REQUIRED FOR PRINTER FILE WRITER PRINTER NOT CURRENTLY AVAILABLE
spo input	DB 22
spo output	22/WRITER PRINTER BACKUP FILE printer-backup-designate-disk-name/ PBnnnnn OPENED
spo output	22/WRITER PRINTER BACKUP FILE printer-backup-designate-disk-name/ PBnnnnn CLOSED

A printer backup file called PBnnnnn has been written to the system printer backup designate disk (if available, otherwise the system disk will be used). Note that nnnnn is a five digit progression number generated by the system.

Example 2.

To direct the output of the program BM902 to printer backup on disk named PRINT02 with the file name BMPRINT the following occurs:

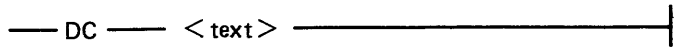
spo output	01/PRINTER PROGRAM SUSPENSION <14> ASSISTANCE REQUIRED FOR PRINTER FILE BM302 PRINTER NOT CURRENTLY AVAILABLE
spo input	DB 01/PRINTER TO PRINT02/BMPRINT
spo output	01/PRINTER WRITER BACKUP FILE PRINT02/BMPRINT OPENED
spo output	01/PRINTER WRITER BACKUP FILE PRINT02/BMPRINT CLOSED

DB Output Messages

Message	Possible Causes	Suggested Action
DB faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program requiring printer device. Re-enter direction to backup.
DB faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program requiring printer device. Re-enter direction to backup.
mix/progname NOT IN SUITABLE STATE TO ACCEPT "DB". ENTER "MX" FOR MORE DETAILS	1. The mix number specified did not require a printer.	1. Use MX to find the mix number of the program requiring printer device.
	2. The initiating message was incomplete.	2. Check input. Re-enter direction to backup.
DB faulty input IGNORED – SPECIFIED PRINTER BACKUP DISK NOT ONLINE. RETRY COMMAND WITH DISK ONLINE	The disk specified is not present.	Load and ready required disk and re-enter direction to backup.
DB faulty input IGNORED – DUPLICATE PRINTER BACKUP DISKS ONLINE. POWER OFF ONE AND RETRY COMMAND	Two disks with the specified name exist on the system. This will only occur on systems which run with Pseudo packs.	Remove one of the disks and re-enter direction to backup.
DB IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The program was not executed from this SPO device.	Ensure that the SPO device used is the one that initiated the program or use the SYSTEM SPO device.

DC (Data Communications operator input)

Syntax:



This intrinsic allows the operator to enter messages from the SPO (SCL) device to the Message Control System (MCS), if the data communications activity is in process. The message text, after being stripped of the "DC" characters and leading spaces, is transferred to the MCS input message queue and marked as "operator input".

The interpretation of the message text is defined by the particular MCS.

Example:

To enter the text "RY S 0" the following would be input:

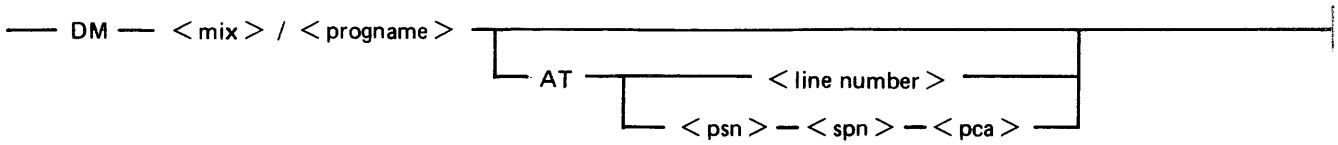
```
spo input     DC RY S 0
spo output    FROM MCS: RY OK (126)
spo output    FROM MCS: STATION 0/AP300XA31 READY (212)
```

DC Output Messages

Message	Possible Causes	Suggested Action
"DC" INVALID – NO MCS PROGRAM RUNNING	There is no MCS in the mix.	Check input. Execute the MCS if required. Re-enter DC command.
"DC" IGNORED – NO DATACOMM BUFFER SPACE AVAILABLE FOR MCS MESSAGE. TRY AGAIN LATER	There is no DC message space, in which to transfer the message text.	Wait a short time, then re-enter message. If unsuccessful several times, request technical assistance.
"DC" IGNORED – FIRST DEFINED MCS NOT RUNNING. USE "SM" TO SPECIFY MCS REQUIRED	The MCS is not running.	Check input. Execute the MCS if required. Re-enter DC command.

DM (Dump program, only implemented on B 1000)

Syntax:



This intrinsic allows the operator to create a dumpfile at any point of program execution, without discontinuing the program. When the dump is finished, the program can be restarted with the GO command.

If the mix/progname only is specified, the dumpfile will be created immediately. For COBOL programs, a line number can be specified (in the range 1 to 65535) and the dumpfile will be created when the program reaches that line number. Note that the COBOL program must be compiled with a $LINECOUNT card.

For MPLII programs, the break point where the dump file is created can be described by:

psn = code segment number
spn = procedure number
pca = procedure offset

where psn and spn are in the range 0 to 255, and pca is in the range 0 to 65535. The correspondence between a particular program statement and the above parameters can be found by using the $XMAP control card at compile time or MPL.LISTS.

Example:

The COBOL user program called TESTER is to be dumped at line number 25000. (The mix number of TESTER is 20, this may be found using the MX command).

```
spo input      DM 20/TESTER AT 25000
spo output     DUMPFILE SYSTEM/DMFIL20 CREATED
               RUN DUMP ANALYSIS UTILITY TO DETERMINE PROGRAM FAULT
```

DM Output Messages

Message	Possible Causes	Suggested Action
DM faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix	Use MX to find the mix number of the program to be discontinued. Re-enter DM input.
DM faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program to be discontinued. Re-enter DM input.

(continued)

Message	Possible Causes	Suggested Action
DM faulty input INCOMPLETE – BOTH MIX NUMBER AND PROGRAM NAME MUST BE SPECIFIED. ENTER "MX" FOR MORE DETAILS	The program name is also required.	Re-enter DP input with program name.
DUMPFILE NOT CREATED FOR mix/progname "STRING" DUE TO DISK HARDWARE ERROR	An error has occurred while opening, writing to or closing the dumpfile. The program, however, can still be restarted if desired.	Call for technical assistance.
DUMPFILE packid/ DMFILmix CREATED. RUN DUMP ANALYSIS UTILITY TO DETERMINE PROGRAM FAULT	The dump file is created. with the name disk-name/ DMFILmix-number.	Now DUMP ANALYZE utility can be run to determine program fault.

DP (Discontinue and Dump)

Syntax:

This intrinsic is similar to the "DS" intrinsic. The difference is that the disk work space (Virtual Memory on Disk) is not cleared and returned to an available status.

Instead, the disk work space is updated from memory with all the most current information about the program. The disk work space is then made into a file and given a name, "DMFILnn" where 'nn' is the mix number of the program that was DP'ed.

The peripherals and memory in use by the specified program are made available to other programs.

DP is used when a technical analysis of a particular program is required following a failure during its operation. The DMFILE produced using the DP intrinsic can then be analyzed using the system's DUMP/ANALYZER utility. See section in this manual relating to system being used for description of the DUMP/ANALYZER utility.

Example:

The user program called GL060 must be terminated. The following would be input on the SPO device to dump the program.

```
spo output      20/GL060 MUST BE TERMINATED
spo output      ENTER:
                "DS 20/GL060" - TO TERMINATE OR
                "DP 20/GL060" - TO TERMINATE WITH DUMP
spo input       DP 20/GL060
spo output      DUMPFILE DMFIL20 CREATED
spo output      RUN DUMP ANALYSIS UTILITY TO DETERMINE PROGRAM FAULT
spo output      20/GL060 TERMINATED (DP'ED)
```

DP Output Messages

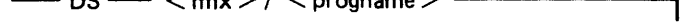
Message	Possible Causes	Suggested Action
DP faulty input INVALID - NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program to be discontinued. Re-enter DP input.
DP faulty input INVALID - PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS.	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program to be discontinued. Re-enter DP input.
DP faulty input INCOMPLETE - BOTH MIX NUMBER AND PROGRAM NAME MUST	The program name is also required.	Re-enter DP input with program name.

(continued)

Message	Possible Causes	Suggested Action
BE SPECIFIED. ENTER "MX" FOR MORE DETAILS		
DUMPFILE NOT CREATED FOR mix/progname "STRING" DUE TO DISK HARDWARE ERROR	For some hardware reason the DMFILE cannot be created.	Call for technical assistance.
DUMPFILE packid/ DMFILmix CREATED. RUN DUMP ANALYSIS UTILITY TO DETERMINE PROGRAM FAULT	The dump file is created with the name disk-name/ DMFILmix-number.	Now dump analysis utility can be run to determine the program fault.
"DP" IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The program was not executed from this SPO device.	Ensure that the SPO device used is the one that initiated the program, or use the SYSTEM SPO device.

DS (Discontinue Program)

Syntax:



This intrinsic causes the orderly termination of the specified program. All peripherals in use by the program are made available to other programs.

NOTE

The CL intrinsic may have to be used to allow the program to terminate if it is waiting for a NOT READY device. See CL intrinsic.

Example:

The user program called GL060 must be terminated. The following would be input on the SPO device to discontinue the program.

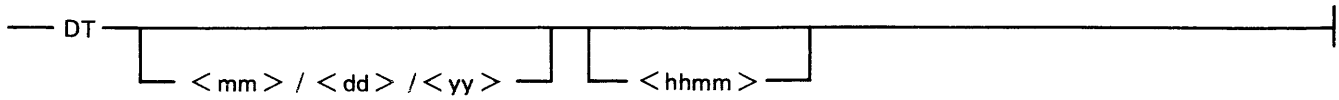
spo output	20/GL060 MUST BE TERMINATED
spo output	ENTER: "DS 20/GLO60" – TO TERMINATE OR "DP 20/GLO60 TO TERMINATE WITH DUMP
spo input	DS 20/GL060
spo output	20/GL060 TERMINATED (DS'ED)

DS Output Messages

Message	Possible Causes	Suggested Action
DS faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program to be discontinued. Re-enter DS input.
DS faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program to be discontinued. Re-enter DS input.
"DS" IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The program was not executed from this SPO device.	Ensure that the SPO device used is the one that initiated the program or, use the SYSTEM SPO device.
DS faulty input INCOMPLETE – BOTH MIX NUMBER AND PROGRAM NAME MUST BE SPECIFIED. ENTER "MX" FOR MORE DETAILS	The program name is also required.	Re-enter DS input with program name.

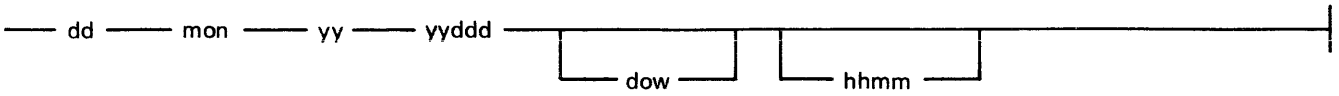
DT (Set System Date and Time)

Syntax:



This intrinsic allows the operator to change the system's date and/or time. Also, it allows the operator to interrogate the system date and time (if there is a real time clock on the system).

The response to a valid DT command is of the format:



where:

dd	=	The day of the month.
mon	=	A three character abbreviation for the month (for example, JAN, FEB)
yy	=	The last two digits of the year (for example, 82, 83)
yyddd	=	The julian date.
dow	=	A three character abbreviation for the day of the week (for example, SUN, MON)
hhmm	=	hh is the hour (24 hour clock) and mm is the minutes.

Examples

Example 1.

To inquire about the system date (and time if the system contains a real time clock):

```
spo input       DT
spo output      12 OCT 82 82285 TUE 1132
```

Example 2.

To set the system date and time to July 9th 1975 and 2:35 p.m.

```
spo input       DT 7/9/75 1435
spo output      09 JUL 75 75190 WED 1435
```

Example 3.

To just change the time to 11:45 a.m.

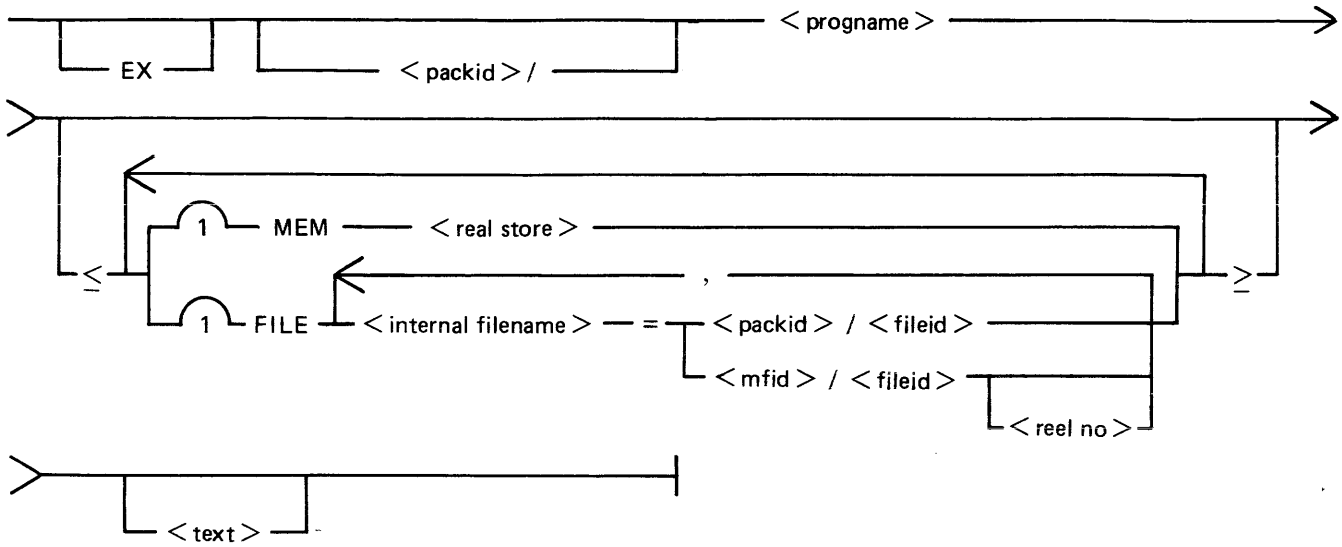
```
spo input       DT 1145
spo output      09 JUL 75 75190 WED 1145
```

DT Output Messages

Message	Possible Causes	Suggested Action
DATE INCORRECT. USE FORMAT "MM/DD/YY" WITH VALID NUMBERS.	The date is incorrect.	Check input and re-enter DT command.
TIME INCORRECT. USE FORMAT "HHMM" WITH VALID NUMBERS	The time is incorrect.	Check input and re-enter DT command.
NO CLOCK ON SYSTEM. TIME PARAMETER IGNORED	The time has been input to a system without a real time clock.	None.
"DT" IGNORED – DATE OR TIME MAY ONLY BE CHANGED BY SYSTEM OPERATOR. OTHER OPERATORS MAY ONLY INTERROGATE DATE OR TIME.	The time and date can only be changed from the system SPO.	Re-enter DT from the controlling SPO device.

EX (Execute a program)

Syntax:



This intrinsic allows the operator to initiate the loading and subsequent execution of the specified program.

The optional fields contained within acute parentheses, "<" and ">", are special instructions to the loader. The special instructions may be in any order. They are identified by their keyword (MEM and FILE) and terminated by another keyword or the closing acute parenthesis.

The use of the real store special instruction MEM is machine dependent; the value specified is a number of bytes. See relevant MCP reference manual.

The FILE special instruction specifies a comma list of file name equations. The equation is in terms of the internal file name as recorded in the internal file name block of the program file. The equation is therefore invariant under File Parameter Block (FPB) modification. A FILE special instruction causes the specified internal filename to be substituted for the filename in the FPB which corresponds to it. It should be noted that filename equation will be ineffective for programs which dynamically edit their FPB. (All utility programs dynamically edit their FPB.)

The text field is dependent on the program. See operating instructions for the particular program.

Examples:

Example 1.

```
spo input EX LIST FRED A
```

This executes the LIST utility and passes the text "FRED A" to the LIST utility.

Example 2.

```
spo input LR =
```

This executes the LR utility and passes the text "=" to the LR utility.

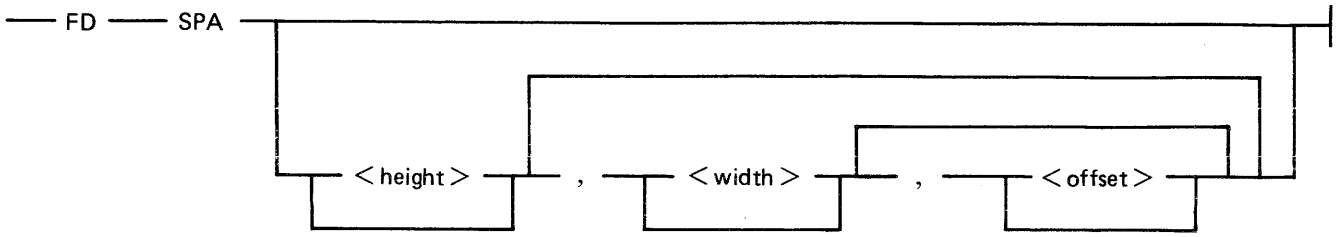
EX Output Messages

For error messages see section 7, Loader Detected Failures.

FD (Form Define not implemented on B 1000)

Syntax:

For serial printer:



For terminal SPOs running Multiple Terminal SPO (MTS), B 90/B 900 systems only:

This intrinsic allows the operator to define a logical page for a serial printer (SPA) or set top of page for SPA. When used with STATION, the command can only be issued from a terminal SPO on a system running MTS.

The three parameters are as follows:

height	This is the page height which defines the number of lines forming a page. When page height has not been defined or has been defined as zero (0), skip to top of form will execute four line advances, giving three blank lines (serial printer only). If specified for a terminal SPO this value must be in the range 1 to 255 and be suitable for the terminal configuration used.
width	This is the page width (maximum record size), which specifies the number of characters in a line. If specified for a terminal SPO this value must be in the range 20 to 255 and be suitable for the terminal configuration used.
offset	This is the physical column number, on the printer, corresponding to column zero (0) of the logical page.

NOTE

The WIDTH and OFFSET added together must not be greater than the number of physical print positions on the serial printer. For example, if the physical printer had 255 columns, the maximum printing area is given by a WIDTH of 255 and OFFSET of zero.

The FD command may also be applied without any parameters to a serial printer. This declares the current position of the loaded form to be at top of page.

When any of height, width, or offset are changed the new values are recorded, and remain in effect after system shutdown and warmstart.

Examples:

Example 1.

spo input FD SPA 63,120,105

This sets 63 lines as the page height, 120 characters as the page width and first character on the line will be at 105th print position of the physical printer.

Example 2.

spo input FD SPA , ,95

This changes the start position of the first character print on a line to the 95th print position of the physical printer. The HEIGHT and WIDTH settings will remain as they were.

Example 3.

In order to change the top of page position, move the paper to the desired position then input:

spo input FD SPA <empty>

Example 4.

To set a SPO capable terminal to have page size of 24 and width of 80 (providing the terminal has been successfully configured with these values) input from the terminal:

spo input FD STATION 24,80

FD Output Messages

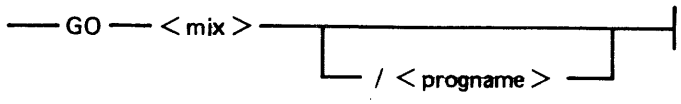
Message	Possible Causes	Suggested Action
FD faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS.	There is no serial printer on the system.	Check with OL to find mnemonic for the serial printer on the system. Re-enter FD command.
FD faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE.	The serial printer is in use.	Wait until the printing has ended then re-enter FD command.
FD faulty input INVALID – PARAMETERS ILLEGAL OR OUT OF RANGE. REFER TO SOG FOR ALLOWABLE FORMAT.	The FD specifications are not acceptable.	Check input and re-enter FD command.

(continued)

Message	Possible Causes	Suggested Action
FD faulty input INVALID – HEIGHT MUST BE IN RANGE 1-255 AND WIDTH IN RANGE 20-255. RETRY "FD STATION" WITH CORRECT PARAMETERS.	The parameters specified in an FD command are found to be out of range.	Check input and re-enter FD STATION command with valid parameters.
FD faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS.	An illegal peripheral was specified in an FD command.	Check input and re-enter FD command.
FD faulty input INVALID – "FD STATION" MAY ONLY BE USED TO ALTER SCL TERMINAL SCREEN SIZE. REFER TO SOG FOR MORE DETAILS.	An FD station command was entered from SPO which is not a terminal SPO.	None.

GO (Restart a Stopped Program)

Syntax:



This intrinsic allows the operator to restart a program which has been stopped with the ST or DM (B 1000 only) command or a PAUSE communicate.

Examples:

Example 1.

To restart the program whose mix number is 3:

```
spo input GO 3
```

Example 2.

To restart the program PRO20 whose mix number is 20:

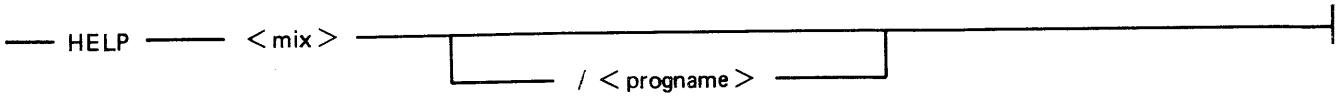
```
spo input GO 20/PRO20
```

GO Output Messages

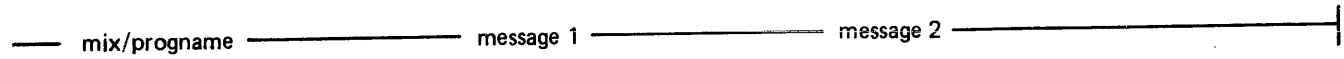
Message	Possible Causes	Suggested Action
GO faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program to be restarted. Re-enter GO command.
GO faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program suspended waiting for "GO". Re-enter GO command.
"GO" IGNORED – mix/ progname NOT STOPPED. ENTER "MX" FOR MORE DETAILS STOPPED	Specified program was not suspended as a result of an ST command. (Not produced on a B 90 when the program is executing.)	Check with MX for suspended program waiting on "GO". Re-enter GO command.
GO IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR.	The program was not executed from this SPO device.	Ensure that the SPO device used is the one that initiated the program or use the SYSTEM SPO device.

HELP (Detailed Task Information on B 90 Only)

Syntax:

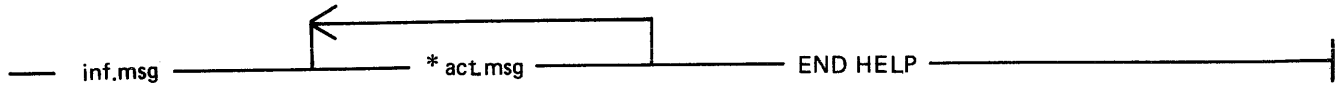


This intrinsic allows the operator to inquire about the state of any program currently processing. If the specified program is not suspended when the HELP intrinsic is issued, then a status message similar to the MX intrinsic response is produced:



For message1 and message2 refer to the MX intrinsic.

If the program is suspended, the output of the HELP intrinsic is:



inf.msg – information message
act.msg – one or more action messages, each one commencing with "*"

"*" is used to indicate alternative procedures to resolve the suspension. Only one should be chosen by the operator from each list of action messages.

Examples:

Example 1:

```
spo input      LR =
spo output     04/LR PROGRAM SUSPENSION <14>
spo output     ASSISTANCE REQUIRED FOR PRINTER FILE LPRINT
spo output     PRINTER NOT CURRENTLY AVAILABLE
spo input      HELP 4
spo output     *ENTER "RY LPA" TO MAKE PRINTER READY
spo output     *ENTER "AD 04 LPA" TO ASSIGN PRINTER
spo output     *ENTER "DB 04" TO CREATE PRINTER BACKUP FILE
spo output     END HELP
```

Example 2:

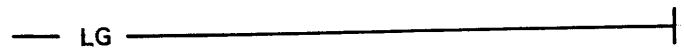
```
spo input      MX
spo output     NMBR PROG-NAME PR STATUS
spo output     12/FATALPROG      B SUSPENDED WAITING ON OPERATOR INPUT
spo output     END MX
spo input      HELP 12
spo output     12/FATALPROG ENCOUNTERED FATAL ERROR AND MUST BE
               TERMINATED
spo output     *ENTER "DS 12/FATALPROG" TO TERMINATE
spo output     *ENTER "DP 12/FATALPROG" TO TERMINATE WITH DUMP
spo output     END HELP
```

HELP Output Messages

Message	Possible Cause	Suggested Action
HELP faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX without mix number.
HELP faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX without the mix number and the program name.

LG (Start system message logging)

Syntax:



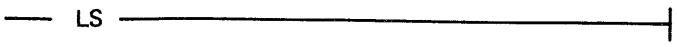
This intrinsic starts system message logging. Logging continues until either the system is powered off (at the next warmstart the default option set via CONFIGURER is acted on), or countermanded by the LS command.

LG Output Messages

Message	Possible Cause	Suggested Action
"LG" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter LG from a controlling SPO device.
SYSTEM LOGGING COMMENCED	Logging was successfully started.	None.
SYSTEM LOGGING ALREADY IN OPERATION	LG was issued while logging was already on.	None.

LS (Stop system message logging)

Syntax:



This intrinsic stops the logging of system messages. If logging was initiated by the LG command, error message logging is also stopped. Logging remains stopped until either the system is powered off (at the next warmstart the default option set using CONFIGURER will be acted on), or LS is countermanded by the LG command.

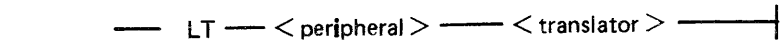
LS Output Messages

Message	Possible Cause	Suggested Action
"LS" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND.	This intrinsic can only be entered from the controlling SPO.	Re-enter LS from a controlling SPO device.
SYSTEM LOGGING STOPPED	Logging was successfully stopped.	None.
SYSTEM LOGGING ALREADY STOPPED	LS was issued while logging was already OFF	None.

LT (Load Translation table, only implemented on B 90)

B 1000 users see section 10 of this manual for LT utility.

Syntax:



This intrinsic builds a logical connection between the specified device and the translation table, such that data passing to/from the device is subject to translation as defined by the translation table.

Only local keyboard, printers that are not connected via datacomm, CRT and SELF SCAN can have translation specified.

This intrinsic causes the translation table, specified in the file SYSTRANS, to be used and stores appropriate information in the file SYSCONFIG so that it is remembered through system power off and warmstart. This only occurs if the device has already had a translation table assigned, when CONFIGURER was executed.

The translator name is a character string of up to 20 characters which must have the following format:

device.style.language.version

for example: SP.B932.UK.V2

For details, refer to SYSTRANS format in section 8.

Example:
To load and assign the German version of the translator to the line printer B 9249-30, the following syntax is used:

```
spo input      LT LPA LP.B924930.GRMNY.V5
spo output     TRANSLATION TABLE LOADED FOR DEVICE LPA
```

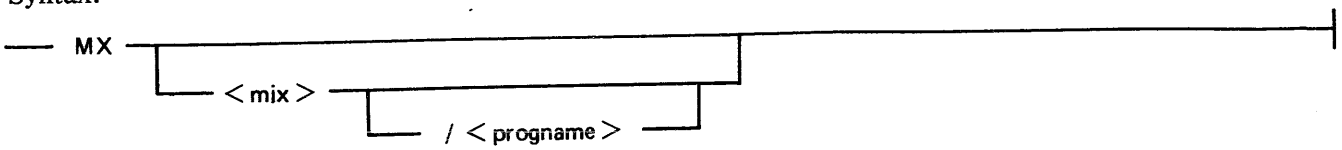
LT Output Messages

Message	Possible Cause	Suggested Action
LT faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE	The device is in use, or is invalid. Also the input to LT may be incorrect.	Check input and that the device is online and ready, before re-entering LT.
LT faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The specified device is not present.	Check input and ensure the device required is connected to the system and is ready.
"LT" IGNORED – TRANSLATION FILE NOT AVAILABLE. COPY SYSTRANS FILE TO	The system file SYSTRANS, which contains the translation tables is not present on the system disk.	Copy the file SYSTRANS to the system disk and re-enter LT.

Message	(continued) Possible Cause	Suggested Action
SYSTEM DISK AND RETRY COMMAND		
"LT" IGNORED – SPECIFIED TRANSLATION TABLE NOT FOUND IN TRANSLATION FILE. SPECIFY ANOTHER TABLE OR ADD NEW TABLE TO SYSTRANS FILE	The required translation table is not present in SYSTRANS.	Check input to ensure that the correct translation table was requested.
"LT" INVALID – SPECIFIED TRANSLATION TABLE NOT APPLICABLE TO DEVICE peripheral. RETRY COMMAND USING DIFFERENT TABLE	The file SYSTRANS on the system disk is incompatible with this system.	Copy the correct version of SYSTRANS onto the system disk and then re-enter LT.
SPECIFIED TRANSLATION TABLE WILL NOT BE RE-LOADED AT NEXT SYSTEM POWER UP SINCE NO SYSCONFIG TABLE ENTRY EXISTS. RUN "CONFIGURER" TO CREATE TRANSLATION ENTRY FOR peripheral AND RETRY COMMAND	This device did not have a translation table assigned to it in the system file SYSCONFIG.	Execute CONFIGURER utility and assign the table to the device.
"LT" IGNORED – HARDWARE ERROR ON SYSTEM DISK WHILST ACCESSING SYSTRANS FILE. RUN "CHECK.DISK" TO LOCATE ERROR	A hardware error has occurred while accessing thE system disk.	Run CHECK.DISK to locate error.
"LT" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be used from the system SPO.	Re-enter LT command from system SPO device.

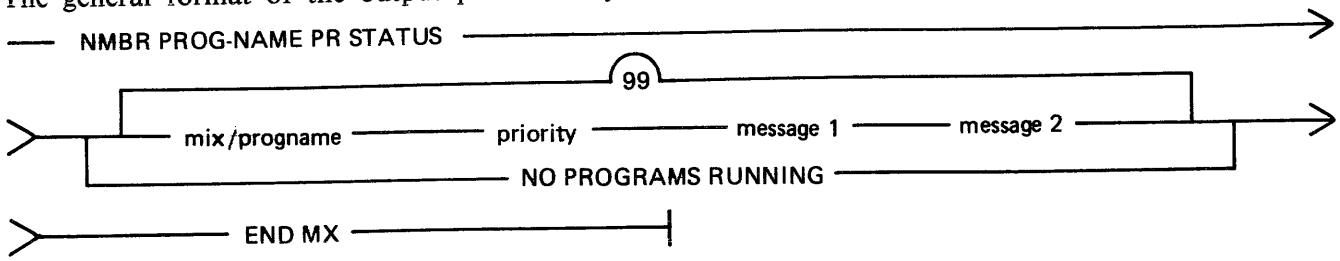
MX (Display Current Mix)

Syntax:



This intrinsic allows the operator to inquire about the state of any program(s) currently processing.

The general format of the output produced by the MX intrinsic is:

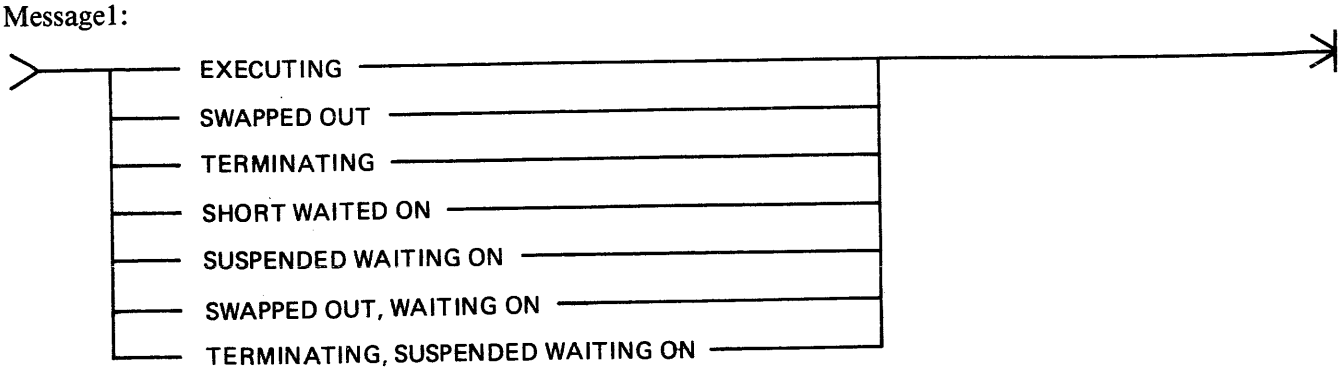


Where priority:

Where :

A – lowest priority (application programs)
B – medium priority (system programs, for example, utilities)
C – highest priority (programs using data communications)

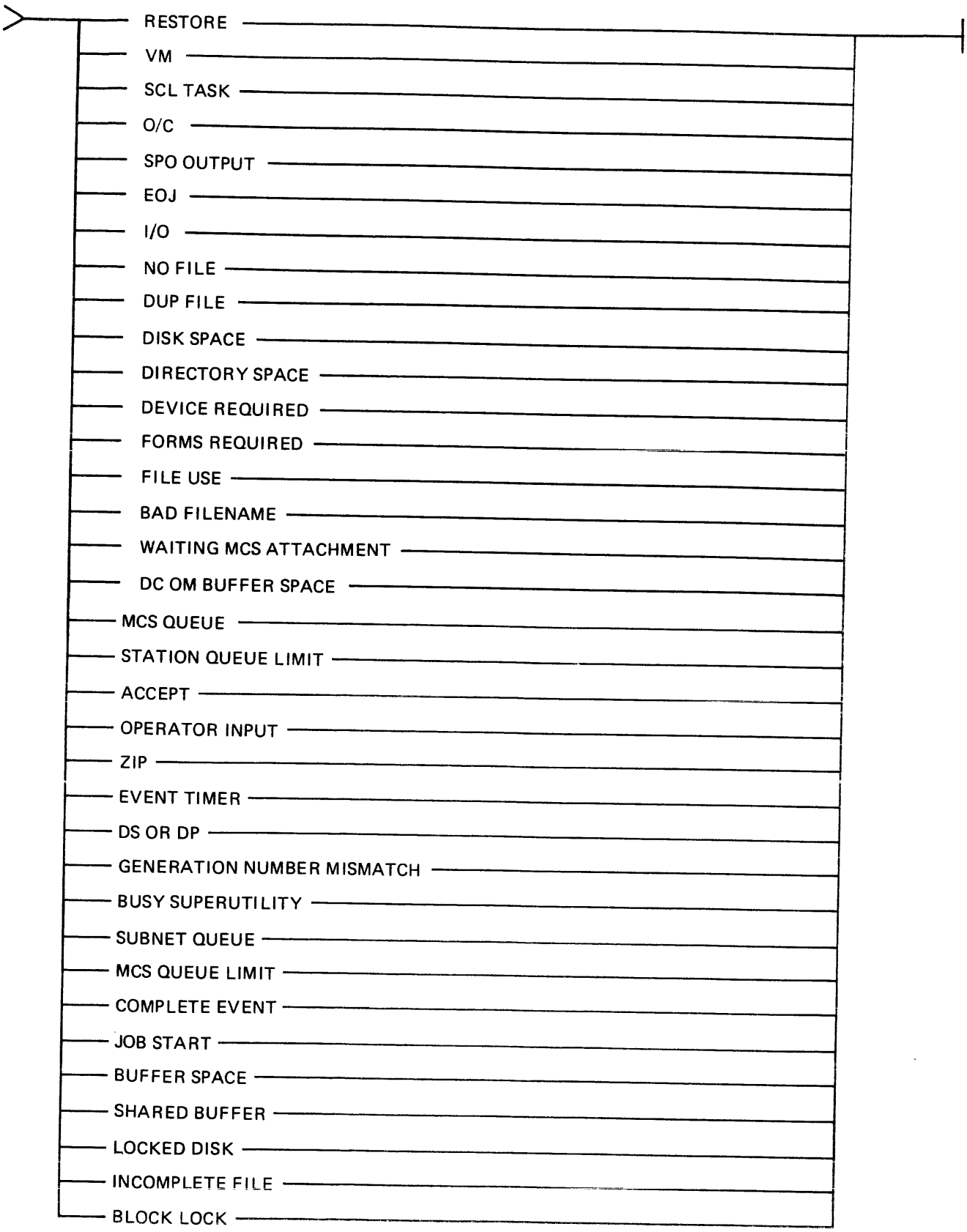
Message1:



Where:

EXECUTING	– the program is processing normally.
SWAPPED OUT	– the program's work area has been transferred to disk.
TERMINATING	– the program is going to end the job.
SHORT WAITED ON	– the program is waiting on the reason stated in message2.
SUSPENDED WAITING ON	– the program has been temporarily halted, waiting on the reason stated in message2.
SWAPPED OUT, WAITING ON	– the program's work area has been transferred to disk to allow other programs to use the work area while the program waits for the reason stated in message2.
TERMINATING, SUSPENDED WAITING ON	– the program will finish once the resource stated in message2 becomes available.

message2:


RESTORE
VM
SCL TASK
O/C
SPO OUTPUT
EOJ
I/O
NO FILE
DUP FILE
DISK SPACE
DIRECTORY SPACE
DEVICE REQUIRED
FORMS REQUIRED
FILE USE
BAD FILENAME
WAITING MCS ATTACHMENT
DC OM BUFFER SPACE
MCS QUEUE
STATION QUEUE LIMIT
ACCEPT
OPERATOR INPUT
ZIP
EVENT TIMER
DS OR DP
GENERATION NUMBER MISMATCH
BUSY SUPERUTILITY
SUBNET QUEUE
MCS QUEUE LIMIT
COMPLETE EVENT
JOB START
BUFFER SPACE
SHARED BUFFER
LOCKED DISK
INCOMPLETE FILE
BLOCK LOCK

Where: Message	Meaning	Suggested Action
RESTORE VM	The program is waiting for Virtual Memory.	None: Do not try to execute too many programs at this time.
SCL TASK	The program is waiting for an MCP intrinsic to be completed (such as the response to an OL).	None: The program will resume execution when the SCL task has finished.
O/C	The program is waiting for a file to be opened or closed.	None. The program will resume execution when the required file has opened or closed.
SPO OUTPUT		
EOJ	The program has gone to end of job.	None.
I/O	Usually indicative of normal processing, involving input/ output activity to a peripheral.	None.
NO FILE	The program requires a particular file, which has not been found, to continue processing.	Check SPO for message indicating name of the file required and make the file available. If in doubt refer to operating instructions for the program.
DUP FILE	The program is attempting to place a file of a certain name on the disk. However, another file by the same name currently resides on the disk. A disk may not contain two files with the same name.	Either remove the existing file with RM or change its name with CH. If in doubt refer to the program instructions.
DISK SPACE	The available space on the disk is insufficient to hold the file the program is attempting to write; or the disk is "checkerboarded".	With the KA utility analyse the amount of space remaining. If the disk is full, remove any unnecessary files with RM. If the disk is checkerboarded, DS the program then use the SQ utility to consolidate disk space. Re-execute the suspended program.
DIRECTORY SPACE	When the disk was initialized, the directory was constructed to contain a fixed number of file names. The directory has now reached its capacity.	Remove with RM any unnecessary files and the program will continue; or DS the suspended program and replace the disk with another having sufficient directory space, and re-execute the program.
DEVICE REQUIRED	The program requires a particular device to continue.	Make the device physically and logically ready by using the RY intrinsic. Or assign an alternative

Message	Meaning	Suggested Action
		device to the program using the AD intrinsic.
FORMS REQUIRED	The program requires a printer with special stationery.	Ensure that a printer containing the required stationery is ready, then assign the program to that printer using the AD intrinsic.
FILE USE	The program requires access to a disk file which is already being used by the maximum number of co-existent users.	None. The program will resume execution when one of the other users closes the file, allowing this user (program) to open the file.
BAD FILENAME		
WAITING MCS ATTACHMENT		
DCOM BUFFER SPACE		
MCS QUEUE		
STATION QUEUE LIMIT		
ACCEPT	The program has displayed an "ACPT" message on the SPO device and is waiting for an appropriate response.	Refer to the program's operating instructions for the required response. Then respond using the AX intrinsic.
OPERATOR INPUT	The program is waiting for some input from the operator. For example a program previously suspended by the ST intrinsic.	Provide the program with the appropriate input. The program will continue executing. For the example given, the GO intrinsic would be used.
ZIP	The program has initiated the execution of another program by means of a ZIP communicate and is now waiting for that program to finish.	None.
EVENT TIMER	The program has issued a wait for a finite number of seconds.	None. The program will resume when the time interval has expired.
DS OR DP	The program has performed an illegal act.	Use the DP command to stop program and dump its workspace. Copy the dump file, program and the current SYS-LOG-file to removable media. Send these files along with an explanation to the person responsible for the program. Or if this is a known problem then use the DS intrinsic to terminate the program.

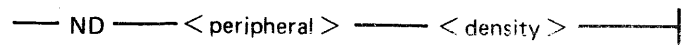
Message	(continued) Meaning	Suggested Action
GENERATION NUMBER MISMATCH		
BUSY SUPERUTILITY	The program requires to use a function which is part of the SYS-SUPERUTL utility. SYS-SUPERUTL is already performing a function.	None.
SUBNET QUEUE		
MCS QUEUE LIMIT		
COMPLEX EVENT	The program has issued a complex wait communicate and is waiting for one or more of the events specified by the communicate to finish.	None.
JOB START		
BUFFER SPACE		
SHARED BUFFER		
LOCKED DISK	Another program has opened the SYSMEM file on a disk to which the suspended program requires access.	None. The program will resume once the the SYSMEM file has been closed.
INCOMPLETE FILE		
BLOCK LOCK	When using shared files, the program has output capability on a shared file and is trying to access a record in a block of that file that has been read by another output capable program.	None. The program will resume once the other program has updated the record or accesses a record outwith that block.

MX Output Messages

Message	Possible Causes	Suggested Action
MX faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX without mix number.
MX faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX without the mix number and the program name.

ND (New Density, only implemented on B 90)

Syntax:



This intrinsic allows the operator to define the print density on suitable console printers and soft switchable CRTs. For printer devices, the density field is a single character which specifies the density as follows:

A. the greatest number of characters per inch available on the printer
B. the second highest number of characters per inch and so on.

1. the greatest number of lines per inch available on the printer.
2. the second highest number of lines per inch, and so on.

For printer devices, when the number of characters per inch changes, the operating system will adjust the values of page width and offset (previously set by a FD command or by default), so that any subsequent output is restricted to the part of the platen available with the previous density. Similarly, page height will be adjusted when the number of lines per inch changes. The new density together with the adjusted values of page height or page width and offset are recorded in a system configuration file and are remembered through system shutdown and warmstart.

For screen devices, the density field is a single character which specifies the density as follows :

A. the greatest number of characters available on the screen.
B. the second greatest number of characters available and so on.

On screen devices, the ND command applies only to the character size used for console file use. If the screen is being used in SPO/Console mode, the SPO mode will have the greatest number of characters available. At warmstart, if any inconsistency is found between the system configuration file information (ND and FD values for printers) and the capability of the current device, then the system configuration file information is ignored and default values used. The system configuration file is unaltered.

The densities available on different printers are as follows:

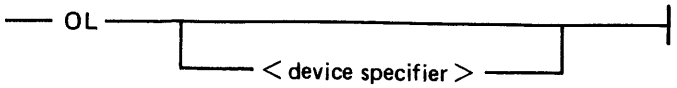
Peripheral	Density	Value
120 cps console printer	15 characters per inch	A
	10 characters per inch	B
90 cps console printer	16 2/3 characters per inch	A
	12 1/2 characters per inch	B
	10 characters per inch	C
	8 lines per inch	1
	6 lines per inch	2

ND Output Messages

Message	Possible Causes	Suggested Action
ND faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE.	1. The specified value of density exceeds the number of options available on the device.	1. Correct and re-enter ND.
	2. The device is in use.	2. Wait until the program using the device has finished with it and then re-enter ND.
	3. The peripheral specified is unsuitable.	3. Check input.
ND faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The specified peripheral is not connected to the system.	Check input and ensure the required peripheral is connected to the system before re-entering ND.
"ND" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter ND from the controlling SPO device.
ND faulty input INVALID – PARAMETERS ILLEGAL OR OUT OF RANGE. REFER TO SOG FOR ALLOWABLE FORMAT	The ND specifications are not acceptable.	Check input and re-enter ND command.

OL (Request Status Information of Peripherals

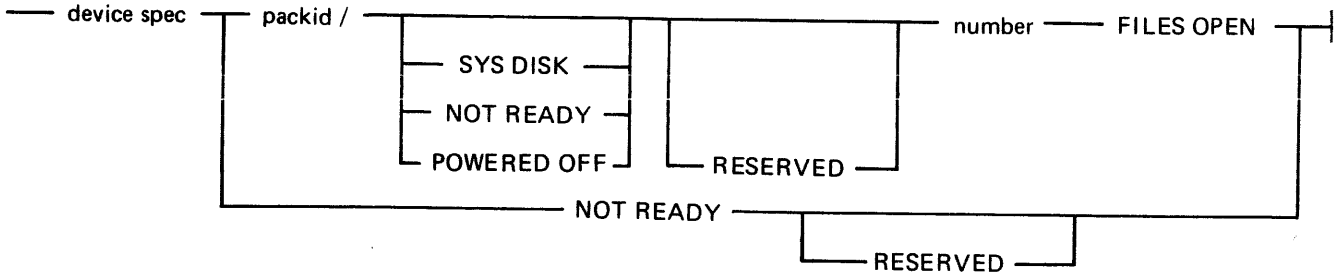
Syntax:



This intrinsic allows the operator to request the status of peripherals on the system. If a specific device is named (for example CTA) in the command, only the status for that peripheral, if any, is given. If the device specifier names a device family, then the status for all devices in that family is produced. If the device specifier is omitted, the status of all devices on the system is given.

The system responses vary depending on whether the device in question is a disk, magnetic tape, data communications, clock, or other device.

Response for a disk device:

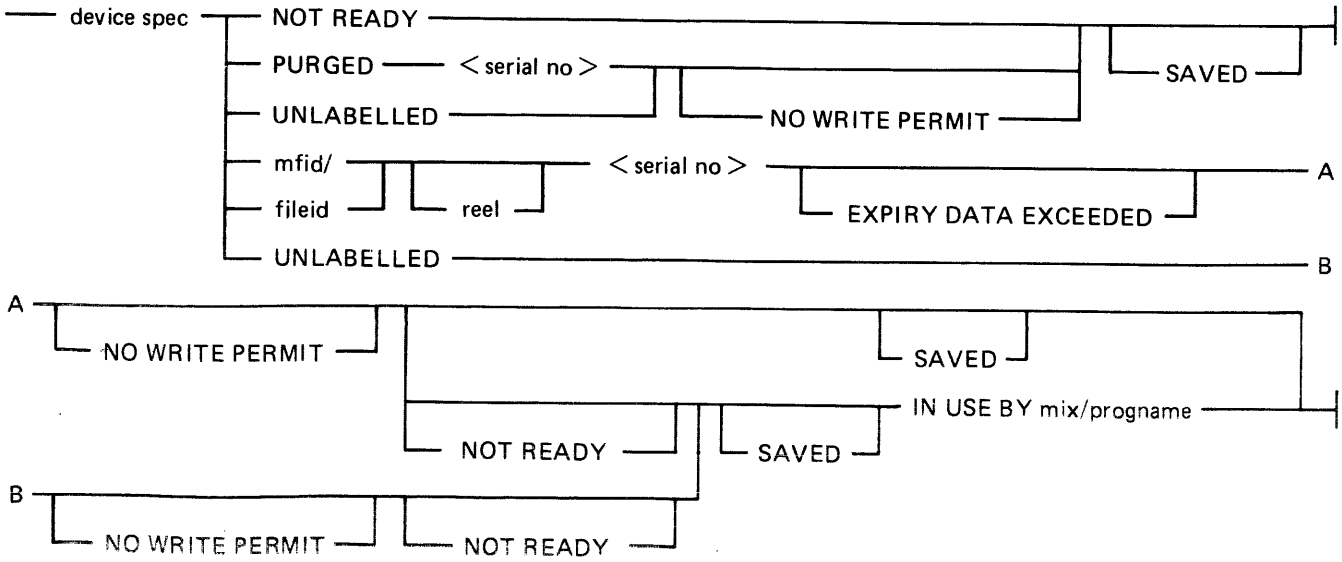


In addition, if a fixed disk assemblage is the system disk, the status of "system disk" applies to all the fixed disk units. The system response for each fixed disk unit will not contain the phrase "SYS DISK", but the last line of the response will be

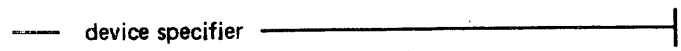
"device type mnemonic = SYS DISK"

indicating that the fixed disk assemblage is the system disk.

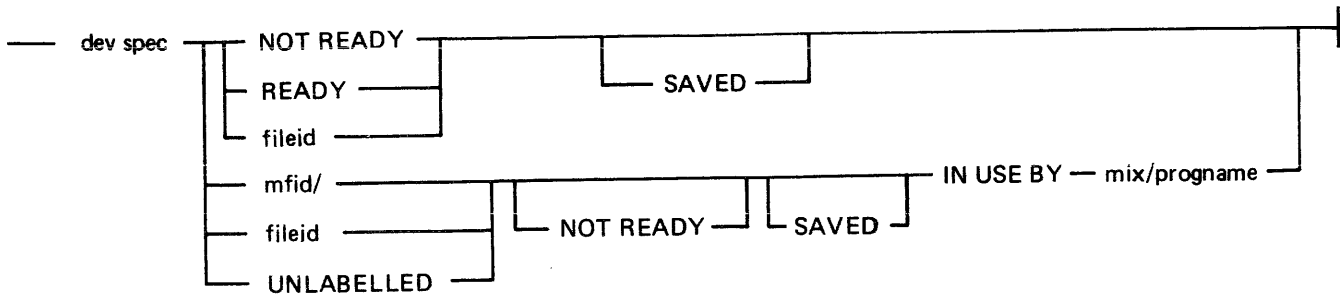
response for magnetic tape:



response for data communication and clock devices:



response for other devices:



Examples:

Example 1.

To find the status of all Burroughs Mini disks (DM) on the system :

```
spo input      OL DM
spo output     DMA C34B90B/  NOT READY 0 FILES OPEN
spo output     DMB NOT READY
spo output     DMC RESERVED 0 FILES OPEN
spo output     DMD NOT READY RESERVED 0 FILES OPEN
spo output     END OL
```

Example 2.

To find the status of line printer LPA :

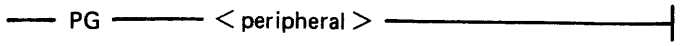
```
spo input      OL LPA
spo output     LPA PRINTOOT IN USE BY 10/LIST
spo output     END OL
```

OL Output Messages

Message	Possible Causes	Suggested Action
OL faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS	A non-existent device has been specified.	Check input and re-input if necessary.
OL faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The device or family specified is not connected to the system.	None.

PG (Purge Tape)

Syntax:



This intrinsic allows the operator to purge (erase) magnetic and cassette tapes, providing the tape is write enabled. This labels the tape as available for output use.

Examples:

Example 1.

To purge a cassette on drive CTA:

spo input	PG CTA
spo output	CTA PURGED 12345

Example 2.

To purge a magnetic tape on drive MTA:

spo input	PG MTA
spo output	MTA PURGED ?????

NOTE

In example 1 the numbers 12345 were produced, this is the original serial number of the tape. In example 2 the question marks are output as this is a new tape with no serial number on it.

PG Output Messages

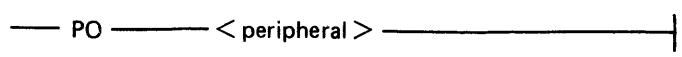
Message	Possible Causes	Suggested Action
PG faulty input IGNORED – UNABLE TO PURGE WRITE-PROTECTED TAPE	The tape is not write enabled.	Write enable the tape and re-enter PG command.
PG faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE.	The device is in use.	Wait until the device is free or use another tape device. Re-enter PG command.
PG faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS.	The device specified is illegal.	Check and correct input and re-enter PG command.
PG faulty input INVALID – DEVICE NOT ON SYSTEM.	The tape device specified is not connected to the system.	Use OL to find tape device connected to the system. If a

(continued)

Message	Possible Causes	Suggested Action
ENTER "OL" FOR MORE DETAILS		suitable device is present, use it. Re-enter PG command.
PG faulty input IGNORED – HARDWARE ERROR ON peripheral. IF DISK, RUN "CHECK.DISK". IF TAPE, REPLACE MEDIA	An Input/Output error has been detected on the tape device. For example, a parity error.	Use another tape and re-enter PG command. If this message is repeated with the same drive, seek technical assistance.
"PG" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter PG from the controlling SPO device.

PO (Power Off a disk)

Syntax:



This intrinsic is applicable to disk devices only. It allows the operator to "logically" power off a disk (instruct the MCP that the disk is no longer required), before it is physically powered off. The system may be powered off when there are no programs in the mix. For a system running with a data comm SPO (SCL) device then NDL, MCS and SPIM can be present in the mix. This causes the MCP to terminate. All system disk files will be closed and SYS-SUPERUTL will go to End of Job (EOJ).

Do not remove a disk from the disk drive, power down the disk unit, or switch off the main cabinet until disks have been logically powered off with PO and the number of files open on the disk is zero (0). Failure to observe this practice might cause disk problems at a later date.

Examples:

Example 1.

To power off the disk named DOCF01 in drive DMB

```
spo input      PO DMB
spo output     DISK DOCF01/  IN DRIVE DMB NOT IN USE
spo output     DISK (IF NOT FIXED) MAY NOW BE REMOVED
```

Example 2.

To power off the disk in drive DKB

```
spo input      PO DKB
spo output     DKB D34B8/  POWERED OFF 2 FILES OPEN
```

At this point the cartridge cannot be removed as there are still files open. The operator must wait for a message indicating that there are no files open (that is zero files open):

```
spo output DKA D34B8/NOT READY 0 FILES OPEN
```

PO Output Messages

Message	Possible Causes	Suggested Action
DISK packid/ IN DRIVE peripheral NOT IN USE. DISK (IF NOT FIXED) MAY NOW BE REMOVED	The disk has been logically powered off and can be removed or physically powered off.	None.
DISK packid/ IN DRIVE peripheral HAS number FILES OPEN AND IS STILL IN USE. A FURTHER MESSAGE WILL	The disk has been marked as logically powered off.	The operator must wait until the number of files open on the disk is zero (0) before the disk is removed or physically powered off.

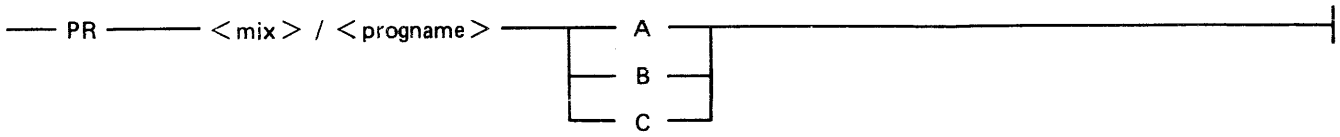
(continued)

Message	Possible Causes	Suggested Action
INDICATE WHEN DISK NO LONGER IN USE		
PO faulty input IGNORED – DRIVE NOT READY	The device is not ready.	None.
PO faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS	The device specified is illegal.	Use OL to find the correct device and re-enter PO command.
PO faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The device is not connected to the system	Use OL to find the correct device and re-enter PO command.
CLOSEDOWN MIX (SEE SOG FOR DEFINITION) MUST EXIST BEFORE SHUTTING DOWN SYSTEM. ENTER "MX" FOR DETAILS. TERMINATE NON-CLOSEDOWN PROGRAMS AND TRY "PO" AGAIN	The system disk is powered off until the mix is empty. That is, no programs running.	Ensure that all programs have gone to End of Job (note: SPIM, NDL, and MCS can remain in the mix of a datacomm SPO system), then re-enter PO command.
CLOSEDOWN MIX (SEE SOG FOR DEFINITION) MUST EXIST BEFORE SHUTTING DOWN SYSTEM. ENTER "MX" FOR DETAILS. TERMINATE NON-CLOSEDOWN DATA COMM PROGRAMS AND TRY "PO" AGAIN	The system is running data comm and the SPO (SCL) is not being run from the data comm.	Halt the data comm and then re-enter PO command.
HARDWARE ERROR ON DISK peripheral DURING "PO". DISK NOT POWERED OFF. RUN "CHECK.DISK" TO DETERMINE FAULT	An Input/Output error has occurred on the drive.	Re-enter PO command. If the error continues, seek technical assistance.
"PO" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter PO from the controlling SPO device.
DISK packid/ IN DRIVE peripheral GONE NOT READY WHILST NOT IN USE WITHOUT "PO"	The user disk has been removed without using the PO intrinsic.	Re-load the disk into the same drive and use the PO intrinsic to logically power off the drive and wait for the system to respond

Message	(continued) Possible Causes	Suggested Action
BEING ISSUED. RELOAD DISK		that the disk may be removed.
If there are any files left open on the disk then this message appears: DISK packid/ IN DRIVE peripheral GONE NOT READY WHILST IN USE. RELOAD DISK		

PR (Assign Program Priority, not implemented on B 90 systems)

Syntax:



This intrinsic allows the operator to alter the priority of a program by moving it to the highest priority position in the class specified.

A This is low or normal priority, used for regular work. Within this class, programs which perform more physical Input/Output operations are given precedence.

B This is medium priority, used for utilities or programs which may be expected to do emergency work. The priority within this class is reverse historical; that is a program of this priority placed in the mix will take precedence over previous programs of the same priority.

C This is high priority, used for data communications programs that are transaction-driven. These are normally dormant, awaiting a transaction, but when required to process a transaction they take high priority to minimize response times. Within this class programs which do more physical Input/Output are given precedence.

Example:

To change the priority of mix number 3 (program REP506) to B:

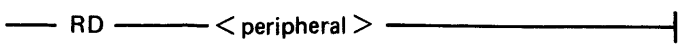
```
spo input      PR 3/REP506 B
spo output     PRIORITY OF 03/REP506 PR CHANGED TO B
```

PR Output Messages

Message	Possible Causes	Suggested Action
PR faulty input INCOMPLETE – BOTH MIX NUMBER AND PROGRAM NAME MUST BE SPECIFIED. ENTER "MX" FOR MORE DETAILS	Mix number and program name do not match or do not exist. Or the priority value is incorrect.	Check with MX for proper input and re-enter PR command.
PR faulty input INVALID – PARAMETERS ILLEGAL OR OUT OF RANGE. REFER TO SOG FOR ALLOWABLE FORMAT	The PR specifications are not acceptable.	Check input and re-enter PR command.

RD (Reserve Disk Drive)

Syntax:



This intrinsic applies only to disks and may not be applied to the system disk.

This intrinsic allows the operator to reserve a peripheral. If the peripheral is not physically ready when the RD is performed, it is marked. When it is subsequently made ready, physically powered on, no AVR will be performed but the device will be marked physically ready and reserved.

If the peripheral is physically ready when RD is issued, the device is marked reserved and no further labelled opens are allowed. Any jobs using the device are allowed to terminate normally.

A reserved device may be "unreserved" by use of the RY intrinsic.

Examples

Example 1.

If the disk in drive DMA is to be reserved, the following should be entered:

```
spo input       RD DMA
spo output      DMA RESERVED 0 FILES OPEN
```

If there was no disk in the drive then the following would happen:

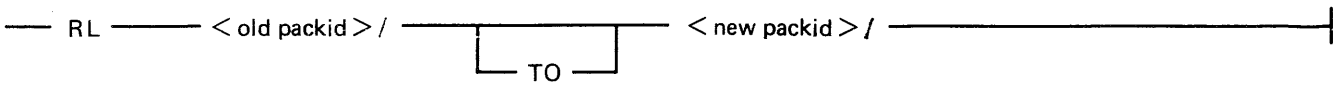
```
spo input       RD DMA
spo output      DMA NOT READY RESERVED
```

RD Output Messages

Message	Possible Causes	Suggested Action
RD faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS	The specified peripheral is either not a disk, is the system disk, or is illegal.	Check input and re-enter reserve command.
RD faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The specified disk is not connected to the system.	Check input and ensure required drive is connected to the system then re-enter reserve command.
"RD" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter RD from the controlling SPO device.

RL (Relabel Disk/Pack)

Syntax:



The RL intrinsic allows the operator to change the disk name, packid and pseudo-packids that are present on the system.

Three conditions are required for this command to be accepted:

1. If a pseudo-packid is being relabelled, all physical units that have packids listed in the PPIT must be physically ready.

2. If the disk to be relabelled is not a system disk, there must be no files open on the disk.

3. If the disk to be relabelled is a system disk, a null mix must exist (no programs running). On a remote SPO (SCL) system the only programs running should be SPIM, the MCS that is managing the remote system (usually MCSOBJ) and NDLSYS.

When the disk requested to be changed is a pseudo-pack, this command causes the entry for the packid in the pseudo-pack identifier table to be changed to the new packid. When the packid requested to be changed identifies a physical disk, this command also causes the packid in the disk label field of the designated physical unit to be updated with the new disk name and assigns the unit to the system with this new disk name.

WARNING

RL will not correct disk file headers of dual pack file components or Key File Parameter Blocks (KFPB). That is, a key file on the disk that is linked to a data file on the same disk will not have its KFPB corrected to point to the new name of the disk.

Example:

To change the diskname DCO286 to DOCOLD:

```
spo input      RL DOC286/ TO DOCOLD/
spo output     DOC286/ RELABELLED AS DOCOLD/
```

RL Output Messages

Message	Possible Causes	Suggested Action
"RL" IGNORED – DISK packid/NOT PRESENT. POWER ON DISK AND RETRY COMMAND	The specified disk is not present on the system, or is associated with a reserved disk.	Use OL to find if the disk required is present and the name of the disk. Re-enter RL command.
"RL" IGNORED – DISK packid/ ALREADY ON SYSTEM. REMOVE/ RELABEL DUPLICATE DISK OR CHOOSE DIFFERENT DISK NAME AND RETRY COMMAND	A duplicate pack condition has been detected.	Remove the duplicate pack and re-enter RL command.

(continued)

Message	Possible Causes	Suggested Action
RL faulty input IGNORED – ALL DISKS BELONGING TO FIXED DISK ASSEMBLAGE MUST BE ONLINE BEFORE COMMAND ISSUED. ENSURE DISKS ARE POWERED ON	At least one physical unit from the PPIT (pseudo-pack only is not ready.	Ready the units and re-enter RL command.
RL faulty input INVALID – DISK NAME INCORRECTLY FORMED. REFER TO SOG FOR ALLOWABLE FORMAT	The disk name specified is not in CMS standard format.	Check input and re-enter RL command.
"RL" IGNORED – CANNOT RELABEL DISK AT THIS TIME. TRY AGAIN LATER	There are programs running on the system disk, other than those required for remote SPO (SCL) operation.	Wait until a null mix occurs, then re-enter RL command.
RL faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE.	There are files open on the disk to be relabelled.	Wait until there are no files open on the disk before re-entering RL command.
RL faulty input IGNORED – HARDWARE ERROR ON peripheral. IF DISK, RUN "CHECK.DISK". IF TAPE, REPLACE MEDIA	A hardware error prevented the command being executed.	Re-enter RL command. If this condition continues to occur, seek technical assistance.
"RL" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter RL from the controlling SPO device.

RY (Ready Device)

Syntax:

This intrinsic is used to "READY" a peripheral so the MCP can use it as a resource. When warm-starting, the system will automatically ready all peripherals on the system that are powered on. RY may also be used to ready a previously powered off (PO) user disk, a peripheral that is saved (SV) and disk that has been reserved (RD).

When this intrinsic is applied to a fixed disk assemblage, it applies to all the fixed disk units. The peripheral field must be specified as 'device type mnemonic=' (for example DF=) and a request to ready a single fixed unit will be rejected. When the intrinsic is accepted, all fixed disk units, except those which are physically powered off, will be readied.

NOTE

For B 9252 line printers on B 90 systems, the printer normally remains logically ready at all times. If the printer does go logically not ready, the job using the printer is suspended. If a VF command is sent to the printer from the SPO and the printer is logically not ready, the SPO hangs. Recovery is possible in both these cases by powering the printer off and then on.

Examples:

Example 1.

To ready the line printer LPB, enter the following:

```
spo input       RY LPB
spo output      LPB READY
```

Example 2.

To ready the magnetic tape on drive MTB, enter the following:

```
spo input       RY MTB
spo output      MTB NOT READY
```

In this case the drive is not physically ready. This could be because there is no tape in the drive, the drive is not powered on, or the drive failed to recognize the tape for some reason.

Example 3.

To ready the disk drive DMA, enter the following:

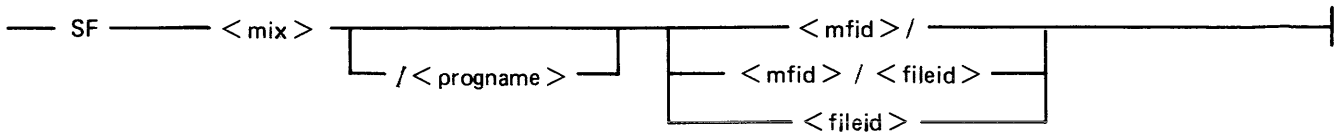
```
spo input       RY DMA
spo output      DMA DOC286/  0 FILES OPEN
```

RY Output Messages

Message	Possible Causes	Suggested Action
RY faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The peripheral specified is not connected to the system.	Check input and ensure the required device is connected to the system. Re-enter RY.
RY faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS	The peripheral specified is invalid.	Check input then re-enter correct peripheral mnemonic in RY.
"RY" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the SPO.	Re-enter RY command from the system SPO device.

SF Substitute File (Not implemented on B 90)

Syntax:



This intrinsic is valid only for disks.

The substitute file intrinsic allows the operator to direct a program to a particular disk file if it is waiting on a "NO FILE", "NO PACK", "DUPLICATE FILE" or "BAD FILE NAME" condition.

This command causes temporary modification to the program's file parameter block (FPB). The modification remains in effect for the current execution only, or until it is remodified by the program during execution.

This intrinsic must not be applied to printer backup files. AD should always be used.

The operator may not anticipate a program's needs and pre-modify an FPB that might cause a suspension in the future.

Examples:

Example 1.

Program AP10 (mix number 01) requests a disk file called APD2T on disk APD. To direct the program to use the file APD2A on the same disk:

```
spo output      01/AP10 <10> WAITING APD/APD2T DK ON FILE
spo input       SF 01/AP10 APD2A
```

Example 2.

To direct the same program in example 1 to use file APD2T on the disk called APD1, input one of the following:

```
spo input       SF 01/APD10 APD1/
                or
spo input       SF 01 APD1/
```

Example 3.

To direct the same program in example 1 to use file ATEMP on the disk called ARTD, input the following:

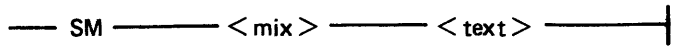
```
spo input SF 01 ARTD/ATEMP
```

SF Output Messages

Message	Possible Causes	Suggested Action
SF faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program requiring device assignment. Re-enter SF.
SF faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program requiring device assignment. Re-enter SF.
mix/progname NOT IN SUITABLE STATE TO ACCEPT "SF". ENTER "MX" FOR MORE DETAILS	Specified program was not suspended waiting.	Check with MX for mix number

SM (Multi-MCS data communications Send Message)

Syntax:



This intrinsic allows the operator to input a message to a multi-MCS task. The text will be formatted by the data comm subsystem and inserted into its MCS input queue. The text may contain carriage return, line feed and bell characters which are actioned by the MCP and left in the character string passed to the data comm subsystem.

Example:

To enter the text "RY S 0" to the MCS (usually MSCOBJ) in the mix slot 15, input the following:

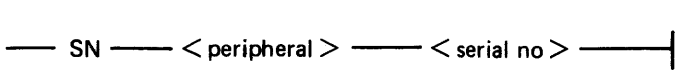
```
spo input     SM 15 RY S 0
spo output    FROM MCS: RY OK (126)
              spo output FROM MCS: STATION 0/AP300XA31 READY (212)
```

SM Output Messages:

Message	Possible Causes	Suggested Action
"SM" IGNORED – NO MCS RUNNING WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	There is no MCS having the specified mix number.	Use MX to find mix number of the required MCS. Re-enter SM command.
"SM" INVALID – MIX NUMBER OF MCS MUST BE SPECIFIED. ENTER "MX' FOR MORE DETAILS	No mix number was specified.	Re-enter SM command with correct mix number.
"SM" IGNORED – NO DATACOMM BUFFER SPACE AVAILABLE FOR MCS MESSAGE. TRY AGAIN LATER	There is no DC message space, in which to transfer the message text.	Wait a short time, then re-enter message. If unsuccessful after several attempts, seek technical assistance.
"SM" INVALID – NO MCS PROGRAM RUNNING	The MCS is not running.	Check input. Execute the MCS if required. Re-enter SM.
"SM" IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The MCS was not executed from this SPO device.	Ensure that the SPO device used is the one that initiated the MCS, or use the SYSTEM SPO device.

SN (assign a Serial Number to a tape)

Syntax:



This intrinsic is valid for magnetic tape or cassette devices only. It causes the tape to be purged and the purge label written to the tape (in a manner similar to the PG intrinsic) will contain the specified serial number.

The serial number is specified as up to five ASCII characters and will appear in all labels written to the tape. The serial number remains on the tape when purged. The only way to change the serial number is by another explicit SN command.

The tape may now be written to.

Ensure that the magnetic tape is write enabled (a write ring is in place on the tape), or that the cassette is write enabled (the write tabs on the cassette are set for write enable). The drive must also be ready and "on line" to the system.

Examples:

Example 1.

To purge a cassette on drive CTB and assign the serial number 12345, input the following:

spo input	SN CTB 12345
spo output	CTB PURGED 12345

Example 2.

To purge a magnetic tape on drive MTA and assign the serial number 345, input the following:

spo input	SN MTA 345
spo output	MTA PURGED 345

SN Output Messages

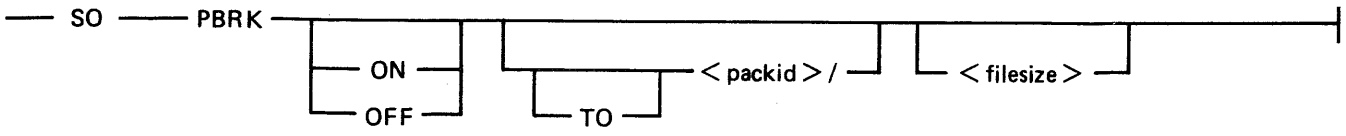
Message	Possible Causes	Suggested Action
SN faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The specified device is not connected to the system.	Check input and ensure with OL that required drive is connected to the system, then re-enter SN.
SN faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE	The specified peripheral is found to be in use.	Check input and re-enter SN.

(continued)

Message	Possible Causes	Suggested Action
SN faulty input FAILED – I/O ERROR ON peripheral. IF DISK, RUN "CHECK.DISK". IF TAPE, REPLACE MEDIA	A hardware error prevented the command from being executed.	Replace media and try again. If the problem continues to occur, seek technical assistance.
SN faulty input IGNORED – UNABLE TO PURGE WRITE-PROTECTED TAPE	The peripheral specified in an SN command is found to be write inhibited.	Ensure there is write enabled media loaded, then re-enter SN.
SN faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS.	An illegal peripheral was specified in an FD command.	Check input and re-enter FD command.
SN faulty input IGNORED – DEVICE ERROR event no. ON PERIPHERAL – THE REQUESTED TRANSLATION TABLE IN FILE SYSTRANSLATE CANNOT BE ACCESSED	A problem has been encountered in accessing the translation table for the specified peripheral.	Seek technical assistance.

SO PRBK (Set printer backup ON/OFF)

Syntax:



This intrinsic is used to change any or all of the following printer backup related options :

1. The ON/OFF option sets the system printer backup option to the requested state. This allows printer output to be directed to a printer or disk backup depending on the options selected by the program requiring the printer, and the availability of a printer. If neither option is specified, then the state of the system option is not changed.

2. The disk name option determines the disk on which all printer backup files are to reside. This disk must be on the system. This disk name will remain in force until changed by the "SO PRBK" intrinsic or the system is re-warmstarted (the designated printer backup disk in the file SYSCONFIG see CONFIGURER). The specification of this option also causes all Printer Back-up file opens waiting on a no disk condition to be restarted. If no disk name is given, the print-er backup designate disk name is unchanged.

3. The filesize option specifies the maximum file size to be used when programs open print files with their File Parameter Block maximum file size field set to zero. This value must be in the range 1 to 999999 and replaces the system default printer backup file size. It remains in force until changed by "SO PRBK" intrinsic or the system is re-warmstarted. When the system is warmstarted, the default value specified via CONFIGURER is used.

One of the three options must be present in the initiating message.

When the option(s) have been updated, the message:

SO PRBK option(s) COMPLETE
PRINTER BACKUP OPTION IS ON or OFF
PB DISK IS packid/ DEFAULT MAXIMUM FILESIZE IS filesize

is displayed.

Examples:

Example 1.

To enable the system printer backup option the following could be used:

```
spo input     SO PRBK ON
spo output    SO PRBK ON COMPLETE
spo output    PRINTER BACKUP OPTION IS ON
spo output    PB DISK IS SYSTEM/  DEFAULT MAXIMUM FILESIZE IS filesize
```

Example 2.

To enable the system printer backup option, specify the printer backup designate disk to be PBDISK (which is on line and ready) and specify the default printer backup filesize of 10000 records (to be used with programs that have a size of zero declared) the following syntax would be used:

```
spo input     SO PRBK ON TO PBDISK/ 10000
spo output    SO PRBK ON TO PBDISK/ 10000 COMPLETE
spo output    PRINTER BACKUP OPTION IS ON
spo output    PB DISK IS PBDISK/, DEFAULT MAXIMUM FILESIZE IS 10000
```

Example 3.

To disable the system printer backup option and set the default printer backup designate disk to the system, which is named SYSTEM, the following syntax would be used:

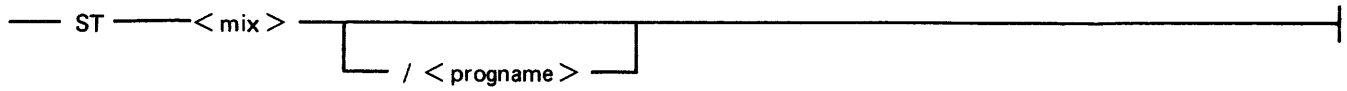
```
spo input     SO PRBK OFF TO SYSTEM/
spo output    SO PRBK OFF TO SYSTEM/ COMPLETE
spo output    PRINTER BACKUP OPTION IS OFF
spo output    PB DISK IS SYSTEM/ DEFAULT MAXIMUM FILESIZE IS 10000
```

SO PRBK Output Messages

Message	Possible Causes	Suggested Action
SO faulty input INVALID – AT LEAST ONE OPTION MUST BE SPECIFIED REFER TO SOG	The initiating message is not correct or none of the three possible options was specified.	Check input and re-enter SO.
SO faulty input IGNORED – DUPLICATE PRINTER BACKUP DISKS ONLINE POWER OFF ONE AND RETRY COMMAND	The disk specified in the SO command is a duplicate.	Power off one of the duplicate disks, then re-enter SO.
SO faulty input INVALID – BACKUP MAXIMUM FILESIZE MUST BE IN RANGE 1-999999	The filesize specified in the SO command is zero or is greater than 999999.	Re-enter SO with a valid number for the filesize.
SO faulty input IGNORED – SPECIFIED PRINTER BACKUP DISK NOT ONLINE RETRY COMMAND WHEN DISK ONLINE	The disk specified in the SO command is not present on the system.	Ensure the required disk is online and ready before re-entering SO.
SO faulty input IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter SO from the controlling SPO device.

ST (Temporarily Suspend a Running Program)

Syntax:



This intrinsic places a temporary halt on an executing program. The program still appears in the mix. The data needed to restart the program exactly where it was stopped may be transferred from memory and stored on disk. The memory that was being used by the "stopped" program will be made available if necessary to the MCP for other use. The GO intrinsic must be used to restart the program.

Examples:

Example 1.

To stop the program whose mix number is 3:

```
spo input       ST 3
spo output      03/TESTPROG STOPPED
```

Example 2.

To stop the program called OTHERPROG in mix slot 5:

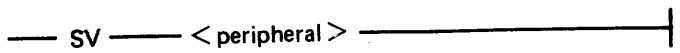
```
spo input       ST 5/OTHERPROG
spo output      05/OTHERPROG STOPPED
```

ST Output Messages

Message	Possible Cause	Suggested Action Action
ST faulty input INVALID – NO PROGRAM WITH THIS MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number does not correspond to a program in the mix.	Use MX to find the mix number of the program requiring device assignment. Re-enter ST.
ST faulty input INVALID – PROGRAM NAME DOES NOT MATCH MIX NUMBER. ENTER "MX" FOR MORE DETAILS	The mix number and program name do not match.	Use MX to find the mix number and the program name of the program requiring device assignment. Re-enter ST.
"ST" IGNORED – CANNOT USE COMMAND TO CONTROL PROGRAM INITIATED BY ANOTHER OPERATOR	The program was not executed from this SPO device.	Ensure that the device used is the one that initiated the program or use the SYSTEM SPO device.

SV (Save Peripheral)

Syntax:



This intrinsic is applicable to all devices except disks.

This intrinsic allows the operator to logically mark not ready a device in order to prevent its use by any program.

A "saved" device may be made available (ready) again with the RY intrinsic or by physically powering the unit off and then on again. If the device is being used by a program when it is saved, the program will be allowed to continue using the device, but any subsequent programs requiring the device will be prevented from using it. A saved device may be AD'ed regardless of its physical status.

Examples:

Example 1.

To save the serial printer SPA:

```
spo input       SV SPA
spo output      SPA SAVED
```

Example 2.

To save the line printer LPA which is being used by a program:

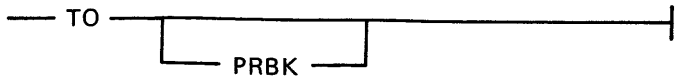
```
spo input       SV LPA
spo output      LPA SAVED IN USE BY 06/PR060
```

SV Output Messages

Message	Possible Causes	Suggested Action
SV faulty input INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND. REFER TO SOG FOR ALLOWABLE FORMATS	The device specified is either a disk or an illegal peripheral.	Check input and re-enter SV.
SV faulty input INVALID – DEVICE NOT ON SYSTEM. ENTER "OL" FOR MORE DETAILS	The specified peripheral is not connected to the system.	Check input and re-enter SV.
"SV" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter SV from the controlling SPO device.

TO PRBK (Test printer backup Option)

Syntax:



This intrinsic is used to interrogate the status of the system printer backup system option.

When "TO" is entered alone, the system will respond with: PRINTER BACKUP OPTION IS <status>. Where <status> will be either "ON", if the system option is to send to backup if the printer is unavailable, or "OFF" if the system option to use printer backup is not operational.

When "TO PRBK" is input, in addition to the above information, the system also outputs the following:

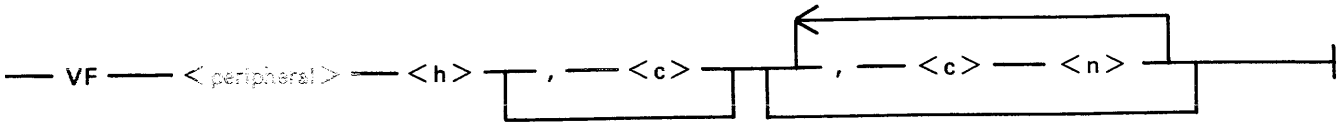
PB DISK IS <printer backup designate disk
DEFAULT BACKUP MAXIMUM FILESIZE IS <default filesize in records>

TO Output Messages

Message	Possible Causes	Suggested Action
TO faulty input INVALID – OPTION(S) NOT RECOGNIZED REFER TO SOG FOR ALLOWABLE FORMAT	The only option available is PRBK.	Check input and re-enter TO.

VF (Vertical Format on Printer, not implemented on B 1000)

Syntax:



Where:

- h – is the page height in lines.
- e – is the end of page line number.
- c – is the channel number.
- n – is the line number.

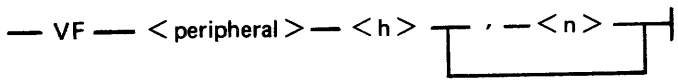
This intrinsic applies only to printers which have soft vertical format control.

This intrinsic allows the operator to define actions to be taken by the printer when certain vertical format commands are sent.

The h field specifies the page height in lines and the optional e field specifies the end of page line number. The c field defines the channel number and the n field is the line number corresponding to that channel. The channel number and line number fields are optional but when specified they must both be present as a pair. The channel number should be 2-11 and page height should not be more than 94.

NOTE
For further details, see LOAD.VFU utility.

The following syntax for VF applies to B 9252 line printers.



where h and n are defined as above.

The default form length on B 9252 printers for U.S. and Canadian character sets is 11 inches (27.9 cm). The default for all other character sets on the B 9252 is 12 inches (30.5 cm).

NOTE
LOAD.VFU expects input in a format that is incompatible with B 9252, so the LOAD.VFU utility is redundant when B 9252 is in use.

Examples:

Example 1.

To format a soft vertical printer with the following parameters:
page height = 66
end of page = 60

channel number 2 corresponds to line 10

```
spo input       VF LPA 66 , 60 , 2 10
spo output      VF LPA 66 , 60 , 2 10 COMPLETE
```

Example 2.

To format a B 9252 printer with the following parameters:
page height = 66
line number = 10

```
spo input       VF LPA 66 , 10
spo output      VF LPA 66 , 10, COMPLETE
```

VF Output Messages

Message	Possible Causes	Suggested Action
VF faulty input INVALID – PARAMETERS ILLEGAL OR OUT OF RANGE. REFER TO SOG FOR ALLOWABLE FORMAT	Typing error in initiating message.	Check input and re-enter VF.
"VF" IGNORED – peripheral NOT READY. POWER ON PRINTER OR USE "RY" TO BRING ON LINE.	The specified peripheral is not ready.	Ensure that the peripheral is connected to the system and ready.
VF faulty input IGNORED – DEVICE MUST NOT BE IN USE WHEN COMMAND ISSUED. ENTER "OL" AND RETRY WHEN DEVICE FREE	The specified peripheral is being used by a program.	Wait until the program has finished with the peripheral, then re-enter VFU.
"VF" IGNORED – peripheral DOES NOT HAVE SOFT VFU CAPABILITY AND CANNOT BE LOADED WITH FORMAT DATA. REFER TO SOG FOR MORE DETAILS	The specified peripheral is not capable of doing soft vertical formatting.	None.
"VF" IGNORED – ONLY SYSTEM OPERATOR MAY ISSUE RESTRICTED COMMAND	This intrinsic can only be entered from the controlling SPO.	Re-enter VF from the controlling SPO device.
VF IGNORED – DEVICE ERROR event no. ON peripheral – THE LP TRANSLATION TABLE text SPECIFIED IN THE SYSCONFIG FILE CANNOT	The line printer translation table for the specified peripheral cannot be found.	Seek technical assistance.

(continued)

Message	Possible Causes	Suggested Action
BE LOCATED IN FILE SYSTRANSLATE		
VF IGNORED – DEVICE ERROR event no. ON peripheral – THE REQUESTED TRANSLATION TABLE IN FILE SYSTRANSLATE CANNOT BE ACCESSED	A problem has been encountered in accessing the translation table for the specified peripheral.	Seek technical assistance.

SECTION 4
CMS-COMMON UTILITIES

INTRODUCTION

This section describes all standard CMS utilities that form part of a CMS system software release. The applicability of any utility depends on the type of hardware available. For example, utilities requiring console files cannot be executed on machines without a console: as an example, CREATE, AMEND, FL and UPDATE cannot be run on a B 900 or B 1900.

Table 4-1 gives a list of all required peripherals for each utility. In this table, required peripherals are denoted by the letter "R", and optional peripherals by the letter "O". One asterisk ("*") indicates that out of all the options, at least one is required. In particular, those utilities requiring a line printer may use a console printer by default if the line printer is not present on the system. Two asterisks ("**") indicate that out of all the options, at least two are required.

STAR FILES

The star-file facility permits the initiating message parameters of most utilities to be specified on a disk file which is referenced in the initiating message.

The utilities which do not support this feature are:

FILEUTL – File Maintenance Utility
CP – Compute
DA – Disk Analysis
KEY.CHECK – Key Validity
XD – Delete Bad Disk Sectors
PPID – Pseudo-Pack Identifier
WL – Which Log

as well as the following B 90-only utilities:

COLDSTART
INSTALL
GEN.DUMPFL – Create empty B 90 Memory Dump File
PATCHMAKER – Patch B 90 machine code, and object program files
CONFIGURER – Configure B 90 System Software
BUILDTRANS – Build Translation Table

CO supports a star-file facility which is slightly different from the general utility star-file facility. Refer to CO, Section 6, for details.

The syntax (as inserted in the initiating message) is:

The utilities which support the star-file feature have the limitation of 400 significant characters within their initiating message. (The initiating message for LD may be up to 600 significant characters in length.)

Table 4-1. Peripherals Required by CMS-Common Utilities

Utility	Console	Disk	Serial Printer	Self-Scan	Line Printer	Cassette or Mag. Tape	Card Reader	Card Punch	Paper Tape	ICMD	Terminal (Data Comm)	Printer (Data Comm)
ADD		R				R						
AMEND *	R	R	O	O								
CH		R										
CHECKADUMP		R				R						
CHECK.DISK		R										
CO *		R	O		O	O	O					
COMPARE **		O	O		O	O	O		O			
COPY **		O				O	O	O	O			
CP												
CREATE *	R	R	O	O								
DA *	O	R	O		O							
DSKUTL	O	R	O		O							
DUMP		R				R						
FL	R	R		R								
FS		R										
ICMD *		R	O		O					R		
IR		R										
KA *		R	O		O							
KEY.CHECK	R	R	O		O							
KX		R										
LB		R										
LD		R				R						
LF		R										
LIST **		O	O		O	O	O		O			
LOAD		R				R						
LR *		R	O		O							
MODIFY *	O	R	O		O						O	
PB *	O	R	O		O							O
PD		R										
PL *		R	O		O							
RM		R										
SCR	O	R	O	O	O	O						
SQ		R										
SYCOPY	R					R						
TAPELR *			O		O	R						
TAPEPD						R						
TRANSLATOR *	O	R		O							O	
TL		R										
UNLOAD		R				R						
UPDATE *	R	R	O	O								
XD		R										

Star files may contain any number of records, with any record size. A single space is considered a significant character, and any double space encountered is considered to be a single space (and hence only one significant character). Spaces at either end of the message are ignored.

Where star files are a feature of a particular utility, the star file may be placed at any point in the initiating message after the utility name. The initiating message may contain any number of star files but these may not be nested: that is, the information within a star file must not contain a call on any other star file.

If the specified file cannot be found, an "UNABLE TO ACCESS STAR FILE <file-name>" message is displayed by the utility.

Examples

1. RM *M101A/RMFILE
   where RMFILE is a disk file on disk M101A containing one record with the contents REP200, REP562, RQ=, RCOPY

2. DA *DISK1/F
   where F is a file containing a list of filenames.

3. COPY *DISK2/B
   where B is a file containing the remainder of the initiating message – "FILEA <BOTH> TO DISK3/FILES"

SYS-SUPERUTL

This system utility provides the following functions:

CH – change the name of a file or a group of files
KX – interrogate disk space
PD – interrogate disk directory
RM – remove a file or group of files
IR – initiate recall of SPO log messages
LB – look back in SPO log
LF – look forward in SPO log

It executes automatically if the program file is on the systems disk when one of these functions is required. This program is also automatically executed at warmstart time and co-ordinates logging functions at that time.

SYS-SUPERUTL supports the following filetypes:

Filetype	Description
00	Normal Data
01-0E	Source Language
0F	Source Library
10-12	Normal Code
13	Protected Code
14-16	Interpreter/SORTINTRINS/MCPX/SAU
21	SYSLANGUAGE local language
22	SYSCONFIG – a required file
30	Virtual Memory/Dump

(continued)

Filetype	Description
31	System Log
40-41	MPLII Compiler Work
81	Key
A0	Printer Backup

Of these, the following filetypes are system files:

13	Protected Code
14-16	Interpreter/SORTINTRINS/MCPX/SAU
21	SYSLANGUAGE local language
22	SYSCONFIG
30	Virtual Memory/Dump
31	System Log

A request for the removal of a system file causes the RM utility to output

<file-name> IS A SYSTEM FILE
<mix-no>/RM REQUIRES DATA. USE "AX"

Then, to remove a system file:

AX <mix-no> OK

The utility has some features which can cause the operator confusion. The utility does not appear in the response to the MX command unless it is actually performing one of its functions, when it appears as 27/PD or 27/CH etc., according to the function which it is currently performing. If an attempt is made to execute one of the SYS-SUPERUTL functions when it is already busy, then a response of

PROGRAM LOAD FAILURE <64>
<function> WILL NOT RUN -
SUPERUTILITY FUNCTION ALREADY RUNNING

is returned.

SYSLANGUAGE

All CMS utilities which output SPO messages, with the exception of PATCHMAKER, GEN-.DUMPFL, and SYCOPY, display these messages from a common SYSLANGUAGE file. THIS FILE MUST BE PRESENT IN ORDER FOR ANY OF THE UTILITIES USING IT TO EXECUTE. Associated with this file is the message:

"<utility name> CANNOT DISPLAY MESSAGE <entry no>
- THE MESSAGE IS NOT PRESENT IN THE LANGUAGE DICTIONARY FOR THIS SYSTEM"

which denotes that the utility executing has attempted to display a message which is not contained in the dictionary file SYSLANGUAGE.

NOTE

FILEUTL also uses a file called SYS.FILEUTL for screen layout information.

PSEUDO-PACKS

Pseudo-packs allow fixed disk units containing multiple disk platters to be handled as one large contiguous disk. This enables the MCP to address the space on all disk platters as one large available area.

The disk structure that makes this possible is the Pseudo-Pack Identifier Table (PPIT) and is only relevant to fixed disk directories. The fixed disk directory generated at disk initialization for systems using pseudo-packs consists of four parts:

1. The name list.
2. The disk file headers for each file.
3. The available table.
4. The Pseudo-Pack Identifier Table (PPIT).

The relationship between these parts is shown in figure 4-1.

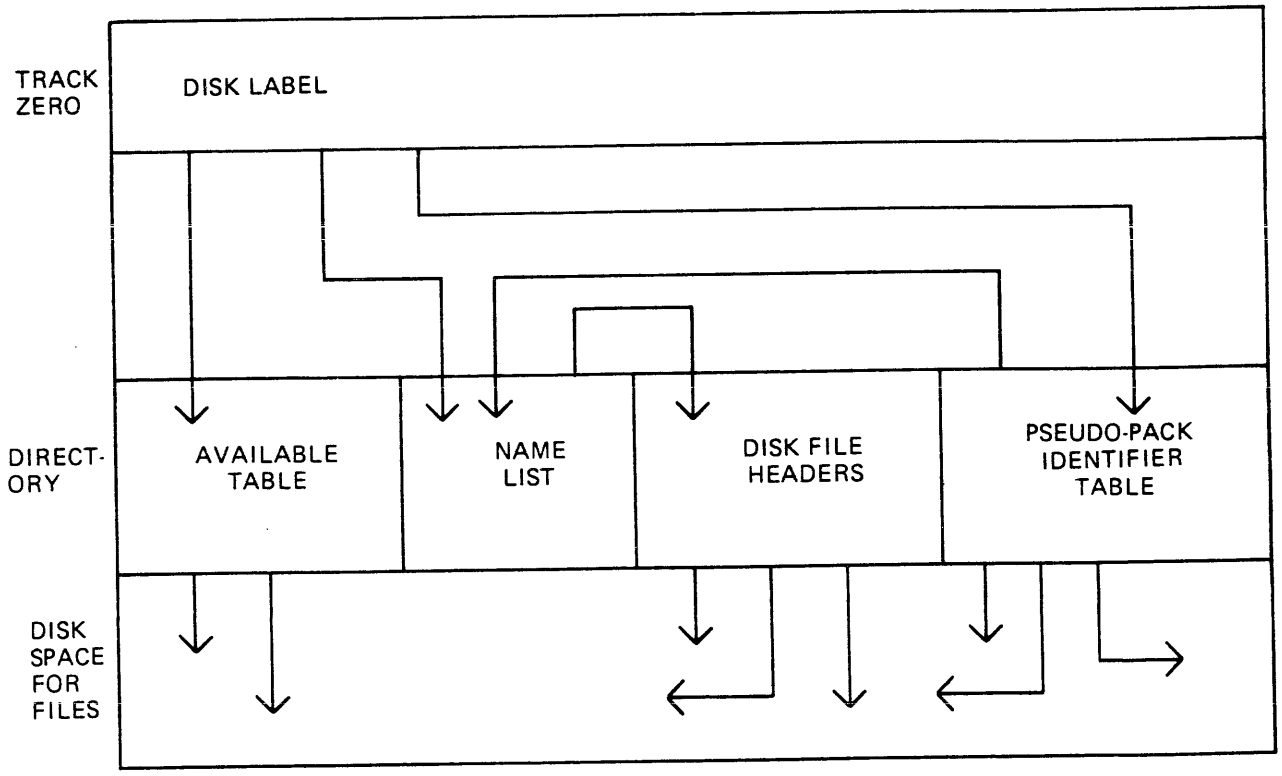


Figure 4-1. Fixed Disk Directory Structure

The available table and disk file headers generated at initialization time have entries which reflect the logical structure of the entire disk unit.

The name list now contains a list of each file on the entire disk unit and a unique identifier, which is used to distinguish files contained on one pseudo-pack from files contained on another pseudo-pack.

The PPIT is a list of all the pseudo-packs declared on the system. An identical PPIT exists on each fixed disk unit.

The ADD PACK intrinsic enables the operator to declare a new pseudo-pack to the system. It also allows the operator to declare the pseudo-pack as restricted or unrestricted. Once a pseudo-pack has been declared restricted or unrestricted, this designation cannot be changed.

When a new file is created on an unrestricted pseudo-pack, areas for the file may be allocated on any of the fixed disks. When a file is created on a restricted pseudo-pack (or with physical unit pack-id), areas for the file are allocated only on the designated unit.

When searching for a file on an unrestricted pseudo-pack, the PPIT and file directories on all fixed disk units are searched, otherwise, only the PPIT and file directory on the designated unit are searched.

The Pseudo-Pack Identifier Display (PPID) utility allows the operator to list the Pseudo-Pack Identifier Table on the operator display terminal (ODT) or line printer.

NOTE

Pseudo packs are currently implemented only on B 900 systems.

LOGGING IN THE 3.05 RELEASE

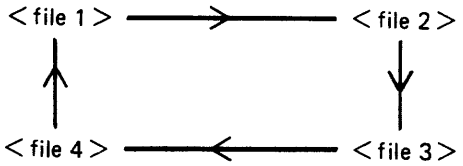
There are two possible types of information which may require to be logged by the MCP. These are maintenance information entries and, if requested, System message entries (SCL).

Maintenance log entries are used by Field Engineering personnel to assess the performance of all peripherals attached to the system. They provide information on the device usage and on any I/O failures which require corrective action to be taken by the system. Maintenance logging will always be present on the system and cannot be stopped and restarted by the user.

System message logging is a method of providing a log of all SCL input and output and is used as a tool in the diagnosis of software failures. It can also provide a hard copy record of system activity for systems using a SELF-SCAN or terminal type SPO device.

Logging in the 3.05 release functions in the following manner.

There are four files called primary log files into which the MCP will temporarily store maintenance entries, and, if required, any system message entries. The primary log files are used as illustrated in the following diagram:
if required, any system message entries. The primary log files are used as illustrated in
the following diagram:



Once a log file has been filled, the next file in the sequence is used. When all four primary log files have been filled, the sequence restarts and the first primary log file will be overwritten by any subsequent log information.

This form of log file usage is called CYCLIC.

At the point where a primary log file has been filled and the MCP begins to route the log information to the next primary log file in the cycle, the TL utility is invoked. This utility will save the contents of a primary log file to prevent the loss of log information when that primary log file is marked as "NEXT ACTIVE" and cleared for use by the MCP.

The maintenance entries are transferred into log file(s) called secondary log files(s). These files are NON-CYCLIC. That is, as each secondary log file is filled, the progression number in the file name of the secondary log file is incremented and a new file in the series is created and used.

If the system message logging option is set in the system configuration file (SYSCONFIG), there are two possible forms which may be adopted. The action taken by the TL utility is affected by which of the two forms has been requested.

The two forms of system message logging are:

1. CYCLIC system message logging. This occurs when the MCP places system messages in the primary log files only. The system messages are not transferred to NON-CYCLIC secondary log files when the primary file is filled. This form provides full system message logging back to the point where the primary files have been overwritten.

2. NON-CYCLIC system message logging. This form of system message logging treats system message entries in a manner similar to maintenance log entries. Each entry will then be preserved in a secondary log file by the TL utility at the point where a change of primary log file is required.

The following paragraphs describe the different aspects of logging in greater detail.

Types of Log File

Primary:	
SYS-LOG-01 SYS-LOG-02 SYS-LOG-03 SYS-LOG-04	These files are created at warmstart by SYS-SUPERUTL, if they are not already present, and are written to and maintained by the MCP. They each contain a hardware configuration entry and maintenance entries. System messages (if requested in CONFIGURER <SYSCONFIG>) are written to these files in a cyclic manner by the MCP.
Secondary:	
SYS-MLGjjjmm	Contains maintenance entries which have been transferred by TL from a ready to transfer primary log-file. Used in both cyclic and non-cyclic System message logging.
SYS-HLDjjjnn	Contains system messages transferred by TL from a ready to transfer primary log file. Used in non-cyclic system message logging.
SYS-LOG-HOLD	Contains system message entries from the primary log files which, at warmstart, were found to contain maintenance entries not transferred by TL to any secondary log file in the previous session of usage. This file is only used with cyclic system message logging.
Where	jjj is the Julian date and nn and mm are progression numbers in the range 00-99.

The secondary log files are created and maintained by the TL utility. The SYS-LOG-HOLD file will be recreated each time it is used. Both SYS-HLDjjjnn and SYS-MLGjjjmm are non-cyclic in nature, that is, when TL is adding entries to either of these files and finds that the file is full, it increments the respective progression number in SYSCONFIG and creates a new file with the new progression number. The TL utility will then continue using the newly created file.

Types of Logging

Maintenance logging is always active on the system. The four primary logs will always be present along with SYS-MLGjjjmm file(s). It is only if SPO logging has been requested via a CONFIGURER option

that system message entries are placed in the primary log files by the MCP. The exception is when a console SELF-SCAN is to be used as an SCL device since, in this case, system messages will always be logged. Maintenance logging functions in a NON-CYCLIC fashion only because it is very important that the data is not lost.

CYCLIC System Message Logging

The four primary log files are used with SYS-MLGjjjmm in this mode. At warmstart, TL is zipped and locates the primary log file(s) which have not had their maintenance entries transferred during the previous session. TL then transfers the maintenance entries from these primary log files into the current SYS-MLGjjjmm file, as well as transferring the system message entries to a file called SYS-LOG-HOLD.

The MCP now starts using SYS-LOG-01.

When a primary log file becomes full, TL is zipped to transfer the maintenance entries from this primary log file to the current SYS-MLGjjjmm file. The system message entries are left in the primary log file.

NOTE

While the MCP is writing to a primary log file, the next primary log file in the series is cleared by the MCP ready for use. For example, if the MCP is using SYS-LOG-01, then SYS-LOG-02 is denoted as NEXT ACTIVE and the MCP clears it for use.

When the MCP starts using SYS-LOG-04, SYS-LOG-01 will be cleared ready for use. Therefore, if the size of the primary log files is small in comparison with the number of system messages and maintenance entries, system message and maintenance entries will be overwritten by the MCP. Maintenance entries are preserved in SYS-MLGjjjmm secondary log file(s).

NON-CYCLIC System Message Logging

The four primary log files along with SYS-MLGjjjmm and SYS-HLDjjjnn are used in this mode. At warmstart, TL is zipped and transfers all the maintenance entries from the primary log file(s) which have not been transferred in the previous session to the current SYS-HLDjjjmm. TL also transfers all the System messages from this primary log file into the current SYS-HLDjjjnn.

The MCP will now start using SYS-LOG-01.

As each primary log file is filled, TL is zipped to transfer maintenance entries from this primary log file to the current SYS-MLGjjjmm. TL also transfers the system messages from this primary log file to the current SYS-HLDjjjnn. Therefore, system messages are preserved along with maintenance entries in their respective secondary log files, SYS-HLDjjjnn and SYS-MLGjjjmm.

Intrinsics and Utilities Linked with Logging

The intrinsics and utilities which are connected with the logging function of the system fall into two distinct categories. There are certain utilities which are directly involved in the creation, maintenance and control of the log files and the data recorded within them. There are also utilities which allow the operator to access the contents of the log files. The following is a list, in alphabetic order, of intrinsics and utilities linked with logging.

CONFIGURER [U]	Allows the following control over logging: 1. If system message logging is required, note that maintenance logging is always ON. 2. The choice of CYCLIC or NON-CYCLIC system message logging. 3. The size of the primary log files (that is, SYS-LOG-01 to 04). 4. The size of the secondary system message file (that is, SYS-HLDjjjnn). 5. The size of the secondary maintenance entry file (that is, SYS-MLGjjjmm). 6. The name of the disk where the secondary log files are to reside. This information is stored in the file called SYSCONFIG.
IR [S]	Initiates recall of system message entries to a SPO capable device. Only system message entries from primary log files which have not been transferred by the TL utility may be accessed. IR also enables the use of LF (Look Forward) and LB (Look Back) functions. IR, LB or LF will be terminated when any other SYS-SUPERUTL function is invoked.
LB [S]	Will cause the display of system message entries from a point further back in the primary log files than the last entries to be displayed to the SPO.
LF [S]	Will cause the display of system message entries from a point further forward in the primary log file than the last entries to be displayed to the SPO.
LG [M]	Starts logging, that is, the MCP begins writing system message entries to the primary log files.
LS [M]	Stops logging, that is, the MCP stops writing system message entries into the primary log files.
PL [U]	Permits the printing of a log file, in full or in part, over a given time interval. It also allows the option of automatically removing the printed log file.
SYS-SUPERUTL	Creates the four primary log files, if they are not present at warmstart. The MCP then writes maintenance entries and, if requested, also system messages into the primary log files.
TL [U]	Creates and maintains secondary log files when required (TL is ZIPped by the MCP automatically).
WL [U]	Displays the current state of the primary log files and the active secondary log files.

NOTE

Items marked [M] are MCP intrinsics and full details of their use are included in section 3 CMS Common Intrinsics.

Items marked [U] are utilities and full details of their use are given under CMS Common Utilities later in this section.

Items marked [S] are functions of SYS-SUPERUTL, full details of their use are given under CMS Common Utilities later in this section.

Suggested Logging Method

The suggested logging method, described below, ensures that no system message is "lost" due to overwriting of the primary log files (as these files are cyclic in nature).

To set up the parameters required, configurer needs to be run and the following changes made to SYSCONFIG:

1. SPO logging must be switched ON.
2. NON-CYCLIC logging is required.
3. The sizes of the primary and secondary log files will have to be determined by usage for efficient use of logging. A suggested size for the secondary system message log files on a fixed disk based system is 400 sectors. All other log file sizes are left at the default values.
4. The backup disk for log files should be left as the system disk.

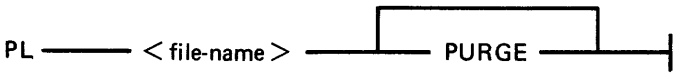
The following logging procedure should be adopted:

1. At warmstart, the operator should execute a PD of the secondary log file family names to find out which log files are present. The following syntax is recommended for PD:

   PD SYS-HLD=, SYS-MLG=

   NOTE
   The files with the family prefix SYS-HLD contain the system message entries and the files with the family prefix SYS-MLG contain the maintenance entries.

2. The operator then prints the system message and maintenance log files, if required, using the following syntax.



Where: <file-name> is the log which is required

PURGE will remove the secondary log file after it has been printed.

See PL utility for other options.

If the above procedure is followed during every warmstart with the PURGE option in the initiating message of PL (for all the SYS-HLDjjjnn and SYS-MLGjjjmm), the user should have only the current session's log files on the system.

Log File Sizes

The size of the log files required depends on several factors.

To minimize operator activity in printing log files, for NON-CYCLIC logging, the size of the secondary log file should be large enough to hold a full day's system messages. Similarly, the size of the maintenance log files for both CYCLIC and NON-CYCLIC modes should be large enough to hold a full day's entries. Under normal circumstances (that is, unless a hardware problem exists on the system), the number of maintenance entries will be small.

In this way, only two files will be created per day for NON-CYCLIC logging. If the system disk space is limited, the secondary log files can be directed to a user disk. Also, the size of the primary log files can be kept to a minimum. However, the smaller the primary log files, the more often TL will be zipped to transfer entries to the secondary files. Frequent use of TL may adversely affect system response time. The length of time taken to transfer entries is proportional to the size of the primary log files.

If the system is warmstarted daily and system disk space is not critical, the primary log files can be made large enough to ensure that they are never filled. In this way TL will only be zipped at warmstart and no interruption to the system will occur during normal use.

In order to calculate the required file size in sectors, the following algorithm is suggested:

$$TOT.size = TOT.msgs.1 + \frac{(TOT.msgs.2 \times \frac{AVG.msgs.2}{32})}{4}$$

TOT.size = total number of sectors to hold all system messages
TOT.msgs.1 = total number of messages shorter than 33 bytes
TOT.msgs.2 = total number of messages longer than 32 bytes
AVG.msgs.2 = average length of messages longer than 32 bytes

TOT.size should then be increased by 10 percent to allow for fluctutations in system usage.

NOTE

The total number of messages produced by a system can be determined by examining the system log files on a day of heavy activity.

ARCHIVING USING UPDATE FACILITY

By using the keyword UPDATED in the initiating message of LD or DUMPADISK, the operator may choose to have only updated or newly created files dumped without having to identify such files by name.

This is possible from the 3.04 release onwards, as the UPDATED bit in the Disk File Header (DFH) has been implemented (see the MCP Reference Manual, form number 2007555). When a disk file is successfully written to, the MCP sets the UPDATED bit in the DFH of that file. The DFHs of both related Key and Data files of indexed pairs are handled together. This ensures that a Key file will be marked updated along with its Data file on a successful rewrite, overwrite, add or delete.

Any new or updated file therefore has its UPDATED bit set automatically. The UPDATED bit is reset when a file is successfully backed-up by one of the following utilities: LD (DUMP and UNLOAD), CHECKADUMP, and DUMPADISK.

If DUMP & CHECK is requested, LD does not itself reset the UPDATED bit. Instead, CHECKADUMP resets the UPDATED bit in the DFH of any file that compares successfully. If CHECKADUMP is initiated other than being zipped by LD, then it does not alter the UPDATED bits.

The B 90 Stand Alone Utility (SAU) does not alter the UPDATED bit when copying files. The setting of the UPDATED bit will therefore be identical in both the old and new copies of the file.

The B 90 COLDSTART utility always resets the UPDATED bit in the DFHs of files it loads. All other utilities set the UPDATED bit when they create or write to a file.

The LR utility includes the setting of the UPDATED bit in its printout.

The following is an example of a simple Archiving System, assuming a five day working week.

1. On each day, Monday through Thursday, only updated files are dumped.

   The initiating message for dumping to tape is:

LD DUMP & CHECK UPDATED PRINT.DIR FROM <disk-name> TO <tape-name> =

The initiating message for dumping to disk is:
DUMPADISK PRINT.DIR DUMP UPDATED FROM <disk-name> =

2. On Friday, all files are dumped.

   The initiating message for dumping to tape is:
   LD DUMP & CHECK PRINT.DIR FROM <disk-name> TO <tape-name> =

   The initiating message for dumping to disk is:
   DUMPADISK PRINT.DIR DUMP FROM <disk-name> =

These five tapes/disks could be recycled weekly, or kept for as long as thought necessary.

Assuming that the tapes/disks are recycled weekly, and a disk is lost and must be rebuilt, the following recovery procedure could be used:

1. Initialize the disk in question – <disk-name-user>.

2. Working backwards from the previous day's backup tape/disk to the last full dump tape/disk (the previous Friday).

   The initiating message to load back from tape is:
   LD ADD PRINT.DIR TO <disk-name-user> FROM <tape-name> =

Example:

Assume daily tapes are called MONDLY, TUEDLY, WEDDLY and THURDLY.
Assume weekly tape is called FRIWKLY.

1. If the failure occurs on a Monday, then:
   1) Initialize <disk-name-user>.
   2) ADD PRINT.DIR FROM FRIWKLY TO <disk-name-user> =

2. If the failure occurs on a Thursday, then:
   1) Initialize <disk-name-user>.
   2) ADD PRINT.DIR FROM WEDDLY TO <disk-name-user> =
      ADD PRINT.DIR FROM TUEDLY TO <disk-name-user> =
      ADD PRINT.DIR FROM MONDLY TO <disk-name-user> =
      ADD PRINT.DIR FROM FRIWKLY TO <disk-name-user> =

Once the disk has been rebuilt in this way using LD or DUMPADISK, all of the UPDATED bits will be set. This results in all of the files being dumped next time a DUMP UPDATED is requested. To reset the UPDATED bits, it is necessary to re-dump all the files.

NOTE

The specification of PRINT.DIR in the initiating messages of LD and DUMPADISK causes an accurate directory to be printed.

See LD utility and DUMPADISK utility for the full syntax.

COMMON UTILITY OUTPUT MESSAGES

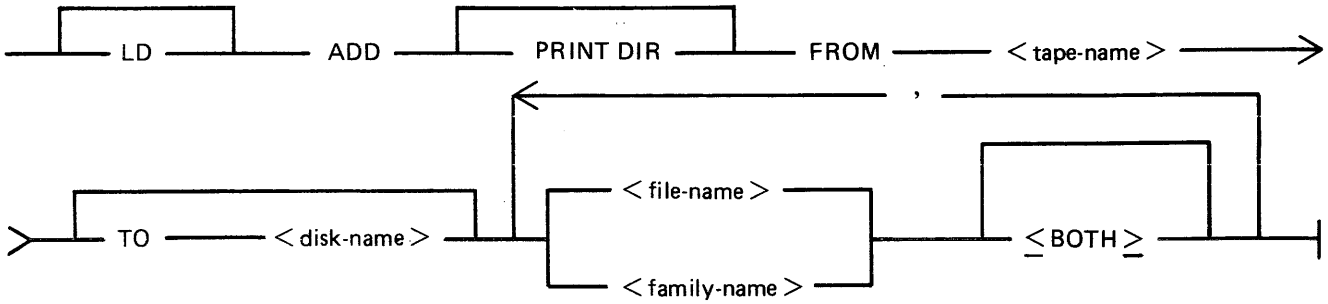
Message	Possible Causes	Suggested Action
INCORRECT INPUT TO <utility-name>	The input to the specified utility is wrong. The second message given explains the type of error.	Look for type of error in the second message displayed by the utility.
-INVALID CHARACTER IN <file-name>	Disk name or file name contains characters(s) not permitted by the system. Valid characters are: A-Z, 0-9, . (dot), - (dash).	Correct input and re-enter.
-FILE NOT FOUND	Specified file name is not on the disk.	Check input for correct file and disk name.
-FILE IS CURRENTLY BEING USED	Specified file is being used by another program.	Re-enter when other program has stopped using the file.
-REQUEST IS INCOMPLETE	Input message is incomplete.	Check input and re-enter.
-DISK <disk-name> IS NOT ON LINE	Specified disk is not on-line to computer.	Make disk available. "RY" drive.
INCORRECT INPUT TO <program name> <character string	Input message is incomplete or incorrect.	Check input and re-enter.
CANNOT RUN <utility name>	The utility cannot run. The reason is given in the next error message displayed.	None.
UNABLE TO CONTINUE WITH <utility name>	The utility cannot continue. The reason is given in the second message displayed.	Look for reason in the second message displayed by the utility.
-FILE IS NOT A SOURCE OR A DATA FILE	The file requested must be file type source or data @00@-@0F@.	Check file and re-enter.
-CANNOT HANDLE A REQUEST OF THIS LENGTH	The initiating message is too long.	Check input and re-enter.
-REQUESTED FILES WERE NOT FOUND	The specified files are not in the directory of the specified disk.	Check input for correct file names.
- DISK <disk-id> IS CURRENTLY LOCKED BY ANOTHER PROGRAM.	Another program has opened the specified disks SYSMEM with lock.	Wait for the other program to go to end-of-job.
-SYSTEM DISK IS CURRENTLY LOCKED BY ANOTHER PROGRAM.	Another program has opened the system disks SYSMEM with lock.	Wait for the other program to go to end-of-job.

(continued)

Message	Possible Causes	Suggested Action
FILENAME <filename> EXCEEDS 12 CHARACTERS	The specified filename is too long.	Correct input and re-enter.
-UNABLE TO ACCESS STAR FILE <filename>	The specified file is either not on disk or is in use.	Make file available and re-enter.
-DISKNAME <diskname> EXCEEDS 7 CHARACTERS	The specified diskname is too long.	Correct input and re-enter.
-UNABLE TO ACCESS STAR FILE <filename>	The specified file is not in the directory of the specified disk.	Check input for correct file name.

ADD (ADD FROM LIBRARY TAPE TO DISK)
(LD – TAPE UTILITY)

This function is part of the utility LD. ADD provides the capability of copying files or groups of files, which do not already exist on the disk, from a library tape to the disk specified by disk-name, or if no disk-name is specified, to the system disk.



PRINT.DIR	If PRINT.DIR is specified, a printout of all the files loaded will be produced at the end of the ADD. When PRINT.DIR is specified, the message "<file-name> LOADED" will be suppressed. Displays will be limited to exception conditions, that is, if a file is not loaded for some reason.
FROM <tape-name>	This is the name of the tape from which the files are to be loaded.
TO <disk-name>	If the TO option is specified, the name of the disk to which the files are to be loaded must be given. If the TO option is not used, then the files will be loaded on to the system disk.
<file-name> <family-name>	This is the file or family of files to be added to the disk. Groups of files may be added by separating the file names or family names by a comma ",".
<BOTH>	If <BOTH> is specified immediately following a request to load a key file then, provided that the pertinent data file does not precede the key file on the library tape, the data file will also be loaded and a suitable amendment will be made to the key file so that it points to the disk which now holds the data file (rather than the disk from which the data file was dumped).

Since the ADD function is part of the utility LD, "LD" is actually what appears in a mix message. To discontinue the ADD function, "DS <mix-number>/LD" must be used.

Examples

Example 1. To copy all the files which are not already present from the tape named UPDATE to the system disk, the following syntax is used:

```
LD ADD FROM UPDATE =
```

Example 2. To copy the keyfile named QUARCK and its associated data file from the tape called QACK to the disk called Q3, the following syntax is used:

```
LD ADD FROM QACK TO Q3 QUARCK <BOTH>
```

Example 3. To copy all the files from the tape called BACKUP in the family JOB.,

which do not already exist on the disk DISK1, the following syntax is used:

LD ADD PRINT.DIR FROM BACKUP TO DISK1 JOB.=

With the addition of the option PRINT.DIR, a printout of all the files which ADD loaded will be produced.

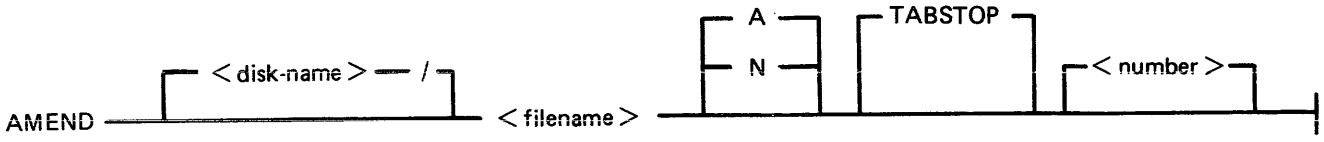
Output Messages

For ADD output messages, refer to LD utility.

AMEND (Disk File Amending – Applicable to B 90 Only)

This utility is used to modify records within an existing data or source file. The CREATE and UPDATE utilities use many similar features. It is only available for use on systems which have console files. The utility supports the star-file feature. If a file is AMENDed, the generation number of the file is incremented by one.

Format:



Input may be either alphanumeric (A) or hexadecimal (N) (see CREATE for details). The default is A.

Entering TABSTOP in the initiating message causes AMEND to set up TAB positions coinciding with the end of the console line as well as any other tabs specified. Tabs must be used with AMEND, despite the fact that no new records are being entered.

Default tab positions have been chosen to allow a maximum number of characters to be inserted on one line. AMEND uses 9 for the record number and 110 for the contents of the record. In addition, manually selected tabs may still be used. The end-of-console line-tabs (depending on record size and file-type) are as follows:

Source or Data alphanumeric : 111 221 331 441
Data Hexadecimal : 56 111 166 221 276 331 386 441 496

These tab positions are the same as those set for CREATE with regard to record input, although AMEND has no facility to input new records.

Examples:

	Tabs Set at:
AMEND FILEA TABSTOP	111 221 331
AMEND FILEB A TABSTOP 51 61 221	51 61 111 221 331 441
AMEND FILEC N TABSTOP	56 111 166 221 276 331 386 441

In these examples,

FILEA is a sourcefile with a record size of 410,
FILEB is a datafile with a record size of 500,
FILEC is a datafile with a record size of 450.

The "number" option may be used to set tab positions for character input (see CREATE for details).

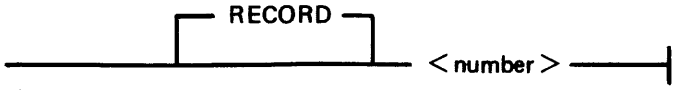
The utility operates in two modes: "Record Modify" (PK2) and "Record Select" (PK3).

PK1	PK2	PK3	PK4	PK5	PK6
NEXT	MODIFY	SELECT	–	–	END

An OCK4 "help" option is provided, which will output the above options when pressed in either Modify or Select mode. In order to show which mode the utility was in when OCK4 was pressed, an asterisk (*) is printed next to that mode on the Help display.

PK1 is used to select the next sequential record in the file to be printed. The use of PK1 terminates "Record Modify" and "Record Select" modes, therefore a re-selection of mode must be made before continuing.

If PK3 ("Record Select" mode) is used, the required record is identified by logical record number using this format:



The "number" may take any value from 1 to the number of records in the file.

PK2 is used to make corrections to existing records. This PK operates as PK2 in CREATE utility.

The point in the record at which alterations are to be made is selected by typing an identifying group of characters immediately preceding the bytes(s) of the record to be altered. The portion of the record to be replaced or inserted follows the identifying characters, delimited by a colon (:). If alterations are to be made at the beginning of the record, no identifying characters are necessary. A starting byte position for the identifying character string search can be specified in the console input.

Example:

To amend a source file called MYFILE, record size 40 bytes, tab set at 5, 10, 15, 20:

AMEND MYFILE 5 10 15 20

First select a record by pressing PK3, and then enter "20" for logical record 20 in MYFILE. The utility selects and prints the contents of record 20:

20 ABCDEFGHIJKLMNOPQRST

To replace characters, press PK2 and type the replacement

D : ZZZZ : OCK1

resulting in "20 ABCDZZZZIJKLMNOPQRST"

Or if insertion of characters is desired, type the characters to be inserted into the record:

Z : XXXXXX : OCK2

resulting in "20 ABCDZXXXXXXZZZOPQRST"

NOTE

The insertion from character six to eleven will result in the shifting of characters "ZZZIJKLMN" from byte position 12 to the boundary of the next tab

position, which is 15. Therefore only 3 characters "ZZZ" are shifted from 12 to 14 and "IJKLMN" are lost. The text from the next tab position 15 onwards is not affected.

Starting Byte for Modification

A starting byte can be specified for the modification of a record.

If both the identifying string and the start position are specified, the utility scans from the start position for that identifying string. The portion of the record before the start position is ignored. If that identifying string does not exist, "BYTE WITHIN RECORD SPECIFIED NOT FOUND" is displayed and the utility then awaits re-input.

Format:

identifying string : amending character string : start position
or
: amending character string : start position
or
identifying string : amending character string :

Example:

The following file (named FILEA) is to be amended.

Rec. No.	Contents of Record
1	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
2	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
3	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
4	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
5	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
6	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCC DDDDDEEEEE
7	AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE

1. This first example illustrates the modification of each record by replacement.

   Enter: "AMEND FILEA"

   The following is displayed:

   ?DATA AMEND

   PK1 PK2 *PK3 PK6
   NEXT MODIFY SELECT END

   Press PK1. This causes the next record to be displayed (in this case, the first):

   1 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE

   Enter the amending command and press OCK1 (for replacement):

   BBBBB:XXXXX: and terminate with OCK1.

AMEND now displays the AMENDed record:

```
    1 AAAAAABBBBBXXXXXDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE

Press PK1.

    2 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
    :XXXXX:11 (OCK1)
    2 AAAAAABBBBBXXXXXDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
Press PK 1
    3 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
    BBBBB:XXXXX:2 (OCK1)
    3 AAAAAABBBBBXXXXXDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
Press PK1
    4. AAAAAABBBBBCCCCCDDDDD AAAAAABBBBB CCCCCDDDDDEEEEE
    4 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBXXXXXDDDDDEEEEE
Press PK1
    5 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
    :XXXXX:31 (OCK1)
    5 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBXXXXXDDDDDEEEEE
Press PK1
    6 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
    BBBBB:XXXXX:7 (OCK1)
    6 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBXXXXXDDDDDEEEEE

    ?END AMEND
```

2. In this next example, insertion is being performed:

    Enter: AMEND FILEA

    As for replacement, the following is displayed:

```
    PK1 PK2 *PK3 PK6
    NEXT MODIFY SELECT END
```

    Press PK1 to display the next record:

```
    1 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
```

    Enter the amendment followed by OCK2:

```
    BBBBB:XXXXX:  (OCK2)
    1 AAAAAABBBBBXXXXXCCCCC DDDDDAAAAABBBBBCCCCCDDDDD
Press PK1.
    AAAAAABBBBBCCCCCDDDDD AAAAAABBBBB CCCCCDDDDDEEEEE
    :XXXXX:11 (OCK2)
    2 AAAAAABBBBBXXXXXCCCCC DDDDDAAAAABBBBBCCCCCDDDDD
Press PK1
    3 AAAAAABBBBBCCCCCDDDDD AAAAAABBBBBCCCCCDDDDDEEEEE
    BBBBB:XXXXX:2 (OCK2)
    3 AAAAAABBBBBXXXXXCCCCC DDDDDAAAAABBBBBCCCCCDDDDD
Press PK1
```

```
    4 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
    BBBBB:XXXXX:26 (OCK2)
    4 AAAAABBBBBCCCCCDDDDD AAAAABBBBBXXXXXCCCCCDDDDD
Press PK1
    5 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEE
    :XXXXX:31 (OCK2)
    5 AAAAABBBBBCCCCCDDDDD AAAAABBBBBXXXXXCCCCCDDDDD
Press PK1
    AAAAABBBBBCCCCCDDDDD AAAAABBBBB CCCCCDDDDDEEEEE
    BBBBB:XXXXX:26 ( OCK2)
    6 AAAAABBBBBCCCCCDDDDD AAAAABBBBBXXXXXCCCCCDDDDD

    ?END AMEND
```

Output Messages

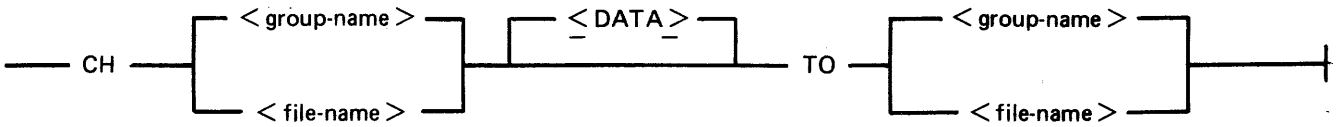
Refer to the CREATE utility later in this section for output messages.

CH (Change File Name)

(a function of SYS-SUPERUTL)

This utility allows the operator to change the name of a file or group of files on disk. The <DATA> option allows the data file of an indexed pair to be changed, and it will also cause the keyfile to refer to the new data file name (the keyfile name does not change).

Format:



The message "END CH" is displayed when CH has finished.

Examples:

To change the name of single file:

```
CH BPS320D/DCSTSK36K TO DCSTSK
CH DCSTSK TO INDISK3TSK
```

To change a group of files:

```
CH BPS320A/AR= TO BP=
CH PRB= TO PR=
```

To change several different files:

```
CH DCSTSK TO INDISK3TSK, BPS320A/AR TO BP
```

To change the name of the data file of an indexed pair:

```
CH AR200K <DATA> TO AR200BU
```

Note: if a change of group file name is specified with the <DATA> option, then the data file should appear in the directory after the keyfile. If this is not the case, then the name of the data file is changed first, and when the attempt to change the keyfile name is made, a "data file-name NOT FOUND" message will be displayed. This will not occur when changing the name of a single indexed file.

Output Messages

Message	Possible Causes	Suggested Action
<filename> NOT CHANGED	The specified file is not changed. The reason is given in the second message displayed.	Look for the reason in the second message displayed.
FAMILY <familyname> NOT CHANGED	The specified files are not changed. The reason is given in the next message displayed.	Look for the reason in the second message displayed.

(continued)

Message	Possible Causes	Suggested Action
-INVALID FILE NAME WOULD BE GENERATED	The file-id to be generated is either SYSMEM or spaces or longer than 12 characters.	Re-input with other file name.
-THE FILE IS ALREADY ON DISK	Attempt has been made to duplicate the name of a file on disk.	Re-input with other output filename.
-KEY FILE <filename> NOW POINTS TO DATA FILE <filename>	Successful completion of data file name change.	None.
CH WILL NOT RUN - SUPERUTILITY FUNCTION ALREADY RUNNING	Another function of SYS-SUPERUTL is being executed.	Wait until SYS-SUPERUTL is free, then re-input.
<filename> CHANGED TO <filename>	Successful completion of filename change.	None.

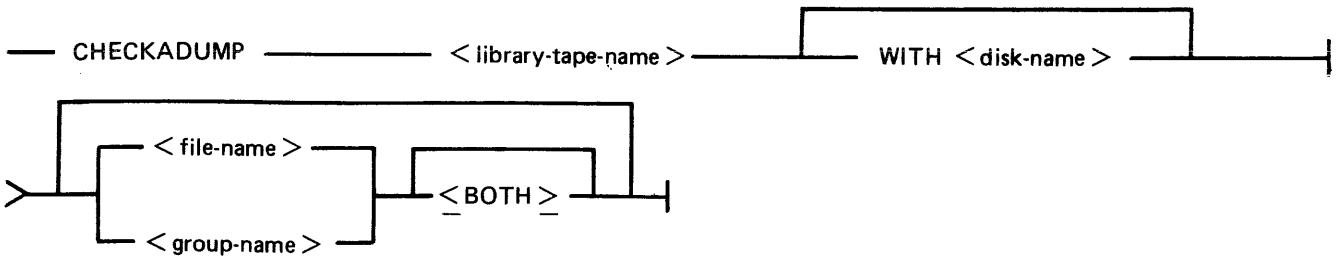
CHECKADUMP (Compare Library Tape with Disk)

This utility allows the operator to compare information in files on a library tape with corresponding files on disk. It is used to verify that a library tape is correct after files have been DUMPed, or that disk files are correct after files have been ADDed or LOADed. Specified tape is processed sequentially, file by file, and the disk is searched for corresponding files. The utility will notify the operator on up to four errors in a given file. If there are more than four errors, it will ignore the rest of that file, and proceed to the next file on tape.

The maximum size of the initiating message is 600 bytes in length.

If CHECKADUMP has been ZIPped by the LD utility and the comparison produces no conflicts, then the 'Updated bit' is reset. See ARCHIVING USING UPDATE FACILITY for an explanation of 'Update bit'.

Syntax:



<library-tape-name>	This is the name of the library tape.
<disk-name>	This is the name of the disk on which the files to be compared reside.
<file-name> <group-name>	These are the files to be compared.
<both>	If the option "<both>" follows a keyfile, then the corresponding data file will also be compared.

Examples

To compare files on the tape called PRTAPE with the corresponding files on the system disk:

```
CHECKADUMP PRTAPE
```

To compare files on the tape called ARTAPE with the corresponding files on a disk called ARDISK2:

```
CHECKADUMP ARTAPE WITH ARDISK2
```

To compare the file TESTFL on the tape called BRTAPE with the corresponding file on the disk called ARDISK3:

```
CHECKADUMP BRTAPE WITH ARDISK3/TESTFL
```

To compare the keyfile KEY01 and the data pertaining to it, both keyfile and data file are on the system disk, and the tape name is TAPE:

```
CHECKADUMP TAPE KEY01 <BOTH>
```

Output Messages

Message	Possible Causes	Suggested Action
-<tape name> IS NOT A RECOGNIZED LOAD/ DUMP TAPE	First record of tape not recognized by CHECKADUMP. Tape may not have been created by LD utility.	None.
<file name> NOT CHECKED	CHECKADUMP has not checked the file. The reason is given in the second message displayed.	Look for the reason in the second message given.
FAMILY <family name> NOT CHECKED	CHECKADUMP has not checked the family name. The reason is given in the second message displayed.	Look for the reason in the second message given.
-DIRECTORY ON TAPE <tape name> IS CORRUPT	More or fewer entries in directory on tape than specified in the first record of tape.	None.
COMPARISON ERROR DETECTED WHILE CHECKING <file name>	Comparison error found while checking filename. Reason is given in the second message displayed.	Look for the reason in the second message given.
-DISK FILE HEADERS DO NOT MATCH	Header in body of tape is not identical to respective header in disk directory.	Recreate dump tape.
-FILE SIZE MISMATCH	Difference in sizes of disk and tape files.	Recreate dump tape.
-FILE TYPE MISMATCH	Difference in file types of files being compared.	Recreate dump tape.
-RECORD SIZE MISMATCH	Difference in record sizes of files being compared.	Recreate dump tape.
-BLOCK SIZE MISMATCH	Difference in block sizes of files being compared.	Recreate dump tape.
-GENERATION NUMBER MISMATCH	The file has been modified since the creation of the tape.	As file has been changed, new back up is necessary.
NO DISCREPANCIES BETWEEN DUMP TAPE <tape name> AND DISK <disk name>	CHECKADUMP found no discrepancies while checking contents of the tape.	None.
DISCREPANCIES FOUND BETWEEN DUMP TAPE <tape name> AND DISK <disk name>	Discrepancies found while checking the contents of the tape.	Recreate dump tape.

(continued)

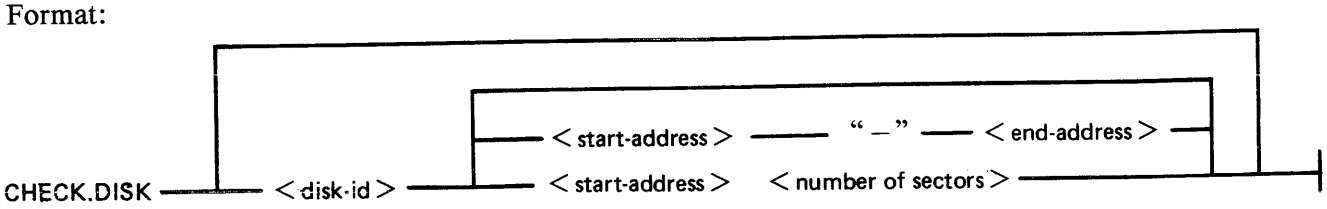
Message	Possible Causes	Suggested Action
-HARDWARE ERROR DETECTED WHILE READING DISK <disk name>	The utility has found an irrecoverable read error while reading. the specified disk.	Check integrity of specified disk.
-FILE WAS NOT SUCCESSFULLY DUMPED	The specified file has not been dumped successfully.	Recreate dump tape.
-TAPE SHOULD HAVE BEEN PURGED	The information on the tape is such that CHECKADUMP cannot handle it.	Recreate dump tape after purging the tape.
-NO FILES TO CHECK	There are no files on the tape which can be checked with the files on the specified disk.	Check input and re-enter if necessary.

Note: Refer to "Common Utility Output Messages" for additional messages.

CHECK.DISK (Check all Sectors of a Disk)

This utility checks either a specified area, or the whole area, of the specified disk by checking blocks of 32 sectors. If an error occurs in a block, each sector within that block is checked individually.

Format:



It is possible that, because of a hardware failure, an error may be detected when a block read is being performed but no errors are detected during the subsequent sector-by-sector read of that block. This is referred to as an "inconsistent error".

When inconsistent errors are encountered, CHECK.DISK continues execution. It is therefore possible to have several read error messages output by the MCP while the utility displays a "NO HARDWARE ERRORS DETECTED" message on completion of the check.

Checking of only defined areas is permitted by specifying sector ranges in the initiating message (start and end addresses or number of sectors).

These sector ranges may be up to eight decimal numbers or up to six-digit hex numbers delimited by the characters @ (AT) or . (PERIOD).

Example:

```
CHECK.DISK ARBK 32-128
or CHECK.DISK ARBL @000020@-@000080@
```

The disk-name must be specified if sector ranges are required. If sector ranges are not specified, the utility will default to checking the complete disk from sector zero to the end address.

The following fetch values are returned by CHECK.DISK for abnormal termination:

Fetch Value	Description of Cause
2	Illegal parameters found.
32	A non-fatal parameter found; for example, sector range exceeding maximum disk address.

Output Messages

Message	Possible Causes	Suggested Action
– <disk-name> IS NOT A PHYSICAL DISK	An attempt was made to run CHECK.DISK on a pseudo-disk.	Check input for correct disk name.
HARDWARE ERROR DETECTED WHILE CHECKING SECTOR <number> ON DISK <disk-id>	The utility has found an error. The impact of the error will be given in the second sentence displayed.	Look for the reason in the second message given.

(continued)

Message	Possible Causes	Suggested Action
-SECTOR IS DELETED AND SHOULD NOT CAUSE SYSTEM PROBLEMS	Self-explanatory.	None.
-SECTOR IS RELOCATED AND SHOULD NOT CAUSE SYSTEM PROBLEMS	Self-explanatory.	None.
-SECTOR IS AVAILABLE	The sector containing the error is referenced in the available table and does not cause a system problem at this time. However once the available area is occupied it will cause a problem.	Use "XD"to delete the sector, or save the information and re-initialize the disk.
-HARDWARE ERROR DETECTED WHILE READING LABEL ON DISK <disk-name>	The utility has found an error in the label of the specified disk.	Use Stand-alone utiltiy "IN" or DSKUTL IN to re-initialize the disk.
-SECTOR IS ALLOCATED TO FILE <file-name>	The sector in error is a part of the specified file.	Use COPY with the RECOVER option to save the error free part of the file.
-SECTOR IS ALLOCATED TO THE DIRECTORY AVAILABLE TABLE	The sector in error is a part of the available table.	To prevent problems the disk should be initialized.
-SECTOR IS ALLOCATED TO THE FILE DIRECTORY NAME LIST	The sector in error is a part of the name list.	To prevent problems the disk should be initialized.
-ALLOCATION OF SECTOR IS UNKNOWN	Due to other problems on the disk the utility cannot find where the sector in error is allocated.	To prevent problems the disk should b be initialized.
-ALLOCATION OF SECTOR CANNOT BE DETERMINED DUE TO ERROR ON DIRECTORY	Due to other problems on the disk the utility cannot find where the sector in error is allocated.	To prevent problems the disk should be initialized.
-SECTOR IS ALLOCATED TO A TEMPORARY FILE	The sector in error is a part of a temporary file. Any program using this file will detect an error and may go DS or DP.	Make the sector available and run "XD".
-SECTOR IS ALLOCATED TO A TEMPORARY FILE ON PSEUDO-PACK <disk-name>	The sector in error is a part of a temporary file. Any program using this file will detect an error and may go DS or DP.	Make the sector available and run "XD".

(continued)

Message	Possible Causes	Suggested Action
HARDWARE ERROR DETECTED AROUND SECTOR <number> WHILE CHECKING DISK <disk-name>	The utility detected an error but cannot find out the exact address. The reason is given in a second message.	Look for the reason in the next message displayed.
-EXACT LOCATION CANNOT BE DETERMINED	Due to other problems on the disk the utility cannot find the exact location of the sector in error.	To prevent problems, the disk should be initialized.
-SECTOR IS IN TRACK ZERO	The utility has detected an error in track zero of the specified disk.	Initialize the disk.
-FILE <file-name> IS NO LONGER ACCESSIBLE	The specified file can no longer be accessed since the file header is unreadable.	Back up the rest of the files and re-initialize.
NO SECTORS CHECKED ON DISK <disk-name>	The utility was not able to check the specified sectors on the specified disk. The reason will be given in the second sentence displayed.	Look for the reason in the second message given.
DISK <disk-name> CHECKED	Normal output.	None.
DISK <disk-name> SECTOR RANGE <sector range> CHECKED	Normal output.	None.
DISK <disk-name> SECTOR <number> CHECKED	Normal output.	None.
-<number> HARDWARE ERROR(S) WERE DETECTED WHICH COULD CAUSE SYSTEM PROBLEMS.	Self-explanatory.	Back up the files and re-initialize the disk if the number is more than zero.
-HARDWARE ERROR(S) WERE DETECTED BUT SHOULD NOT CAUSE SYSTEM PROBLEMS	Self-explanatory.	None.
-NO HARDWARE ERROR(S) DETECTED	Your disk is in a perfect condition.	None.
-TOO MANY HARDWARE ERRORS DETECTED, PLEASE POWER OFF DISK <disk-name>	The disk is in a very bad condition. If re-initialize does not solve the problem, you may need technical assistance.	Re-initialize the disk then re-run CHECK.DISK

(continued)

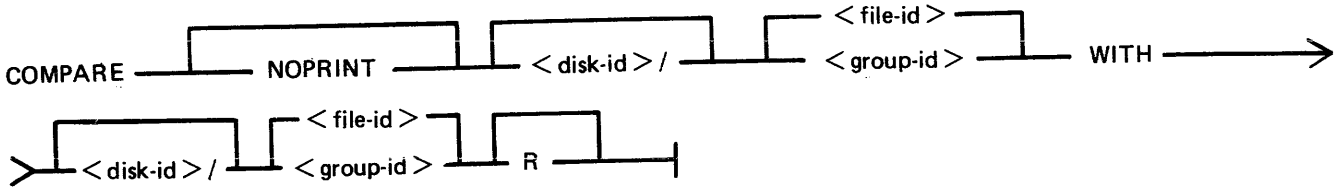
Message	Possible Causes	Suggested Action
SECTOR RANGE FROM <number> IGNORED, BEYOND END OF DISK <disk>	The sector range after the specified number is too high for the specified disk.	Check input.
END OF DISK <disk-name> REACHED DURING RANGE FROM SECTOR <number>	While checking a part of the specified disk the end of the disk was reached before the end of the specified range.	None.
-SECTOR IS ALLOCATED TO THE PSEUDO DISK TABLE	The sector in error is part of the PPIT.	Back up the files and re-initialize the physical disk.
-SECTOR IS ALLOCATED TO THE FILE DIRECTORY HEADER AREA	The sector in error is a part of the file directory header area.	Back up the files and re-initialize the disk.
CHECK.DISK ON <disk-name> COMPLETED - NO ERRORS	Normal EOJ messages.	None.
CHECK.DISK ON <disk-name> SECTOR <address> - NO ERRORS	Only one sector specified for checking and found to be readable.	None.
CHECK.DISK ON <disk-name> AREA <address> - <address> COMPLETED	An area of sectors checked.	None.
SECTOR <start-address> BEYOND END OF DISK <disk-name>	The utility will continue with the next sector range specified.	None.

Note: refer to "Common Utility Output Messages" for additional aid.

COMPARE (Compare Files)

This utility compares corresponding records in two files, or in pairs of files within two groups. A realignment feature is also available as an aid to detecting missing records.

Format:



The NOPRINT option results in suppression of the full printed error listing. Instead, the following is displayed on the SPO only when the first error occurs.

FIRST DIFFERENCE FOUND BETWEEN FILES

RECORD <record-number>, BYTE <offset> OF <file-name1>
RECORD <record-number>, BYTE <offset> OF <file-name2>

The following warning messages can be displayed before comparison of records of two files begins.

1. WARNING – FILETYPES OF <filename1> AND <filename2> ARE DIFFERENT

   if the file types of the two files are different.

2. WARNING – RECORD SIZE OF <filename1> AND <filename2> ARE DIFFERENT
   – ONLY <X> RECORDS WILL BE COMPARED

   if the default or specified record sizes of the records, when X is the smaller of the two record sizes, will be compared.

3. WARNING – SPECIFIED KEY SIZES FOR <filename1> AND <filename2> ARE DIFFERENT
   – ONLY <X> BYTES WILL BE COMPARED

   If the specified comparison key length of one file differs from the default or specified record size or specified comparison key length of the other, only the first X bytes from the specified or default offsets will be compared. The utility continues in each case.

Examples

To compare file PQ60R on the system disk with file PQ60RS on disk PRB3:

```
COMPARE PQ60R WITH PRB3/PQ60RS
```

To compare the groups of files beginning with AR and the files A27Q on disk ARBK1 and ARBK2:

```
COMPARE ARBK1/AR= WITH ARBK2/AR=,
   ARBK1/A27Q WITH ARBK2/A27Q
```

To compare the file IV20F on the system disk with the file of the same name on disk I32, with realignment:

COMPARE IV20F WITH I32/IV20F R

Format Of Printer Output

If corresponding records are different, the following is printed:

NOTE

The hex-byte of @00@ is printed as "00".

"<file-name1>" denotes the first specified file name;
"<file-name2>" denotes the second specified file name;
"comparison key" denotes the specified part of a record or the complete record by default.

When using the NOPRINT option, SPO messages remain unchanged.

1. DIFFERENCE(S) FOUND FROM BYTES @'n'@ and @'m'@ where n is an offset in the record of <file-name1>, and m is an offset in the record of <file-name2> (in hexadecimal) of the first pair of differing bytes.

2. Either:

   RECORD. 'n' OF FILE <file-name1> COMPLETE RECORD – (if whole record in <file-name1> is being compared)

   or

   RECORD. 'n' OF FILE <file-name2> COMPARISON KEY – (if only part of a record in <file-name1> is being compared)

3. @'n'@ 'hex-string' " 'ASCII character string' "

   where n is the offset within the record (in hexadecimal), hex-string is the first 32 byte section of the comparison key in hexadecimal, and character-string is the same bytes in ASCII. This line is repeated until the whole record or comparison key has been printed. The last line is truncated if it represents less than 32 bytes.

   NOTE

   Only the bytes actually being compared are printed.

4. RECORD 'n' OF FILE <file-name2> DIFFERENCE(S) –

5. Line(s) representing section(s) of the comparison key of records in the same format as 3.

   If a comparison key has been specified for one or both files, and the offset differs for each file, then a map of the comparison key from the record in <file-name2> is printed; the record offsets at the beginning of each line section differ from those in 3, but each line corresponds to the line in the same ordinal position in 3.

   If the offset for the comparison key in each file is the same (this includes the whole of each record being compared), then only line(s) representing sections of the comparison key which contain one or more differing bytes will be printed; the record offsets at the beginning of each line correspond to the line with the same record offset in 3. If one or more sections have

no differing bytes, then "NO DIFFERENCES IN SECTION(S) NOT PRINTED" is printed at the end of 5.

In lines which are printed, only bytes which differ from 3 are actually printed; ".." (hex) and "." (ASCII) is printed in place of the bytes which do match. This means that all differing bytes can be spotted by the user.

Comparison of groups of files works as in the following example:

Assume DISK1 contains the files A, B, C, D, AB, AC, ABC, BC.

Assume DISK2 contains the files A, B, C, D, AB, AC, ABC, BC, BD, EF.

Then
COMPARE DISK1/= WITH DISK2/= compares all files on DISK1 with the corresponding files on DISK2.

But
COMPARE DISK2/= WITH DISK1/= compares files on DISK2 with the corresponding files on DISK1, and will fail to find DISK1/BD and DISK1/EF.

Similarly,
COMPARE DISK1/A= WITH DISK2/A= compares files A, AB, AC and ABC on DISK1 with the corresponding files A, B, AC and ABC on DISK1 with the corresponding files on DISK2.

Also
COMPARE DISK1/A= WITH DISK2/AB= compares the following pairs of files:
DISK1/A with DISK2/AB,
DISK1/AB with DISK2/ABB, (not found)
DISK1/AC with DISK2/ABC,
DISK1/ABC with DISK2/ABBC (not found)

R Option

The realignment option works in the following manner:

If three consecutive records fail to compare then an attempt is made to compare the third record of the second file with the next two records of the first file.

If all these five comparisons fail then an attempt is made to compare the fifth record of the first file with the fourth, fifth, sixth and seventh records from the second file.

If this comparison fails, then the comparison is terminated with an appropriate message (see later).

If a correct comparison occurs at any stage, then the compared records are used as synchronization for restarting normal comparisons.

For example, consider FILE1 containing 10 records A, B, C, D, E, F, G, H, I and J, and FILE2 containing twelve records K, L, M, N, O, P, Q, R, S, T, U and V.

The utility compares record A with record K, then B with L, then C with M. If all these comparisons fail, then if realignment is specified record M is compared with records D and E. If this also fails, record E is compared with records N, O, P and Q. If none of these compare, the comparison is terminated.

Note that if there is a missing record in one file, and realignment is NOT specified, a comparison error will arise on every succeeding record until end-of-job.

Additional Capabilities

Further features in this utility are summarized in the railroad chart given in figure 4-2, which gives the complete input specifications.

For B 900 systems, the utility attempts to open SYSMEM on all PPIT listed units for directory scanning and searches for a PPIT entry with a tag of @20@ for the system pseudo disk-name.

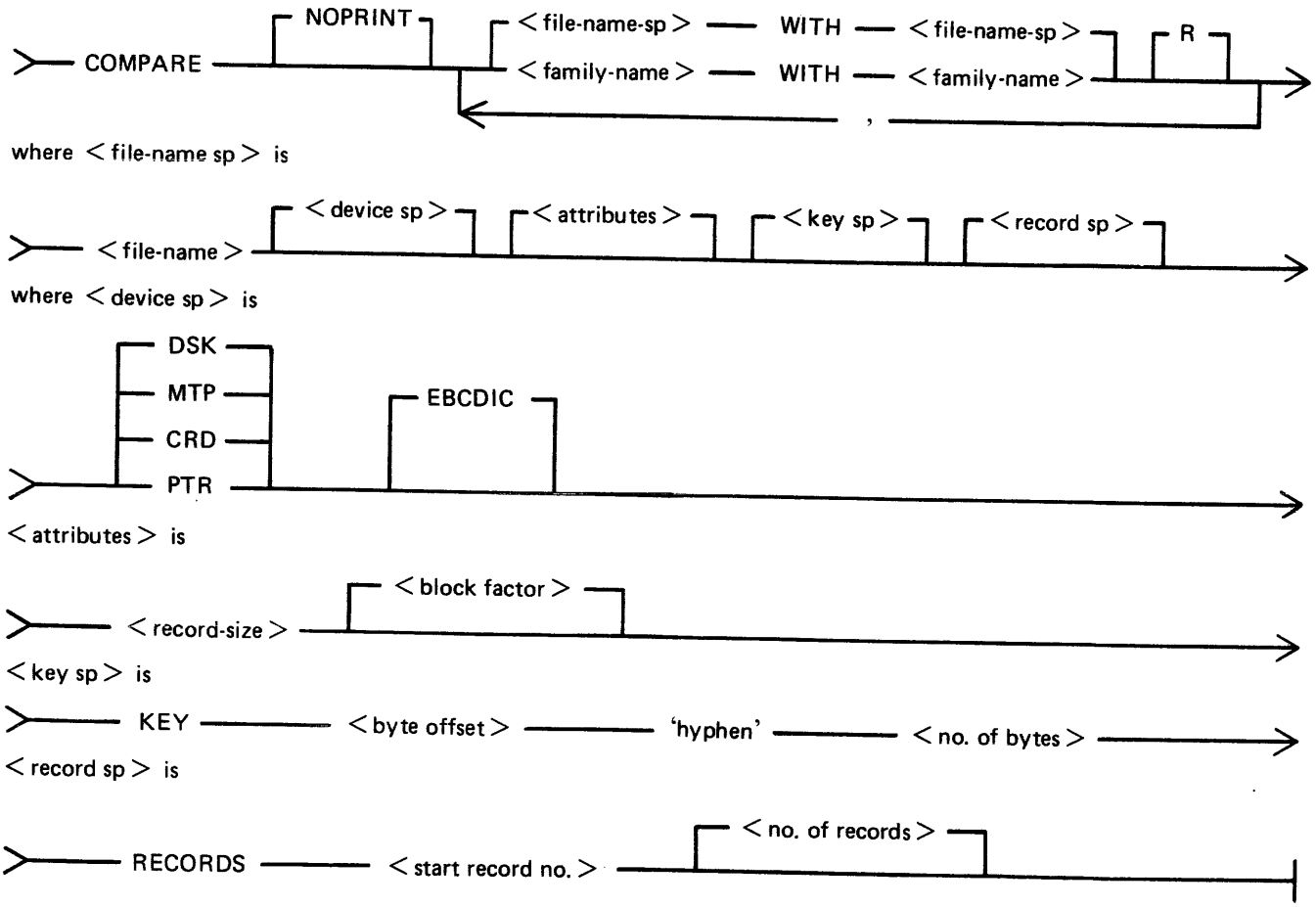


Figure 4-2. Railroad Chart for COMPARE Utility

Non-Disk Devices

Files on devices other than disk may be compared by following the file name by one of the following keywords:

CRD – any 80-column or 96-column card device
PTR – any paper tape input device
MTP any magnetic tape or cassette device
DSK any disk device (the default; this keyword is for documentation only)

Examples

To compare records on a cassette file ARDUMP/FILE020 with a file AR578QQ on disk WDSK:

```
COMPARE ARDUMP/FILE020 MTP WITH WDSK/AR578QQ
```

(note that the two-part name is valid for multi-file tapes or cassettes, refer to section 2 for naming conventions).

To compare two card files DAT1 and DAT2:

```
COMPARE DAT1 CRD WITH DAT2 CRD
```

Record and Block Sizes

The record size (and the number of records per block) may be specified after the file name and device keyword if applicable.

Examples:

To compare a system disk file CU265 with a magnetic tape file TPF, treating data blocks on the tape as 80-byte records blocked 9 records to a block:

```
COMPARE CU265 DSK WITH TPF MTP 80 9
```

To compare a system disk file SCR01 containing 90-byte records with a system disk file SCR02 containing 180-byte records, but reblocking the second file as 90-byte records:

```
COMPARE SCR01 WITH SCR02 90 2
```

Note that if the records to be compared are of different lengths, and reblocking is not specified, then only the number of characters in the shorter record are compared.

If EBCDIC is used, the file will be translated from EBCDIC on input.

The option KEY allows the comparison to be done only on the field defined, the remainder of each record will be ignored. The first number is the offset of the field within the record, the second is its length. If two files have keys of different lengths, the shorter length will be assumed for both the files. A hyphen must be specified between the offset and length:

NOTE

The EBCDIC option is only applicable with non-disk devices.

Examples:
Compare fields starting at byte 11 for 4 characters of FILE1 with FILE2

```
COMPARE FILE1 KEY 10-4 WITH FILE2 10-4
```

The option RECORDS allows the comparison to be done only on the records specified. The first number is the starting record number and the second number is the total number of records available for comparison. No other record will be read from that file.

Example:

Compare records 12, 13, 14 of FILE1 with records 10, 11, 12 of FILE2.

```
COMPARE FILE1 RECORDS 12      3 WITH FILE2 RECORDS 10      3
```

Limitations

The maximum record size is 1024 bytes. If a file exceeds this record size, it may be compared by reblocking. For example, a file with record size of 1200 can be compared by reblocking as 600 bytes blocked 2, or as 300 bytes blocked 4. The higher the blocking factor, the slower the comparison will be. (If the record size is a prime number P, it can be reblocked as 1-byte records blocked P.)

The use of a star-file terminates the list of pairs of files to be compared. For example,

```
COMPARE A= WITH DK2/A=, X= WITH DK2/X=,
    STFILE, B= WITH DK2/B=
```

will compare A=, X=, and all files mentioned in the file STFILE, but will ignore the comparisons of B=

Output Messages

Message	Possible Causes	Suggested Action
<file name> NOT COMPARED	The utility could not compare the specified file. The reason is given in the second message.	Look for the reason in the next message displayed.
FAMILY <family name> NOT COMPARED	The utility could not compare the specified files. The reason is given in the second message displayed.	Look for the reason in the next message displayed.
-FAMILY MAY BE INCOMPLETE	The utility is requested to compare a family of files on a pseudo disk but at least one physical disk is not ready.	Ready the drive using "RY" command and re-enter.
-HARDWARE ERROR DETECTED WHILE READING PSEUDO-PACK TABLE ON DISK <disk-name>	The utility detected an error in the PPIT of the specified disk.	Check integrity of the specified disk.

(continued)

Message	Possible Causes	Suggested Action
-FAMILY <family name> MAY BE INCOMPLETE	The utility is requested to compare a family of files on a pseudo disk but at least one physical disk is not ready.	Ready the drive by using "RY" command and re-enter.
FIRST DIFFERENCE FOUND BETWEEN FILES	The utility has found differences. The next message displayed indicates where these differences occur.	Look for the reason in the next message displayed by utility.
-SPECIFIED KEY <key spec> IS INVALID OR BEYOND END OF RECORD	The key specification in the input message is not suitable for this file.	Check input and re-enter.
-SPECIFIED RECORD RANGE IS INVALID	The numbers following the keyword RECORDS in the initiating message are invalid.	Check input and re-enter.
-RECORD SIZE IS GREATER THAN THE UTILITY LIMIT OF <record size>	The utility is not able to handle record sizes greater than 1024.	None. Utility limitation.
HARDWARE ERROR DETECTED WHILE READING RECORD <number> OF <file-name>	The utility has encountered a read error on the specified record.	Check integrity of the specified file.
UNABLE TO CONTINUE COMPARING <file-name> AND <file-name>, CANNOT REALIGN FILES	The utility is not able to realign the specified files.	None.
END OF FILE REACHED ON <file-name> BEFORE <file-name>	The file specified first is smaller than the second file.	None
<filename> COMPARED WITH <filename>, <number> RECORD(S) ARE DIFFERENT	Self-explanatory.	None.
-SPECIFIED RECORD <number> IS BEYOND END OF FILE	The record number specified in the initiating message is not in the file.	Check input.
FILE TYPES OF <file name> AND <file name> ARE DIFFERENT	The specified files are not of the same types.	None.
RECORD SIZE OF <file name> AND <file name> ARE DIFFERENT	The specified files do not have the same record size.	None.
SPECIFIED KEY LENGTHS FOR <file name> AND	Self-explanatory.	None.

(continued)

Message	Possible Causes	Suggested Action
<file name> ARE DIFFERENT		
-ONLY <number> BYTES WILL BE COMPARED	Self-explanatory.	None.
-SPECIFIED RECORD OR BLOCK SIZE IS INVALID	The utility is not able to handle the specified record or block size.	None. Utility limitation.
RECORD SIZES OF <file-name AND <file-name> ARE DIFFERENT	The files to be compared have different record sizes. Only a part of the record will be compared.	None.
-ONLY "n" BYTES WILL BE COMPARED	Self-explanatory.	None.

COPY (File Copy)

This utility allows the operator to copy files from one medium to another.

See figure 4-3 for syntax.

If, as a result of copying a file, a duplicate filename would be created, the original file on the destination disk is removed automatically.

If the option "& COMPARE" is specified, the utility will compare the newly created output file with the specified input before closing the output file.

If errors are detected it will output:

COMPARISON ERROR DETECTED WHILE CHECKING <output file>
<input file> NOT COPIED

If the file being copied is a keyfile and the <KEY> option is used, the keyfile is copied and the new keyfile refers to the original data file.

If the file being copied is a keyfile and the <BOTH> option is used, the keyfile and the corresponding data file are copied. The data file is given the keyfile name with the letters, "QQ" appended. The new keyfile is made to refer to the new data file name.

If the file being copied is a keyfile and neither <KEY> nor <BOTH> options are used, only the corresponding data file is copied. The records of the new data file are copied in keyfile order.

Examples:

To copy a file called AR200 from the system disk to a disk called ARBU:

COPY AR200 TO ARBU/AR200

To copy files called AR200 and AR300 from the system disk to a disk called ARBU:

COPY AR200 TO ARBU/AR200, AR300 TO ARBU/AR300

To copy a file called APTASK from the system disk to APBU, changing its name to APTASKB:

COPY APTASK TO APBU/APTASKB

To copy all files beginning with letters "PR" from disk PR2 to disk called PRBU:

COPY PR2/PR= TO PRBU/PR=

Copying Keyfiles

Assume there is a keyfile called PR200K which refers to a data file called PR200.

The statement
COPY PR200K <KEY> TO PRB/PR200K will create a new keyfile PR200K on disk called PRB which references the original data file, PR200, on the system disk.

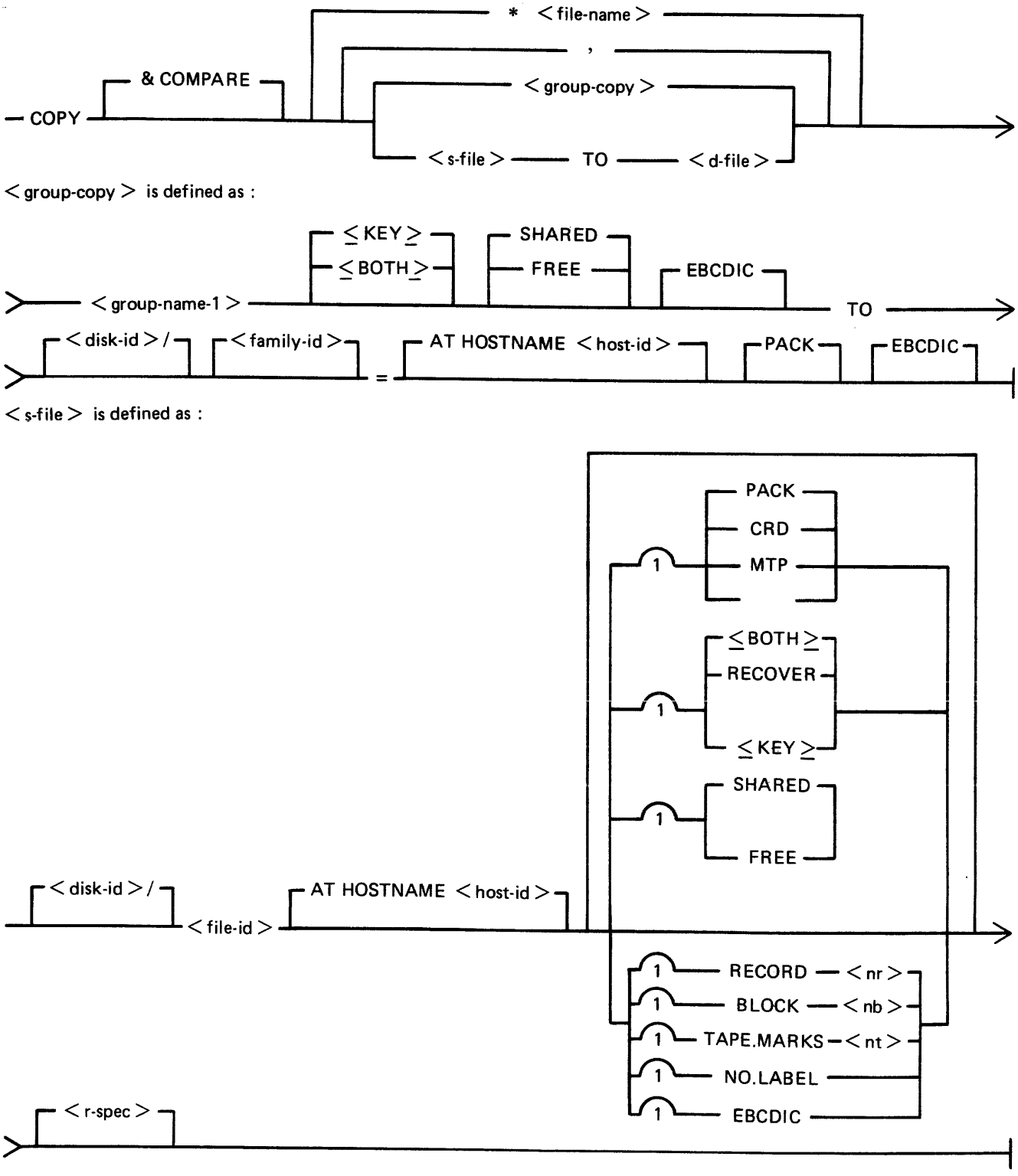
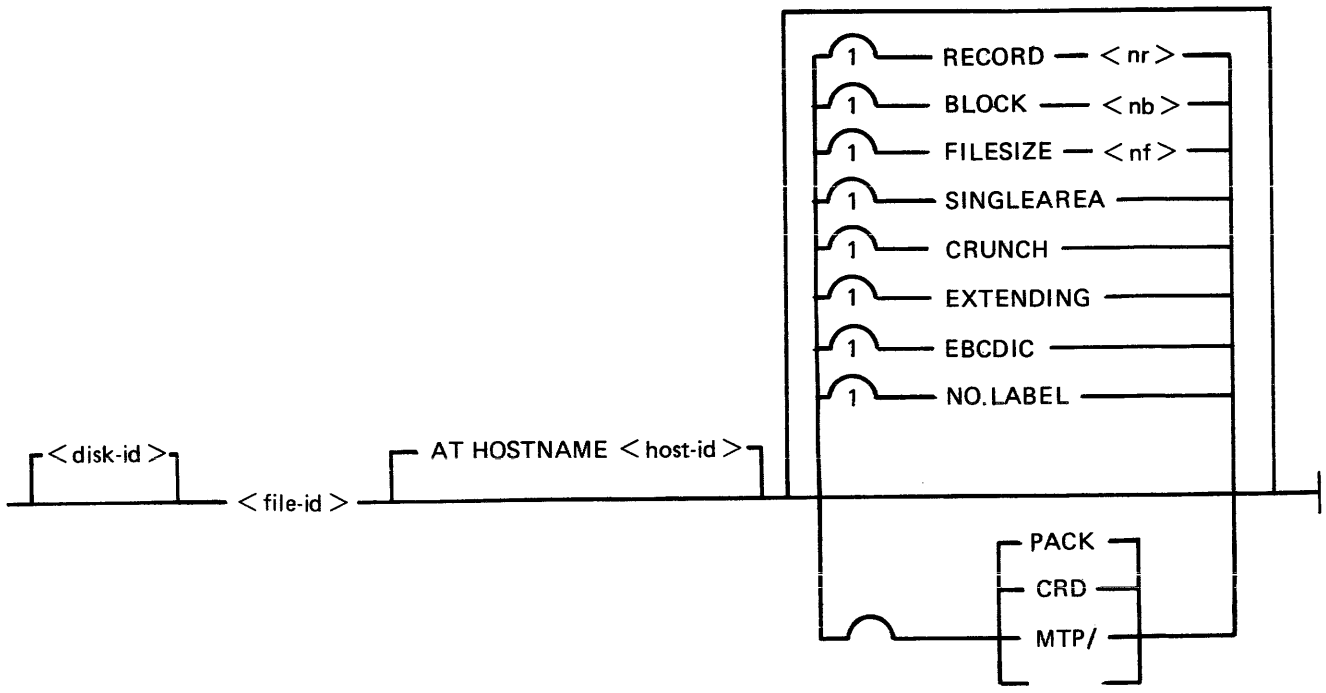


Figure 4-3. Railroad Chart for Copy Utility (Sheet 1 of 2)

< d-file > is defined as :



< r-spec > is defined as :

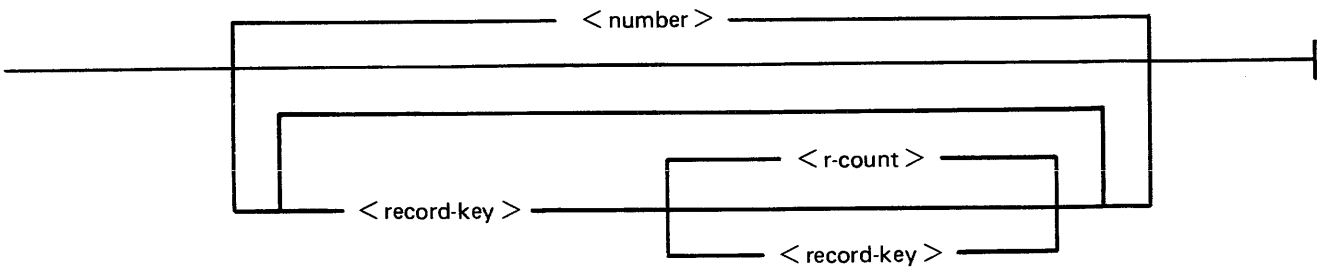


Figure 4-3. Railroad Chart for Copy Utility (Sheet 2 of 2)

The statement

COPY PR200K <BOTH> TO PRB/PR200K will create a new keyfile and data file on disk called PRB. The name of the new data file will be PRB/PR200KQQ and the keyfile (PRB/PR200K) will refer to this new data file.

The statement

COPY PR200K TO PRB/PR200K will create a new datafile PR200K on the disk PRB. No new keyfile will be created but the records in the new data file will be created in key order according to the keyfile.

Additional Capabilities

Further features in this utility are summarized in the railroad chart given in figure 4-3, which gives the complete input specifications.

NOTE

The facility to copy from or to an EBCDIC file is not supported for disk files.

Non-disk Devices

Files may be copied to and from media other than disks. Abbreviations for the valid devices are as follows:

MTP – magnetic tape or cassette
CRD – punched card
PTP – paper tape

Examples:

To copy a cardfile called PRFILE to a disk called PRBU:

```
COPY PRFILE CRD TO PRBU/PRFILE
```

To copy a disk file called PR300 to a single-file magnetic tape:

```
COPY PR300 TO PRTAPE MTP
```

NOTE

This tape is in "COPY" tape format, not "LOAD/DUMP" format. To access this tape file again it will have to be placed on an appropriate device by the "COPY" utility, not "LOAD/DUMP".

To copy a cardfile called PRFILE to paper tape:

```
COPY PRFILE CRD TO PTFILE PTP
```

NOTE

Paper tapes are always "unlabelled", and when accessing it, MCP will issue an appropriate message requiring an "AD" intrinsic response from operator. See "AD" intrinsic.

Unlabelled Tapes

Input tapes with no CMS labels ("unlabelled" tapes) may be accessed by the COPY utility.

In addition, unlabelled output files may be created which use the Disk File Header attributes of the input disk file.

The NO.LABEL option allows the copying of unlabelled files. On recognizing an unlabelled file, the MCP will print "<mix-number>/COPY <28> WAITING SPURIOUS/<tape-name> AT DEVICE ASSIGNMENT REQUIRED". The operator must then respond with an appropriate "AD" input message (see "AD") to identify the unlabelled file.

The end of file recognition for unlabelled files is determined by tapemark count. The TAPE.MARKS option allows the operator to specify the total number of tapemarks which will indicate end of file to the utility when copying an unlabelled file. The default value is 2. Each tape mark which is encountered will contribute to this total. Therefore, a standard labelled CMS file will be copied up to, but excluding, the trailing label if NO.LABEL is specified by itself. (The standard CMS labelled tape format is "label; tape mark, data, tape mark, label", see CMS MCP Reference manual.) The operator must be aware of the format of any file which is to be copied when using the NO.LABEL option.

If the RECORD size is not 180 bytes, refer to the section on Record/Block modification.

Example:

To create a disk file called EMPL from first file of a magnetic tape with non-standard label (the format being: LABEL, TAPEMARK, DATA, TAPEMARK):

```
COPY TP MTP NO.LABEL TAPE.MARKS 2 TO EMPL
```

NOTE

MCP will issue a message asking for unlabelled tape TP. Operator must respond with "AD" input. Additionally, the first record of file EMPL will contain a copy of the non-standard label. An unlabelled tape may be created by specifying NO.LABEL with MTP as the output file attribute.

Record and Block Sizes

Record and/or block sizes may be modified for all file types, input and output.

The number of bytes in the record or block is specified using the corresponding "numbers". The record and block sizes of input files are always taken from the file itself (Disk File Header). Record and block sizes of non-disk input files are determined as follows:

Record size:

If RECORD is specified, "number" becomes the new record size.
If RECORD is not specified record size defaults (see below).

Block size:

If BLOCK is specified, "number" becomes the new block size.
If no BLOCK specified, but RECORD is specified, record size becomes new BLOCK size.

If neither BLOCK nor RECORD is specified, Block Size defaults (see below).

Default Values:

Output disk = same as input disk.
Input labelled tape/cassette = from tape label.
Input unlabelled tape/cassette = 180 bytes.
Cards = 80 or 96 bytes, depending on device.

If the record size is increased, then the additional bytes will be filled with spaces if the input file is a source or data file, or with binary zeros for any other type of file.

Example

To copy an 80-column card file labelled PROGSRC to a disk file called PROGSRC on a user disk "USR", and make the record size and block size of the disk file 80 bytes and 720 bytes respectively:

```
COPY PROGSRC CRD TO USR/PROGSRC RECORD 80 BLOCK 720
```

To copy a disk file PRBU/PR300 to magnetic tape with large blocks suitable for tape media:

```
COPY PRBU/PR300 TO PRTAPE MTP RECORD 180 BLOCK 1800
```

File Size

The "FILESIZE attribute" of a disk file may be specified for the output disk file. Note that only assigned areas are copied. This feature does not increase disk space at the time of copying, but allows programs to add further records if required. At that time disk space may be needed. The maximum filesize you may declare depends on the type of disk unit the file has to be on, the record and block size of the file and whether or not the single area option is on for the file.

Example 1:

A file with a record size of 180 and a blocksize of 80 to be copied as as a multi-area file to a B 9493-80 (allocation unit is 8) will have a maximum filesize of 1048448 records. When the same file is to be copied to a B 9493-10 (allocation unit is 1) then the maximum filesize is 1048560 records. A file with a record size of 80 and a blocksize of 720 to be copied as a multi-area file to a B 9493-80 will have a maximum filesize of 1048320 records.

Example 2:

To copy FILE1 and increase its "FILESIZE" to 1500, replacing the original by the copy:

```
COPY FILE1 TO FILE1 FILESIZE 1500
```

Single Area

The "SINGLEAREA" attribute may be specified for the output disk file. This ensures that the new file will occupy a single disk area.

Example:

```
COPY FILE2 TO FILE2 SINGLEAREA
```

Crunching Files

The "CRUNCH attribute" may be specified for the output file. This causes any unused disk space at the end of the file to be returned to the system.

Example:

```
COPY PRB/PR200 TO PRB78/PR200 CRUNCH
```

WARNING

A file cannot be "uncrunched" once it is crunched. This means it cannot be extended. It can only be used for inquiry. This option is therefore useful for storing history files.

Extending Disk Files

Records can be added to the end of an existing disk file with the option "EXTENDING". The existing file must have identical attributes to the file being copied.

Example:

A data file called DFTUES was created with Tuesday's data. To add this data to the end of a file called DFMON (containing Monday's data):

```
COPY DFTUES TO DFMON EXTENDING
```

(Note the size of DFMON must be large enough to contain all required records).

EBCDIC

The EBCDIC option may be specified for either the input or output file. When an ASCII input file has to be copied to an EBCDIC output file, use the following syntax:

```
COPY <file1> TO <file2> EBCDIC
```

When an EBCDIC input file has to be copied to an EBCDIC output file enter:

```
COPY <file1> EBCDIC TO <file2> EBCDIC
```

Free And Shared

Files which are in use for output (but not locked) by another program or are "shared" may only be copied by specifying "FREE" or 'SHARED" as appropriate.

Recover

If a file contains records which are unreadable due to bad disk areas, that file may be copied only by specifying RECOVER as an input file parameter. Only source, data, EBCDIC data and binary data files may be recovered.

Unreadable records in the input file will be replaced by ASCII 'X' character filled records in the output file, unless the file is EBCDIC, in which case an EBCDIC 'X' will be used. Copy will report the record number of each substituted record.

Selected File Copy

Selected record numbers from the input file may be copied.

Example:

To copy 500 records starting at record #1200 from file FILE1 to file FILE2:

```
COPY FILE1 1200 500 TO FILE2
```

NOTE

Pairs of numbers may be specified within each pair; the first number specifies a relative record number and the second specifies number of records to be copied. If an extra number is specified, the last number specifies copying from that record to the end of the file.

Example:

To copy records 100 to 149, 300 to 499, and 1000 to end of file:

```
COPY FILE1 100 50 300 200 100 TO FILE2
```

Selected Index File Copy

For indexed files, copying of records can be selected based on content of the key. There are two options: the number of records or an ending key value can be specified.

Examples:

PQR is a keyfile containing personnel records. To copy 15 records from the corresponding data file starting from the record with personnel number 01786 to a data file, PSNL:

```
COPY PQR 01786 15 TO PSNL
```

Using the same keyfile, to copy all data records from personnel number 01786 to 01800 to data file, PSNL:

```
COPY PQR 01786 - 01800 TO PSNL
```

NOTE

The second option is specified by the hyphen in the COPY statement. Note that at least one space is required before and after all key values (personnel number in this case).

The above keys can optionally be expressed in hexadecimal format at follows:

```
   COPY   PQR   @06FA@   15 TO PSNL
or COPY   PQR   @06FA@   -@0708@
```

Save Factor

New magnetic tapes are given a save factor of 999.

Copying Files To And From Remote CMS And Non-CMS Systems

This facility may be used on a B 900 system which is a host in a BNA network.

Disk files may be copied to and from remote host systems by specifying: "AT HOSTNAME <host-id>" after the appropriate file or family name.

NOTE:

1. A family of files can be copied to but not from a remote host non-CMS system.
2. If files have to be copied to or from a disk pack on a remote host non-CMS system, "PACK" must be specified following the host-id of that system.
3. Index files cannot be copied from one host system to another.

Output Messages

Message	Possible Causes	Suggested Action
<file-name> COPIED TO <file-name>	COPY successful.	None.
<file-name> REMOVED	COPY successful. If a duplicate file is encountered by COPY, it is automatically removed.	None.
-<filename> CONTAINS NO RECORDS	COPY successful, but no records are actually written to the new file.	Check for correct file-name.
<filename> NOT COPIED	The specified file is not copied. The reason is given in the next error message displayed.	Look for the reason in the second error message displayed.
FAMLIY <family name> NOT COPIED	The specified family is not copied. The reason is given in the next message displayed.	Look for the reason in the second error message displayed.
-INVALID FILE SPECIFICATION FOR <file-name>	The specification, or combination of specifications. given is not permitted.	Check input for correct specifications.
-FILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	The specified file is in use by another program.	Use "SHARED" or "FREE" option to copy the file.
-REQUESTED FILES WERE NOT FOUND	The specified files are not found in the directory of the specified disk.	Check input for correct disk and file-names.
-RECORD SIZE EXCEEDS MAXIMUM SET FOR THIS RUN, PLEASE RESUBMIT	Copy has encountered a file with a record size greater than expected.	Check input for correct record size.
- FILE NAME GREATER THAN 12 CHARACTERS WOULD BE GENERATED	Copy of a group of files was requested. At least one filename that was to be generated would have been more than 12 characters long, maximum length of file-names is 12 characters.	Check input for correct family name.
COMPARISON ERROR DETECTED WHILE CHECKING <file-name>	File not copied because of an error found when comparing the output file with the input file.	None.
-DISK <disk-id> IS NOT A CMS DISK	A family has been specified for copying from a NON-CMS system. Families cannot be copied.	Do selective copies of every file mentioned in the input file specification.

(continued)

Message	Possible Causes	Suggested Action
-NO RECORDS IN KEYFILE <file-name>	<BOTH> option was used. Utility was not able to access a data file because of some failure in the keyfile.	Check input for correct syntax or use SORT to create a new keyfile.
-KEYFILE OR DATAFILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	Another program has opened the file for output use.	Use "FREE" or "SHARED" option in initiating message if other program does not use the file with LOCK-ACCESS.
-KEYFILE OR DATAFILE NOT FOUND	Specified file is not in the directory of the specified disk.	Check input for correct disk and file-names.
-KEYFILE OR DATAFILE IS ON A DISK WHICH IS CURRENTLY LOCKED BY ANOTHER PROGRAM	Specified keyfile or datafile is on a disk whose SYSMEM is opened with lock-access by another program.	Wait until the other program is finished and re-enter.
-ASSOCIATED DATAFILE <file-name> NOT FOUND	The datafile which belongs to the specified keyfile is not in on line.	Bring associated datafile on line and re-enter.
-HARDWARE ERROR DETECTED WHILE READING KEY <key>	Error encountered when reading specified key.	Check datafile for errors or use "RECOVER" for copying the readable records.
-HARDWARE ERROR DETECTED WHILE READING RECORD <record number>	Error encountered when reading specified record.	Check datafile for errors or use "RECOVER" for copying the readable records.
-HARDWARE ERROR DETECTED WHILE WRITING RECORD <record number>	Error encontered when writing specified record.	Check output medium.
-OUTPUT FILE TOO SMALL	Filesize of output file is too small to contain all the records of the input file.	Re-enter the COPY with suitable FILESIZE option.
-<file-name> NOT FOUND	A file has been specified for extending but has not been found.	Check input and re-enter. Check for correct medium.
-OUTPUT FILE <file-name> NOT COMPATIBLE	A file has been specified for extending and is not of the same type or record size as the input file.	Check that input and output file are the same type and have the same record size.
-KEYFILE OR DATAFILE IS ON A DISK WHICH IS CURRENTLY LOCKED BY ANOTHER PROGRAM	Self-explanatory.	None.
-FILE IS NOT AN EBCDIC FILE	A disk file has been specified for translation from EBCDIC which is not an EBCDIC data type file.	None.

(continued)

Message	Possible Causes	Suggested Action
FAMILY MAY BE INCOMPLETE	A family is specified for copying from an unrestricted pseudo-disk and at least one of the physical disks is not available.	None.
FAMILY FOR COPYING MAY BE INCOMPLETE	A family is specified for copying from an unrestricted pseudo-disk and at least one of the physical disks is not available.	None.
<file-name> CANNOT BE COMPARED USING THE SPECIFIED OPTIONS	The specified file is not a diskfile, or is an indexed file, or inappropriate options have also been specified.	None.
END OF FILE REACHED ON <file-name> DURING RANGE FROM RECORD <record number>	End-of-file encountered while section of file indicated by record-number number is being copied.	None.
END OF FILE REACHED ON <file-name> DURING RANGE FROM <key>	End-of-file encountered while section of file indicated by key is being copied.	None.
<file-name> RANGE FROM RECORD <record-number> IGNORED SINCE OUT OF SEQUENCE	A non-disk file or null-key keyfile is being copied with record selection, the selected ranges must be in sequential order.	None.
<file-name> RANGE FROM RECORD <record-number> IGNORED SINCE END OF FILE REACHED	Once the end of null-key keyfile or a non-disk file is reached, no further selections can be copied.	None.
SUBSTITUTE RECORD <record-number> IN FILE <file-name> WILL CONTAIN X CHARACTERS	The RECOVER option has been used and an unreadable record has been found at <record-number>, a substitute "X"-filled record has been written to the output file.	Check output file for "X"-filled records and replace them with the original contents using the the FILEUTL utility.
-RECORD SIZE LIMIT IS 10000	The utility cannot handle record sizes greater than 10000.	None. Utility limit.
-INDEXED FILE ACCESS NOT PERMITTED ON A REMOTE HOST	It is not permitted to copy indexed files with the AT HOSTNAME option.	None.
-FILE OF THIS TYPE CANNOT BE COPIED BETWEEN HOST SYSTEMS	It is not permitted to copy non data files with the AT HOSTNAME option.	None.

(continued)

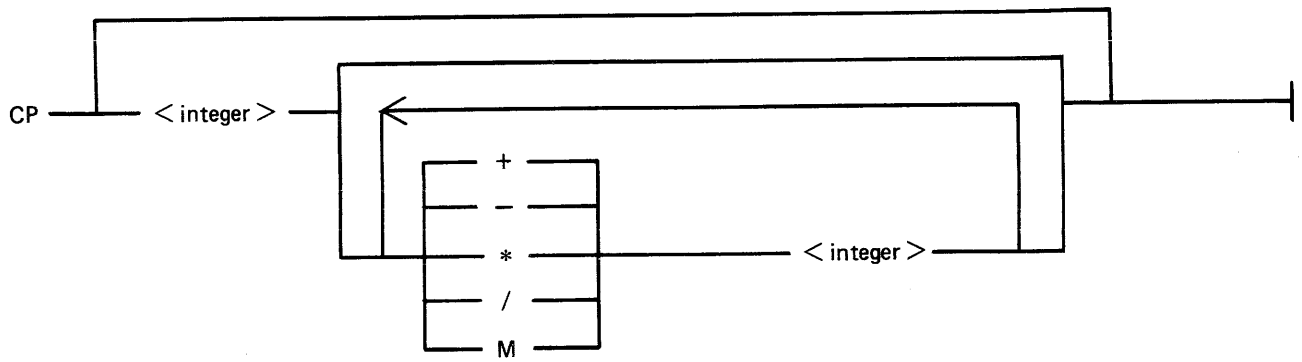
Message	Possible Causes	Suggested Action
ILLEGAL PARAMETER LIST <file-name>	A file-name greater than seven characters has been entered and COPY to TAPE has been requested.	Re-enter the file-name using not more than seven characters.

CP (Compute)

This utility allows simple computations to be made, with the answer displayed in decimal and hexadecimal. Input may be either decimal or hexadecimal. Hex values must be enclosed in @ symbols.

The utility may be initiated with a single computation to perform, in which case it will do the calculation and terminate. If no calculation is initially provided, the utility issues an ACCEPT to enable the computation to be entered. In this case the utility will do the calculation and then issue further ACCEPTS until a null input to the ACCEPT is given.

Format:



The numbers accepted are any decimal or hex values in the range

0 999999999999999[@38D7EA4C67FFF@]

or the negative equivalent.

Parentheses are not allowed. The calculation is performed on a strictly left-to-right basis. The operators +, –, *, /, and M are for addition, subtraction, multiplication, division and modulus division (the result is the remainder) respectively.

Examples:

To compute the hexadecimal value of the decimal number 12345:

CP 12345

To compute the value of the complex expression:

CP 555 * 3 + 2-100 * 2/5

(Note: the result of the above is 626, because the calculation is done strictly left-to-right).

Output Messages

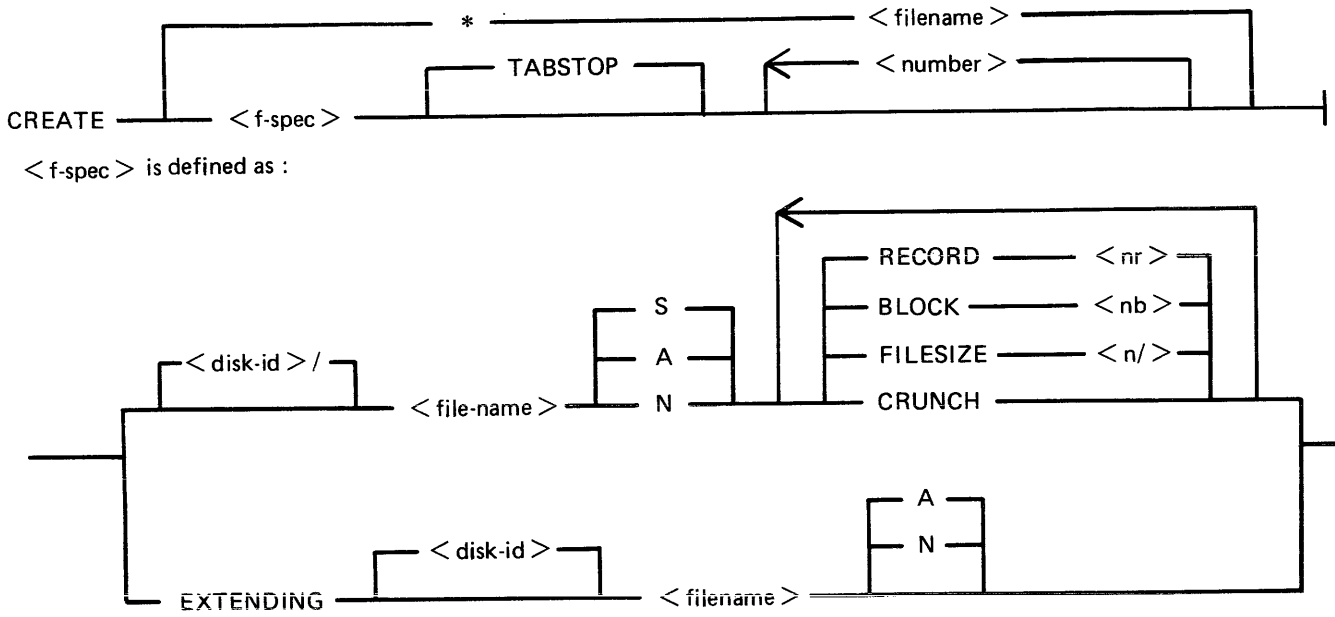
Message	Possible Causes	Suggested Action
– <number> IS NOT A VALID NUMBER	Two consecutive operators (for example +) have been entered with no number between.	None.

(continued)

Message	Possible Causes	Suggested Action
– <number> IS NOT A VALID HEXADECIMAL NUMBER	Illegal symbol; missing an @ symbol at front or back of a hexadecimal value.	Re-input.
– <operator> IS NOT A VALID OPERATOR	Illegal symbol; or an operator (for example, +) has been omitted.	Re-input.
OVERFLOW DETECTED BY CP DURING CALCULATION	Self-explanatory.	None.
DIVISION BY ZERO DETECTED BY CP	Self-explanatory.	None.
– NUMBER EXCEEDED 99999 99999 99999 (@ 03807EA4C67FFF@)	Self-explanatory.	None.
CP: RESULT – <result>	Normal output.	None.

CREATE (Create Disk File – B 90 Only)

This utility allows the operator to create or extend data or source disk files. The "AMEND" and "UPDATE" utilities use many similar features.



When creating a new disk file, certain attributes may be specified.

If the S option is selected, a source file will be created using alphanumeric input. If the A option is used, a data file will be created using alphanumeric input. The N option creates a data file with hexadecimal (numeric) input. If none of these is selected, S is assumed. Alphanumeric input is accepted as typed, but numeric (hexadecimal) input requires two characters (0-9, A-F) for each byte of the record.

The RECORD option allows the operator to specify the number of characters per record of the new file. If no record size is specified, a record size of 80 bytes is assumed for source files, and 180 bytes for data files.

The BLOCK option allows the number of characters per block of a new file to be defined. The defaults are as follows:

If RECORD size was specified but no BLOCK, BLOCK size will equal RECORD size.

If neither RECORD nor BLOCK is specified, RECORD size will be 80 bytes for source files and 180 bytes for data files; BLOCK size will be 160 bytes for source files, 180 bytes for data files.

The FILESIZE option allows the maximum number of records likely to be written to the new file. This is useful in allocating only as much disk space as required by the file. Once the FILESIZE has been specified for a file, that file can never be extended beyond that number of records. However, the COPY utility may be used for increasing the FILESIZE of an existing file. The default is 2048 records.

The CRUNCH option allows the operator to specify that the new file should occupy the minimum area of disk, but never be extended.

The numbers specified for the "numbers" option may be used to set "tab" positions within the record (similar to setting "tabs" on a typewriter). If tabs are set, the operator may input data, press OCK1, and the utility will reposition the print mechanism to the next tab position within the record, and await data input. During this repositioning, CREATE will fill all character positions left unspecified in the record with a "filler" (ASCII space for source input, ASCII zero for alphanumeric input, and binary zero for numeric zero). The record length plus one will be used as a termination tab position, whether or not other tab positions are specified.

CREATE can be used for record sizes up to 500 bytes, but since the utility cannot be given input greater than the width of the console, tab positions are mandatory on files of larger record sizes. For example, a file of 180 byte records requiring alphanumeric input will require at least one tab position (for instance, at position 100). A file of 180 byte records requiring hexadecimal input will require a minimum of three tab positions (for instance at positions 50, 100 and 150). The maximum tab size is 111 in alphanumeric input and 54 in hexadecimal input. That is, the difference between two consecutive tab positions should be less than or equal to 111 in alphanumeric input and less than or equal to 54 in hexadecimal input.

By specification of TABSTOP in the initiating message, CREATE sets up tab positions coinciding with the end of the console line as well as any other tabs specified. Also, when in Record Input Mode (PK1), CREATE accepts information only up to the next tab position.

Default tab positions have been chosen to allow a maximum number of characters to be inserted on one line. CREATE uses nine for the record number and 110 for the contents of the record. In addition, manually-selected tabs may still be used.

Default tab positions for the end of the console lines for CREATE are as follows:

Source or data alphanumeric : 111 221 331 441
Data hexadecimal : 56 111 166 221 276 331 386 441 496

Examples:

	Tabs Set at:
CREATE FILEA RECORD 410 TABSTOP	111 221 331
CREATE FILEB A RECORD 500 TABSTOP 51 61 221	51 61 111 221 331 441
CREATE FILEC N RECORD 450 TABSTOP	56 111 166 221 276 331 386 441 496

The EXTENDING option is used to add records to an existing file. The attributes, such as RECORD and BLOCK sizes, are taken from the old file. The file type is also taken from the existing file. The operator may specify "A" for alphanumeric input or "N" for hexadecimal input. If neither "A" nor "N" is specified, "A" is assumed. If a file is CREATE EXTENDed, the generation number of the file is incremented by one.

Examples:

To create a source file called "ICFILE", record size 100 bytes with 5 records per block, tab position set at 65:

```
CREATE ICFILE RECORD 100 BLOCK 500 65
```

To create a source file called "ICFILE" with record size 80, block 3, and a maximum of 20 records in the file:

CREATE ICFILE RECORD 80 BLOCK 240 FILESIZE 20

To extend a source file called "ICFILE" (note: the utility will automatically prompt the operator for the next sequential record number to be created):

CREATE EXTENDING ICFILE

To create a data file called "CFILE" for hexadecimal input with tab positions set at 50, 100 and 150. (Note: Default record size is 180, block 1):

CREATE CFILE N 50 100 150

The utility operates in two modes: "RECORD INPUT" (entered through PK1) and "RECORD MODIFY" (entered through PK2).

PK1	PK2	PK3	PK4	PK5	PK6
input	modify	–	–	–	EOJ

An OCK3 option is included, to display the current tab position.

An OCK4 "help" option is provided, which will output the above options when pressed. In order to show which mode the utility was in when OCK4 was pressed, an asterisk (*) is printed next to that mode on the Help display.

The "Record Input Mode" (PK1) is used to enter new records through the keyboard. Characters are input followed by OCK1 for each tab position.

The "Record Modify Mode" (PK2) is used to make corrections to the last record input. The point in the record at which alterations are to be made is selected by typing an identifying group of characters immediately preceding the byte(s) of the record to be altered. The portion of the record to be replaced or inserted follows the identifying characters, delimited by a colon (:). If alterations are to be made at the beginning of the record, no identifying characters are necessary. A starting byte position for the identifying character string search may be specified in the console input (see AMEND for details).

If OCK1 is used to terminate input, the characters to be altered will replace the corresponding number of characters in the record.

For example, for a record containing "ABCDEF", the amendment C:XY:OCK1 will result in "ABCXYF".

If OCK2 is used to terminate input, the characters delimited by colons (:) will be inserted at the indicated point. The insertion can cause characters in the record to be moved to the right. The shifting of characters applies only to those characters from the starting byte to the next higher relevant tab position; characters beyond this tab position will not be affected.

For example, a record specified with tab positions at 4 and 8, contains "ABCDEFGHIJ". The amendment C:WXY: OCK2 will result in "ABCWXYDHIJ".

Initially the utility will be in the "Record Input Mode", and on completion of an entry in any mode, it will allow the operator to select the mode not in use, or terminate the utility (with PK6). Unless otherwise instructed, it will continue in the existing mode.

Output Messages

Message	Possible Cause	Suggested Action
– FILE DIRECTORY ON DISK <disk-name> IS FULL	Self-explanatory.	Re-run create on another disk.
UNABLE TO ACCESS <file-name>	The specified file name is not on disk or is in use by another program.	Check input for correct file-name.
– SPECIFIED RECORD OR BLOCK SIZE IS INVALID	Self-explanatory.	Check input for correct record and block size.
OUTPUT FILE TOO SMALL	Attempt was made to add records to specified file beyond its maximum file-size. File is closed with lock.	List the file to check which records have been entered: then use CREATE EXTENDING to add the desired records. Alternatively use UPDATE with the FILESIZE option.
RECORD SELECTION ERROR	Attempt was made to select a non-existent record.	Check input and re-enter if necessary.
ILLEGAL RECORD NUMBER SPECIFIED	Attempt was made to select record greater than the number of records in file, or zero.	Check input and re-enter if necessary.
RECORD REQUESTED IS BEYOND EOF	Attempt was made to select record beyond end of file.	Usually indicates update is complete; utility terminates normally, no action need be taken.
EOF REACHED DURING DELETIONS	Attempt was made to delete a record beyond end of file.	Usually indicates file update is complete; utility terminates normally, no action need be taken.
FILE TYPE IS NOT DATA	Input type A or N has been specified and the type of file given for extension is not data.	Check the file type and re-input correct file type.
E.O.F REACHED	The last logical record of the output file has been entered (applies to the utility	Select PK2 to modify record or PK6 to go to end-of-job.
ILLEGAL PARAMETER LIST – BAD ATTRIBUTE	Specified record and block sizes are incompatible or blocking factor > 255.	Check input and re-enter.
– NO OR INVALID TAB POSITIONS SPECIFIED FOR RECORD SIZE <number>	Tab positions beyond end of record were specified; or input fields larger than capabilities of console.	Check input and re-enter.
NOT HEXADECIMAL CHARACTER INPUT RESUBMIT	Character other than 0-9 and A-F was input.	Check input and re-enter.

(continued)

Message	Possible Cause	Suggested Action
ODD NUMBER OF HEXADECIMAL CHARACTERS INPUT	Warning message that an odd number of hexadecimal characters was input. When input mode is "hexadecimal", utility expects even numbers of input characters.	None. Utility accepts the input, but adds a zero onto the end (right) of the input to even it out.
INPUT ERROR – RESUBMIT RECORD MODIFICATION	Input error during "record modify mode".	Check input and re-enter record modification.
BYTE WITHIN RECORD SPECIFIED NOT FOUND	The identifying string of characters for record modification could not be found in the record specified.	Check input and terminate the entry with the correct OCK.
UNWANTED KEY PRESSED – PLEASE RE-INPUT	Invalid OCK was pressed.	Re-enter input and terminate the entry with the correct OCK.
INPUT IMMEDIATELY BEFORE PK6 HAS BEEN LOST	Characters were input immediately before PK6 was pressed to terminate the utility. These characters will not be written to the specified file.	Restart the utility using modify mode to correct this record if desired.

Note: Refer to "Common Utility Output Messages" for additional messages.

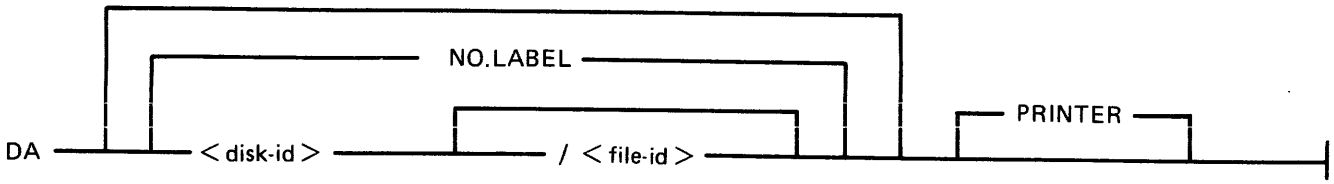
DA (Disk Analysis)

This utility allows the operator to read any portion of a CMS-format disk. It is an interactive program, with the operator entering a range of commands via the console (if the PRINTER option is not specified), or the SPO by means of an accept (AX) command if PRINTER is specified. Output is directed to a line printer when the PRINTER option is invoked.

The utility is commonly used to print the contents of the disk directory. In general, if the PD utility operates correctly for a specific disk, then DA should also run successfully. Specifically, the disk cartridge label, the name list entry and disk file header for SYSMEM must be intact. (Refer to the CMS MCP manual for details of the disk format and directory structures.)

Analysis of Standard CMS Disks

Format:



The utility operates in two modes "disk mode" and "file mode". If no file name is specified, the utility operates in "disk mode". If no disk-name is specified, the system disk is analyzed.

Disk Mode

In this mode the operator can enter a number of commands via the keyboard. These commands can be abbreviated according to the table provided at the end of this section. The format and meaning of each command in disk mode is given below.

END

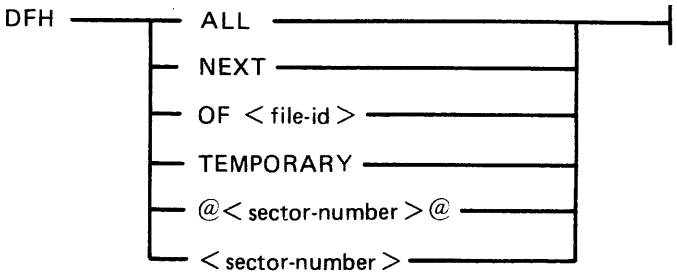
Terminates the utility.

LABEL

Reads and formats the contents of the disk cartridge label.

DFH

Reads and formats the contents of selected disk file headers. This command is followed by other details, given here:



The "TEMPORARY" option displays the headers of all temporary files.

The "ALL" option displays the headers of all files, and their contents if in use.

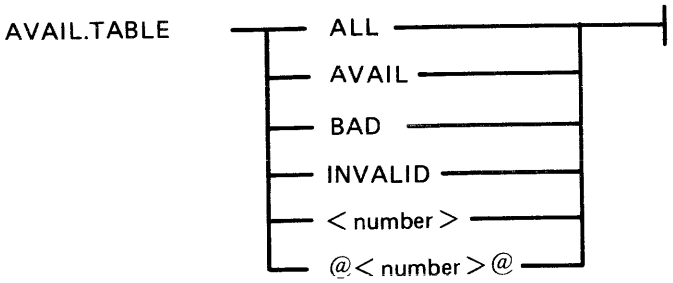
The "OF" option displays the header of the specified file or family of files: all headers will be checked and duplicates will be displayed if found.

The "sector-number" option displays any sector in disk file header format, where the number is a decimal sector address. If preceded by an @ symbol, the sector-number is in hexadecimal. This feature can be used after displaying other parts of the directory, which include sector addresses for disk file headers in hexadecimal.

The "NEXT" option displays the header of the next file in the directory.

AVAIL.TABLE

Reads and formats the contents of selected parts of the disk available space table. This command is followed by other details, given here:



The "ALL" option displays the entire available table.

The "AVAIL" option displays entries for available area only.

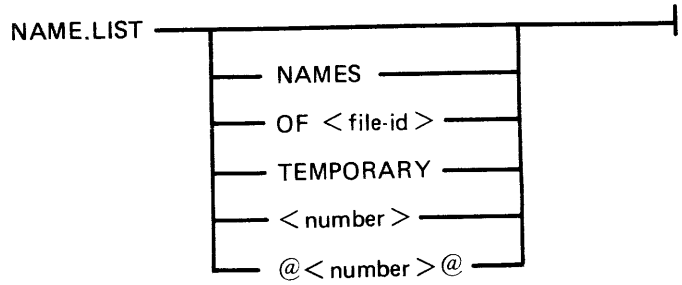
The "BAD" option displays entries for bad sectors only.

The "INVALID" option displays all entries in the available table which are invalid, in that the "length" entry does not equal the difference between "start address" and "end address".

The "sector-number" option displays any sector in available-table format, where the number is a decimal sector address in hexadecimal.

NAME.LIST

Reads and formats the directory name list, including the sector addresses of associated disk file headers. This command may be followed by other details, given here:



If no further details are given, then the entire name list is displayed.

The "NAMES" option displays entries for old (existing) files only.

The "TEMPORARY" option displays entries for temporary files only.

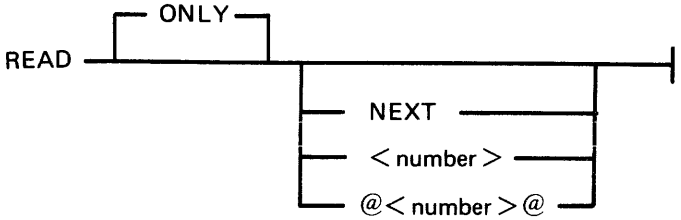
The "OF" option displays the entry for the specified file or group of files: all headers will be checked and duplicates will be displayed if found.

The "sector-number" option displays any sector in name-list format, where the number is a decimal sector address. If preceded by an @ symbol, the sector-number is in hexadecimal.

Format headings for Name.List available table or Pseudo Pack Identification Table will not be printed if no entries have been found relating to the requested option.

READ

Reads and displays the contents of any specified sector in hexadecimal and ASCII format. This command may be followed by other details, given here:



The "ONLY" option inhibits the display of information.

The "NEXT" option will read the next sector. Note that after some operations which involve a search, the "next" sector may be indeterminate. After a READ of sector n, a READ NEXT will read sector n+1. A READ command with no further details is taken as a READ NEXT.

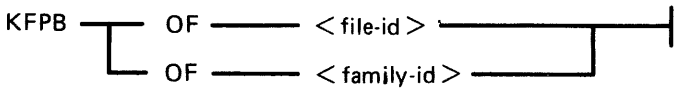
The "number" option reads the sector whose address is the number. If preceded by an – symbol, the sector-number is in hexadecimal.

DISPLAY

Displays the current sector address and contents of the program's sector-buffer. The first DISPLAY command must be preceded by a READ command. A READ ONLY followed by a DISPLAY is equivalent to a normal READ.

KFPB

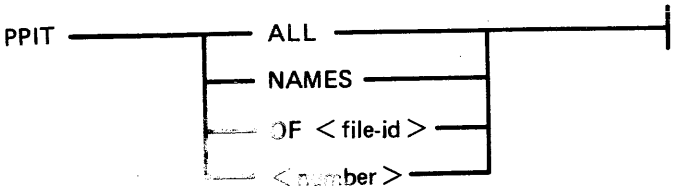
The Key File Parameter Block of a keyfile can be printed.



If the file specified is a key file, then record 1 of that file is selected and the information is printed.

PPIT

The Pseudo Pack Identification Table can be found from the Disk Cartridge Label and requested information may be printed:



The "number" option displays the sectors from the PPIT.

The "NAMES" option displays the used entries of the PPIT.

The "ALL" option displays every sector of the PPIT.

The "OF disk-name" option displays the entry for that disk.

File Mode

In file mode, the utility can be used to read selected records. Only the information commands are valid:

READ
DISPLAY
END

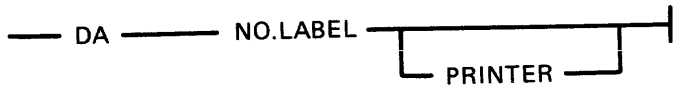
The READ command has the same format as in disk mode, except that the "number" refers to the logical record number in the file, and a READ NEXT will read the next logical record in the file. The amount of information displayed is equal to the file's record length. In DISPLAY option, if the file is a key file, then the KFPB of that key file is displayed.

Analysis of Non-CMS Standard Disks

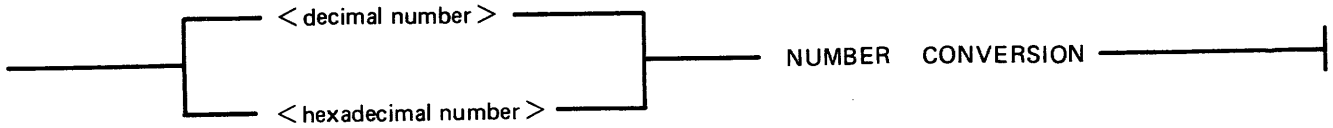
DA can analyze a non CMS standard disk (unlabelled disk). This is accomplished in the following way:

1. The drive of the disk to be analyzed is RDed.
2. "NO.LABEL" is specified in the initiating message.
3. The utility then opens an unlabelled disk and the user is prompted to AD a disk drive that has previously been RDed.

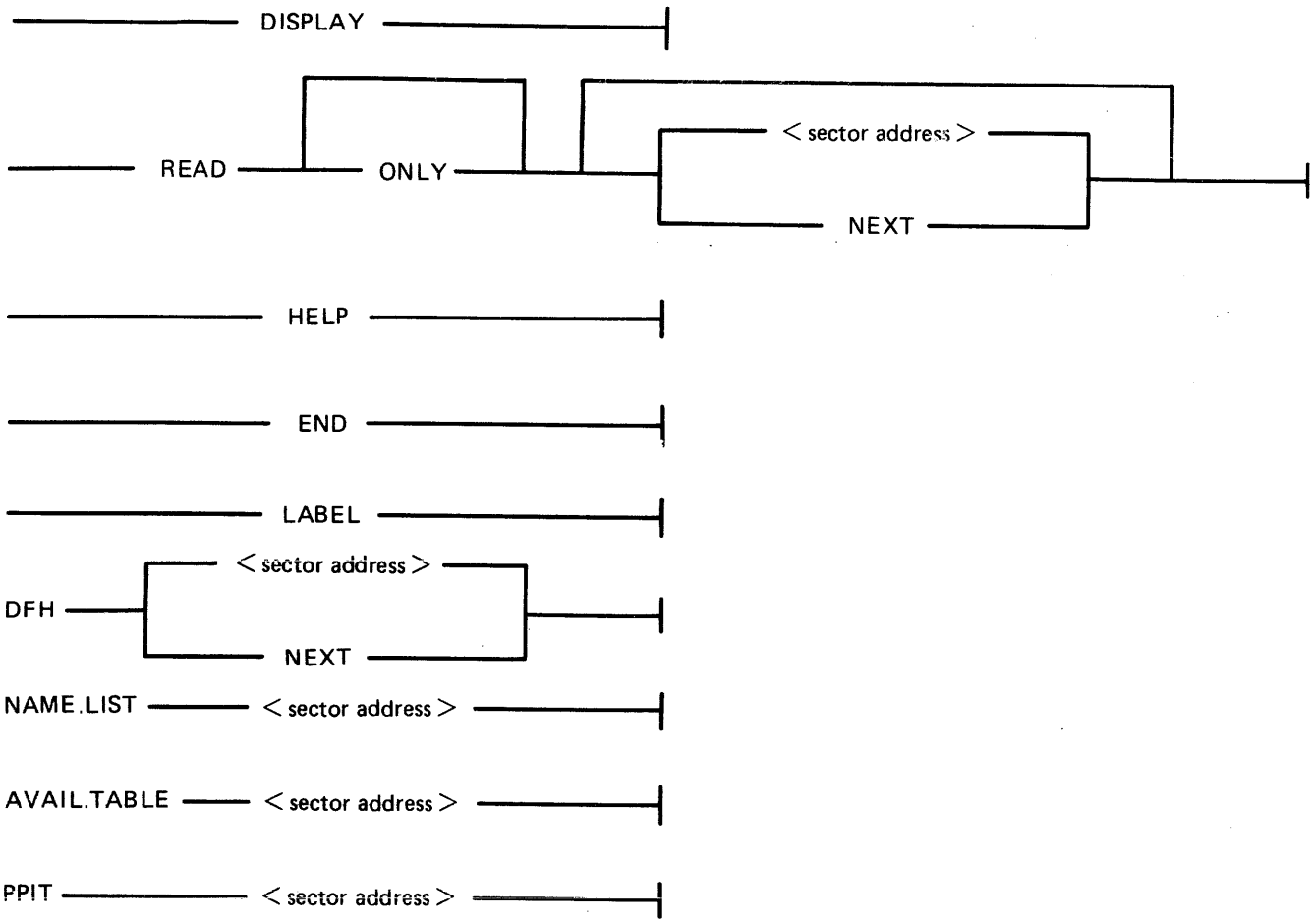
The initiating message for analyzing non CMS standard disks is:



Command options available when analyzing a non CMS standard disk:



Command options (cont.)



Directory searching options (for example DFH OF <file-name>) are not permitted in the unlabelled form of the utility, since it is possible that the disk being analyzed is either a non CMS disk or a CMS disk with corrupted label and/or corrupted directories.

In the unlabelled form of the utility, any sector of the disk may be displayed in the DFH, NAME-.LIST, AVAIL.TABLE or PPIT format.

If a command option contains a <sector address> that does not address any sector of the disk being analyzed, the following message is displayed:

OUTWITH DISK–

In the unlabelled form of the utility, if a command that is not valid in the unlabelled form is entered, the utility displays the message:

NOT VALID IN UNLABELLED MODE

General Notes

In disk mode the sector number starts from zero; that is, "READ 0" will read the first disk sector.

In file mode the record number starts from one; that is, "READ 1" will read the first logical record.

Any I/O error causes the "fetch value" to be displayed, with the current sector address if in disk mode, or current record number if in file mode.

Families

A family of files may be specified in the following command options:

KFPB ,
DFH and
NAME.LIST

A group of disks may be specified in the command option:

PPIT

For example, "KFPB OF K=", "DFH OF ABC=", "NAME.LIST OF A=" and "PPIT OF DDDDDD=" are now permitted.

All directory entries for files belonging to the specified family, or for disks belonging to the specified group of disks, are printer listed if "PRINTER" has been specified in the initiating message, otherwise they are console printed.

Pseudo Disk Tags

If the disk being analyzed has a PPIT, DA checks the Pseudo Disk Tags of the directory entries when scanning for:

a particular file or
a family of files

in the following command options:

NAME.LIST ,
DFH and
KFPB

Initially, Pseudo Disk Tag checking is set for directory entries of files residing on the physical disk specified in the initiating message. The user may alter this, so that files residing on other disks are scanned for. This is achieved by the following command option:

where <disk-name> is one of the following:

1. The name of an unrestricted Pseudo disk.
2. The name of a Pseudo disk restricted to the physical disk being analyzed.
3. The name of the physical disk being analyzed.

When the user has entered the command DISK <disk-name>, one of four messages is displayed, either on the SPO (if "PRINTER" has been specified in the initiating message) or on the console. The messages are:

1. PSEUDO DISK TAG OF <disk-name> IS xy
   The PPIT of the physical disk being analyzed contains the specified <disk-name>. xy is its hexadecimal Pseudo Disk Tag.

2. PSEUDO DISK <disk-name> NOT RESTRICTED TO PHYSICAL DISK <physical-disk-id>
   The specified <disk-name> is present in the PPIT, but it is restricted to a physical disk other than the one being analyzed.

3. <disk-name> NOT FOUND IN PPIT OF <physical-disk-id>
   The specified <disk-name> is not present in the PPIT of the physical disk being analyzed.

4. NO PPIT ON DISK <physical-disk-id>
   There is no PPIT on the physical disk being analyzed.

In each case the utility continues to wait for either the next Accept (if in PRINTER mode) or the next console command.

Abbreviations

For ease of use, input commands and other keywords may be abbreviated, as in the following table:

READ	R
ONLY	O
NEXT	N
DISPLAY	D
END	E
LABEL	L
DFH	DF
ALL	AL
OF	OF
TEMPORARY	T
AVAIL.TABLE	A
AVAIL	A
BAD	B
INVALID	I
NAME.LIST	N
NAMES	NA

Output Messages

Message	Possible Causes	Suggested Action
OUTWITH <format> AREA ADDRESS: <number> LENGTH <number>	In disk mode the sector which has been specified to be read in DFH, NAME LIST, AVAILABLE TABLE or PPIT format is outwith relevant area.	None.

(continued)

Message	Possible Causes	Suggested Action
FILE IS NOT A KEYFILE	The KFPB option has been requested and the file which was specified is not a keyfile.	None.
NOT FIXED DISK SYSTEM	The PPIT option has been requested on a non-fixed disk system and no Pseudo pack identification table exists.	None.
DA TERMINATED	The END command is successful.	None.
ILLEGAL PARAMETER input	No valid disk name or file name has been specified in the initiating message.	Re-enter.
ILLEGAL PARAMETER <character-string>	More than one option has been specified.	Re-enter.
SUCCESSFUL READ REQUIRED FIRST	A DISPLAY command has been entered before a successful read.	Enter a READ command, then a DISPLAY will work.
NOT VALID IN FILE MODE	A command other than READ or DISPLAY has been used in file mode.	Check input.
NONE FOUND or NOT FOUND	No information has been found in answer to a command.	None.
READ ERROR ON LABEL OF <disk-name>	An error was encountered while reading sector zero of the specified disk.	None.
LAST OPTION UNSUCCESSFUL	The DISPLAY option has been requested before a sector or record has been read or after an option has been unsuccessful.	Perform a READ before using display option.
ERROR ON READ-RECORD: –	The disk parity error has been encountered while reading a record in file mode.	None.
BEYOND END OF FILE-RECORD: –	A record beyond end of file has been requested to be read while in file mode.	None.
INVALID SECTOR-ADDRESS: –	A sector address beyond the last physical sector of a disk has been specified to be read while in disk mode.	None.
NOT VALID IN DISK MODE	A command option, that is not valid in disk-mode, has been entered when the utility is running in disk mode.	Check input and re-enter.

(continued)

Message	Possible Causes	Suggested Action
SYSTEM DISK LOCKED	The initiating message specified that the system disk, or a file residing on the system disk, is to be analysed, but the system disk is locked. The utility terminates after displaying this message.	None. Check that no other program is using the requested disk.
DISK <physical-disk-id> IS LOCKED	The initiating message specified that the physical disk named <physical-disk-id>, or a file residing on the physical disk named <physical-disk-id>, is to be analyzed, but the specified physical disk is locked. The utility terminates after displaying this message.	None. Check that no other program is using the requested disk.
NOT VALID IN UNLABELLED MODE	Utility is in unlabelled mode and a command was given which is not valid for this mode.	Check input and re-enter.

DSKUTL

NOTE
On B 1000 systems, only RF and LIST are implemented.

The DSKUTL utility provides the following functions:

IN	Initializes a disk.
RF	Reformats a previously initialized disk according to parameters specified in the initiating message.
LIST	This function provides the facility to list any range of disk sectors (absolute addressing) which are accessible to the system software.
COPY	The COPY function provides the facility to duplicate disks of the same type. Note that only BSMII disks are supported by this function. The entire disk contents are duplicated including the CMS-reserved areas of the disk. Non-CMS format disks can also be duplicated.
BOOT	Replaces the bootstrap code in track zero of a CMS initialized disk without changing any other information the disk contains.

NOTE
This utility has the facility to handle pseudo-disks. This feature is only implemented on B 900 systems. In addition, DSKUTL can only be used by systems which implement the "Open Disk Unlabelled" function in the MCP.

These make the following Stand Alone Utility (SAU) functions available under MCP control:

IN, FE
RF (for any SDI disk type and pre-initialized Caelus disks)
PDX (for any initialized disk)
DISCOPY (for BSMII disks)

The disk types which are supported by the DSKUTL RF, IN, BOOT and LIST functions are detailed in table 4-2.

NOTE
On SDI type disks (BSMII and 211), an execution of the IN function is equivalent to the Reformat Function, RF, with the IGNORE option specified.

In addition to the disk types described in table 4-2, DSKUTL will also support other types of disk, including the fixed and floppy disks for the B 95/CP 9350. These disks are not referred to in table 4-2. The utility will calculate the allocation unit of the disk from its capacity. The physical attributes of B 95 fixed and floppy disks are not relevant.

The Mnemonics and device kinds of B 95 fixed and floppy disks are as follows:

	Mnemonic	**Device Kind**
Floppy	DM	C8
Fixed	DF	C9

Table 4-2. DSKUTL – Supported Disk Types

DISK/PACK	CONTROLLER	MNEMONIC	DEVICE	CYL.	TRACKS/ CYL.	SECTS/ TRK	ALLOC. UNIT	NO. OF SECTORS
1MB MINI	c	DM	C7	88	2	32	1	@001600@
CARTRIDGE	(100TPI) c	DK	CB	203	2	32	1	@0032C0@
CARTRIDGE	(200TPI) c	DK	CB	406	2	32	1	@006580@
201-I FIXED	(single) c	DF	CC	406	2	64	1	@00CB00@
201-I FIXED	(dual) c	DF **	CC	406	4	64	2	@019600@
211 FIXED	(single) s	DF	CD	335	4	80	2	@01A2C0@
211 FIXED	(dual) s	DF	CD	335	8	80	4	@034580@
211 FIXED	(quad) s	DF	CD	335	16	80	8	@068B00@
BSMII MINI	(3MB) s	DM	CE	142	2	59	1	@004120@
BSMII MINI	(4.7MB) s	DM	CE	221	2	59	1	@0065C0@
M 205 PACK	DPC	DP	CF	407	5	90	4	@02C360@
M 206 PACK	DPC	DP	CF	814	5	90	8	@0586E0@
M 207 PACK	DPC	DF/DP	CA	1563	8	90	32	@110D60@
M 223 PACK	DPC	DP	CF	203	20	60	4	@038380@
M 225 PACK	DPC	DP	CF	406	20	60	8	@076720@

c = caelus, s = SDI (Standard Disk Interface) ** = B 900 only
B 1000 systems only support cartridges and pack.

Bootstrap Table For IN (B 90 and B 900 Only)

The CMS Bootstrap is always selected for 211 or 201-I Fixed Disks. MTR sectors are always reserved for the 201-I Fixed Disk whether <MTR> is specified in the initiating message or not. For all other types of disks, if <MTR> is not specified, the CMS Bootstrap is selected and sectors are not reserved.

Table 4-3. DSKUTL IN Bootstrap Table

System	B 90		B 900	
DEFAULT FILE-NAME records 2 – 31 records 32 – 61 records 62 – 91 records 92 – 121	0000000/CMSBOOT CMS BOOTSTRAP MTR/CAELUS BOOTSTRAP MTR/BSMII BOOTSTRAP COLDSTART BOOTSTRAP		0000000/SYSINITBOOT BDS BOOTSTRAP MTR/CAELUS BOOTSTRAP MTR/BSMII BOOTSTRAP Not applicable.	
MTR INITIALIZE	**BOOTSTRAP SELECTED**	**AREAS RESERVED**	**BOOTSTRAP SELECTED**	**AREAS RESERVED**
1 MB MINI	MTR/ CAELUS	0A80-0ABF 0B00-0B3F 1540-157F	MTR/ CAELUS	0A80-0ABF 0B00-0B3F 1540-157F
CARTRIDGE (100TPI)	MTR/ CAELUS	1540-157F 2A80-2ABF 2B00-2B3F	MTR/ CAELUS	1540-157F 2A80-2ABF 2B00-2B3F
CARTRIDGE (200TPI)	MTR/	2A80-2ABF	MTR/	2A80-2ABF

Table 4-3. DSKUTL IN Bootstrap Table
(continued)

MTR INITIALIZE	BOOTSTRAP SELECTED	AREAS RESERVED	BOOTSTRAP SELECTED	AREAS RESERVED
	CAELUS	2B00-2B3F 5540-557F	CAELUS	2B00-2B3F 5540-557F
201-I FIXED (S)	CMS	0080-017F 5200-527F CA80-CAFF	BDS	0080-017F 5200-527F CA80-CAFF
201-I FIXED (D)	Not supported.		BDS	0080-017F 5200-527F CA80-CAFF
211 FIXED (ALL Types)	CMS	NONE	BDS	NONE
BSMII MINI 3 MB	MTR/ BSMII	1321-140C 4012-40FD	MTR/ BSMII	1321-140C 4012-40FD
BSMII MINI 4.7 MB	MTR/ BSMII	1321-140C 64B7-65A2	MTR/ BSMII	1321-140C 64B7-65A2
M 205/206/207/223	CMS	NONE	BDS	NONE
M 225 Disk Packs	Not supported	NONE	BDS	NONE

Area addresses are in allocation units (hexadecimal).

NOTE

On the 211 Fixed Disks there are MTR sectors outwith the virtual addressable area of the disk.

DSKUTL operates under MCP control and provides some of the functions of the Stand Alone Utilities (SAU).

In order to use this utility, the system must be running under an MCP that supports the OPEN/CLOSE disk unlabelled function and a SYSLANGUAGE file must be present. These functions are provided by the B 90 MCP version 3.04 and above.

Before executing DSKUTL, the operator must reserve the disk drive or unit on which the function is being performed by means of the RD intrinsic. When the utility begins execution, it opens an unlabelled disk and the operator is requested to assign the required disk drive or unit mnemonic, which has been previously reserved, by using the AD command.

The disk to be reserved for access by DSKUTL must not have any files open, thus the disk currently in use as the system disk may not be accessed by this utility.

NOTE

In order to use this utility, the disks in use must be in a format recognizable by the MCP as follows:

SDI disks are factory initialized, although not necessarily in CMS format, and are supported by this utility whether or not they have been previously initialized or formatted on a CMS system.

DSKUTL – <COLDSTART> Option B 90 Only

To allow initialization or reformatting of mini disks (1 MB mini, BSMII mini and B 95/CP 9350 mini disks) for Coldstart purposes on a B 90 system, the option <COLDSTART> (including angle brackets) may be specified in the initiating message of IN (initialize), RF (reformat) and BOOT (bootstrap replacement) functions as an alternative to the <MTR> option.

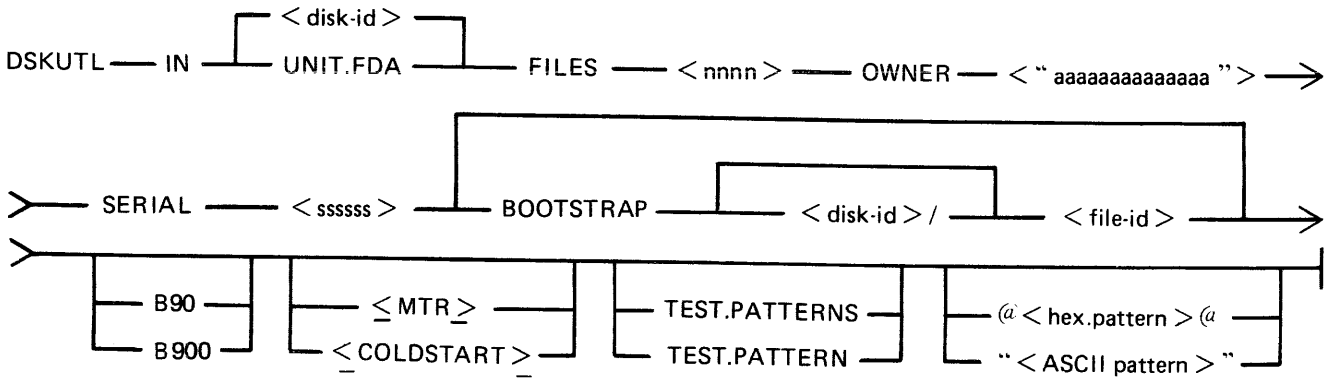
NOTE

This option can only be used on a B 90 system. The bootstrap file used must be CMSBOOT version 3.04 or later.

Format 1: IN (B 90 and B 900 Only)

To initialize a disk using DSKUTL the following steps are required.

The following format provides the initializing (IN) function:



The star-file option is implemented in this utility.

Parameters may also be entered via Accepts – see subsection DSKUTL Reformat and Initialize Parameters – via Accepts.

To initialize a disk for MTR purposes, <MTR> should be specified.

Disk-name — This field specifies the name written to the disk-label of the disk being reformatted.

UNIT.FDA — By specifying UNIT.FDA instead of <disk-name> in the initiating message, DSKUTL initializes a 211 or 201-I fixed disk as a physical unit with a PPIT, provided the system is running as a fixed disk system. The system PPIT is scanned for the first disk-name listed as a physical unit which is not on-line. The disk being initialized is assigned this disk-name and associated logical unit-number when the utility writes the disk-label.
If all physical units listed in the system PPIT are found to be on-line, the disk is assigned a default disk-name of "0000xy" where "xy" is the logical unit-number (in decimal characters) of the first available unit slot in the system PPIT.

FILES nnnn	This specifies the maximum number of files to be held on the disk. <nnnn> is an integer number in the range 1 to 2804.
OWNER "aaaaaaaaaaaaa"	This field is used to specify the owner field in the disk-label. It must consist of an alphanumeric entry of 1 to 14 characters delimited by quote marks (“).
SERIAL ssssss	The serial number field of the disk-label is specified by this field. It must be a six-digit integer number and must include leading zeros.
BOOTSTRAP file-name	The bootstrap file-name, if mentioned, specifies the name of the bootstrap file to be copied to the disk being initialized. If the BOOTSTRAP option is not entered, the utility searches for the default file-name as shown in table 4-3.
TEST.PATTERN or TEST. PATTERNS pattern list	This option is only applicable to Caelus type disks. The option allows the user to specify up to four test patterns to be written to and read from the disk being initialized. The test patterns can be up to 20 characters long if ASCII, or 40 characters long if hexadecimal. Hexadecimal test patterns are delimited by AT symbols (@) and ASCII test patterns are delimited by quotes (“).
<MTR>	The MTR option in the initiating message causes sectors to be reserved in the available table for MTR purposes. The number and location of these sectors depends on the system and disk type. This option is equivalent to the SAU FE function.
<COLDSTART>	This option causes the COLDSTART bootstrap to be written into track zero. This is used to invoke the COLDSTART utility (see COLDSTART utility). This option can only be used for BSMII and 1 MB mini disks on a B 90 system.
Machine spec B 90 or B 900	This option allows the user to specify the bootstrap type applicable to either a B 90 or a B 900. If the bootstrap file is not specified in the initiating message, the utility will expect CMSBOOT to be on the system disk if B 90 is specified, or SYSINTBOOT if B 900 is specified.

NOTE

The angled brackets round <MTR> and <COLDSTART> must be included in the initiating message.

Operation

To initialize a disk, the user inputs an initiating message of the format shown previously. The utility then prompts the user with the following message:

```
PLEASE RESERVE (RD) DRIVE, THEN INSERT DISK FOR "IN" AND THEN ASSIGN (AD)
DRIVE
```

followed by the MCP prompt:

```
<mix number>/DSKUTL PROGRAM SUSPENSION
ASSISTANCE REQUIRED FOR DISK FILE INITIALIZE
USE "AD" TO ASSIGN DEVICE
```

At this point, the user is required to ready (RY) and reserve (RD) the required drive.

NOTE

If the disk to be initialized is virgin, it is necessary to reserve (RD) the drive before inserting the disk.

If Test Patterns have been specified and the disk AD'ed is either a BSMII or 211, the utility will ignore them.

If the disk label is already in CMS format, the utility displays the following prompt:

```
DISK IN DRIVE Dxy IS LABELLED <diskname>
PLEASE ENTER OK IF IN IS TO CONTINUE
```

followed by the MCP prompt:

```
<mix number>/DSKUTL REQUIRED DATA. USE "AX"
```

The disk-name is from the label and the operator is required to AX <mix number> OK for the utility to continue.

All previous information on the disk is ignored.

NOTE

DSKUTL tries to read the LABEL of the AD'ed disk. If the disk is virgin, an error is detected and displayed by the MCP. DSKUTL continues and initializes the disk. DSKUTL displays the following message "CANNOT READ LABEL OF DISK IN DRIVE Dxy, WILL ASSUME DISK IS BEING INITIALIZED FOR THE FIRST TIME

Test Patterns for Caelus Type Disks

For Caelus type disks only (1 MB mini, cartridge, 201 I fixed), the following initialize routine is invoked. Sector addresses are written to the disk using the INIT.TRACK communicate. The utility terminates if an error condition is returned from this communicate. When writing of sector addresses has been completed, the utility writes/reads test patterns to every sector of the disk. If no test patterns are specified by the user, three passes are performed. If one or more patterns are specified, four passes are performed.

The three default test patterns are 180 bytes of Hex 00, 180 bytes of Hex FF and 180 bytes of Hex 63.

The sequence of test patterns used is as follows:

No. of Patterns Specified	Pass 1	Pass 2	Pass 3	Pass 4
0	Hex 00	Hex FF	Hex 63	–
1	Hex 00	Hex FF	Hex 63	P1
2	Hex FF	Hex 63	P1	P2
3	Hex 63	P1	P2	P3
4	P1	P2	P3	P4

where P1, P2, P3 and P4 are specified patterns.

Any specified patterns are repeated to fill 180 bytes.

If a disk is being initialized for ACSYS (SL5) use, a test pattern of "@FF@" should be specified for the last pass so that 180 bytes of "@FF@" remain on all available disk sectors on completion of the initialize function.

The sectors are read back to verify that patterns were written correctly. If a read or write error is encountered on a sector, or test pattern verification fails, the sector is marked as bad in the available table. The bad sector limit is 32 for a 1 MB mini disk and 50 for a cartridge or 201-I. If this limit is exceeded or if any MTR sectors to be reserved or sectors of track 0 are found to be bad, the utility terminates. This initialization routine is not performed on the factory-initialized BSM II or 211 disks.

If <MTR> is specified in the initiating message, depending on the system and disk type, MTR sectors may be read/write checked and reserved in the available table when constructed by the utility.

The appropriate bootstrap code is selected from the specified or default bootstrap file.

The new available table and directories with single SYSMEM entry and the appropriate bootstrap code are written to the disk by the utility. If MTR sectors are to be reserved, they are marked as bad in the available table.

Special test patterns are written to the MTR tracks of the BSMII disk.

If UNIT.FDA was specified, the utility acknowledges with the following message, giving details of the disk-name being assigned to the fixed disk.

DFy WILL BE INITIALIZED AS DISK IN DRIVE <disk-name>

A 13 sector PPIT is written to the disk with the assigned disk-id, logical unit number and pseudo pack tag entered in the relevant physical unit slot. All other entries are made available. When this unit is readied by the user after the utility has successfully terminated, the PPIT is modified to match the System PPIT by the MCP AVR routine.

Finally, the utility writes the label to sector zero.

When initialization has been completed successfully, the utility displays the following message and terminates:

DISK IN DRIVE <drive-id> INITIALISED AS <disk-id>
– BOOTSTRAP IS <type> VERSION <version-number>
– NUMBER OF BAD SECTORS = <No of sectors marked bad>

where – <type> = CMS/WARMSTART if the CMS bootstrap was used,
or
MTR if either MTR-CAELUS or MTR-BSMII was used,
or
COLDSTART if the COLDSTART bootstrap was used.
<version number> = The version number contained in the bootstrap code, if one exists, otherwise
the version number of the bootstrap file.
<disk-id> = The name of the disk that the bootstrap has been copied to.

Pseudo Pack Identification Table Construction

Slots which are reserved for physical units are marked unable to be used by filling with Hex. 82.

Slots which are reserved for Pseudo Packs are marked unused by filling with Hex. 80.

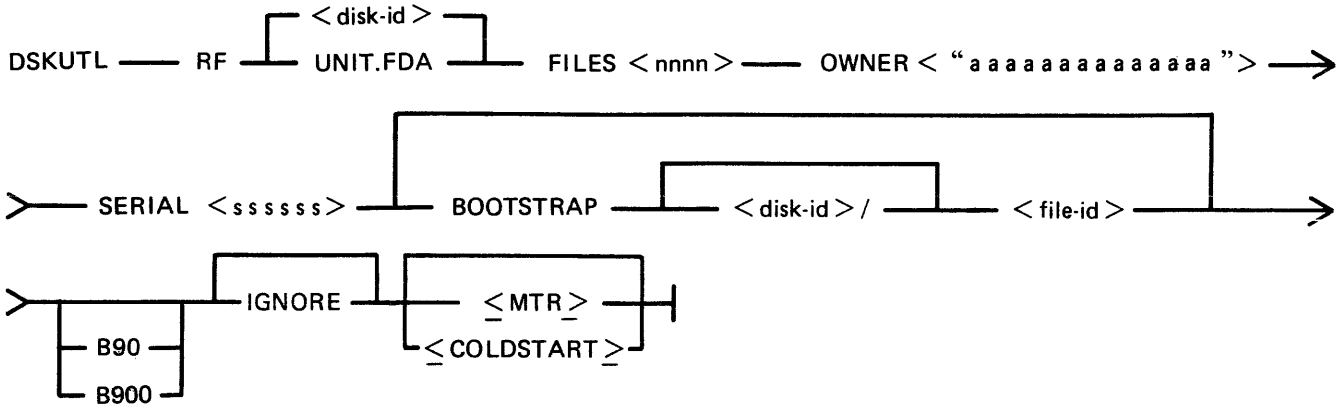
On a 211 type Fixed Disk, 222 slots are available for Pseudo Packs, Hex. 21 – Hex. FE.

On a 201-I type Fixed Disk, 211 slots are available for Pseudo Packs, Hex. 21 – Hex. F3, the last (20th) slot in each PPIT sector is marked as unable to be used.

Format 2: RF

NOTE

The angle brackets must be included in the initiating message for MTR tracks to be reserved. Similarly, the angle brackets must be included around COLD-START.



IGNORE

If the IGNORE option is specified, the utility will ignore any information already contained in the CMS reserved areas of the disk. This option must always be specified if the disk is not already in CMS format.
If IGNORE is not specified, the utility will expect the disk-label and available table to be in CMS format. When re-writing the available table to the disk, the utility will accommodate up to 90 bad entries from the original available table. If more than 90 bad entries are encountered on reading the available table of the disk, the utility will be terminated.

NOTE: The IGNORE option can be used to reformat a disk which has a corrupted label, directories or available table. However, care must be exercised if using this option to reformat Caelus disks (see table 4-2). SDI disks, due to the fact that bad sectors are "hidden" by relocation, should never contain bad entries in the available table other than those reserved for MTR purposes. Caelus disks, however, often contain a number of unusable sectors "hidden" by means of the entries in the available table. Use of the IGNORE option will ignore these entries when constructing the new available table. Subsequent access of these previously noted bad sectors by the MCP can give rise to I/O errors.

Disk-name

This field specifies the name written to the disk-label of the disk being reformatted.

UNIT.FDA (B 900 only)

By specifying UNIT.FDA instead of <disk-name> in the initiating message, DSKUTL initializes a 211 or 201-I fixed disk as a physical unit with a PPIT, provided the system is running as a fixed disk system.

	The system PPIT is scanned for the first disk-name listed as a physical unit which is not on-line. The disk being initialized is assigned this disk-name and associated logical unit-number when the utility writes the disk-label. If all physical units listed in the system PPIT are found to be on-line, the disk is assigned a default disk-name of "0000xy" where "xy" is the logical unit-number (in decimal characters) of the first available unit slot in the system PPIT.
FILES nnnn	This specifies the maximum number of files to be held on the disk. <nnnn> is an integer number in the range 1 to 2804.
OWNER "aaaaaaaaaaaaaa"	This field is used to specify the owner field in the disk-label. It must consist of an alphanumeric entry of 1 to 14 characters delimited by quote marks (").
SERIAL ssssss	The serial number field of the disk-label is specified by this field. It must be a six-digit integer number and must include leading zeros.
BOOTSTRAP file-name (B 90/B 900 only)	The bootstrap file-name, if mentioned, specifies the name of the bootstrap file to be copied to the disk being initialized. If the BOOTSTRAP option is not entered, the utility searches for the default file-name as shown in table 4-3.
<MTR>	The MTR option in the initiating message causes sectors to be reserved in the available table for MTR purposes. The number and location of these sectors depends on the system and disk type. This option is equivalent to the SAU FE function.
<COLDSTART>	This option causes the COLDSTART bootstrap to be written into track zero. This is used to invoke the COLDSTART utility (see COLDSTART utility). This option can only be used for BSMII and 1 MB mini disks on a B 90 system.

A <machine-spec> may now be specified for the IN (Initialize), RF (Reformat) and BOOT (Bootstrap replacement) functions.

<machine-spec> can be either B 90 or B 900.

This allows the user to specify the type of Bootstrap to be used for the IN, RF and BOOT functions. This makes it possible, for example, to create a disk on a B 90 with a B 900 Bootstrap.

If no Bootstrap file-name is specified in the initiating message, DSKUTL will use "CMSBOOT" (if B 90 was specified) or "SYSINITBOOT" (if B 900 was specified) on the systems disk.

No prompt will be issued for <machine-spec> if parameters are entered via ACCEPT/DISPLAYs.

Operation

On B 90 or B 900 the utility immediately opens the specified or default bootstrap file and verifies that the appropriate checksum is correct. The operator is requested to reserve and assign the required disk drive or unit.

If the label of the disk is already in CMS format, the utility displays the name read from the disk-label and requests the operator to confirm that reformatting is required.

The operator must enter:

AX <mix number> OK

in order for the utility to continue.

The new or existing available table and directory entries and a single SYSMEM entry are written to the output disk and on B 90 or B 900 the bootstrap file is copied to the disk.

On B 900 only, if UNIT.FDA was entered in the initiating message, a 13-sector PPIT is written to the disk with the assigned disk-name, and the logical unit-number and pseudo-pack tag are entered in the physical unit slot. All other entries are made available. When the unit is subsequently made ready by the operator on termination of the utility, the PPIT will be modified to match the system PPIT by the AVR routine in the MCP.

Finally, the utility writes the disk-label to sector zero (0).

In order to access the disk which has been reformatted, the user must make the disk ready by use of the RY command when the utility has gone to successful EOJ.

Example

To reformat a BSMII (3/6 MB) disk ("TESTA"), already in CMS format with the following parameters:

Disk-name – DISKA
Number of files – 1000
Owner – USER-FRED
Serial Number – 123456
Bootstrap file is default

Insert disk called "TESTA" in drive DMA.
System displays "DMA TESTA/ 0 FILES OPEN"

Enter "DSKUTL RF DISKA FILES 1000 OWNER "USER-FRED" SERIAL 123456
system displays:
PLEASE RESERVE (RD) DRIVE, THEN INSERT DISK FOR RF AND THEN ASSIGN (AD)
<mix-no>/DSKUTL PROGRAM SUSPENSION<28>
ASSISTANCE REQUIRED FOR DISK FILE REFORMAT
USE "AD" TO ASSIGN DEVICE

Enter: RD DMA
system displays:
DMA RESERVED 0 FILES OPEN

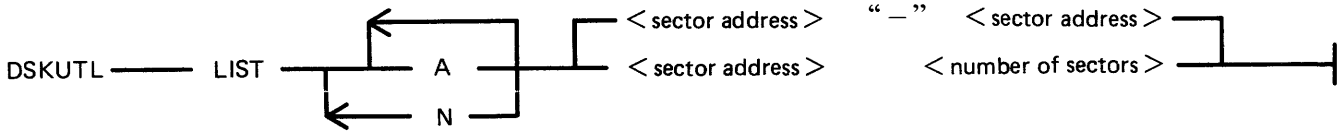
Enter: AD <mix-number> DMA
system displays:
DISK IN DRIVE DMA IS LABELLED TESTA,
PLEASE ENTER OK IF RF IS TO CONTINUE
<mix-number>/DSKUTL REQUIRES DATA. USE "AX".

Enter: AX <mix-number> OK
system displays:
DISK IN DRIVE DMA REFORMATTED AS DISKA

– BOOTSTRAP IS CMS/WARMSTART VERSION XX.XX.XX
– NUMBER OF BAD SECTORS = 0

Enter: RY DMA
system displays:
DMA DISKA 0 FILES OPEN

Format 3: LIST



A or N

Alpha character listing or numeric (hex) listing can be specified by the use of this option in the initiating message. The default is both alpha and numeric in the same format as for the CMS LIST utility.

sector address – sector address

This field specifies the sectors to be listed and consists of two decimal numbers separated by a hyphen. The sectors listed will include the starting and ending sector number.
Example: <1-20> causes sector numbers 1 through 20 to be listed.

sector address number of sectors

This option also specifies the number of sectors to be listed and can be used as an alternative to the above format. The number of sectors specified by the second decimal number will be listed, starting at the sector address specified by the first decimal number.
If sector zero (0) is specified for listing, then the following is written to the printer following the content of sector 0:

Sector Address	Length	Directory
32/@0020@	65/@0041@	AVAILABLE TABLE
97/@0061@	91/@0058@	NAME LIST
188/@00BC@	1001/@03E9@	DISK FILE HEADERS
0/@0000@	0/@0000@	PSEUDO PACK IDENTIFICATION TABLE

The above sample output is from the disk reformatted with the parameters given in the example DSKUTL RF function above. If the disk-label is not in CMS format, this portion of the listing is omitted.

Operation

The utility causes a disk to be opened unlabelled, and prompts the operator to reserve and assign the appropriate disk drive or unit. On assigning the device, the utility lists the specified sectors and goes to EOJ. The drive must now be made ready by the operator before normal access can be made to the disk.

Example:

To list the first 10 sectors of the disk reformatted in the sample DSKUTL RF function above:

Insert disk "DISKA" in drive DMA.
The system displays "DMA DISKA/ 0 FILES OPEN"
Enter "DSKUTL LIST 0 10"

or

"DSKUTL LIST 0 – 9"

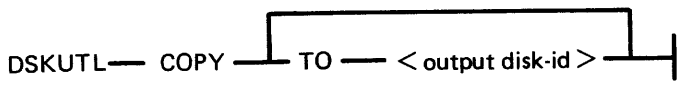
The system displays:

PLEASE RESERVE (RD) DRIVE, THEN INSERT DISK FOR LIST AND THEN ASSIGN (AD) DRIVE

Enter: RD DMA
The system displays "DMA RESERVED 0 FILES OPEN"
Enter "AD <mix-number> DMA"

The utility now lists the specified sector numbers and goes to EOJ.

Format 4: COPY (B 90/B 900 Only)



TO disk-name — For the COPY function, this option in the initiating message causes the disk-name specified to be written to the disk-label of the output disk if the input disk is in CMS format.
If the input disk is not in CMS format or is a fixed disk with a pseudo-pack PPIT, this name is ignored when writing the disk-label.
If no disk-name is specified in the initiating message, the disk-label is copied from the input disk to the output disk.

Operation

On commencing execution, the utility displays the message:

"PLEASE RESERVE (RD) DISK, THEN INSERT INPUT DISK FOR COPY AND THEN ASSIGN (AD) DRIVE"

The operator must now reserve and assign the disk drive required for input. The utility displays the message:

"PLEASE RESERVE (RD) DISK, THEN INSERT OUTPUT DISK FOR COPY AND THEN ASSIGN (AD) DRIVE"

The operator must now reserve and assign the disk drive or unit required for output.

The utility checks that both input and output disks are SDI disks of the same type and capacity. Having established this, the utility proceeds to copy each sector of the input disk to the output disk.

If an error which cannot be corrected is detected, the operator is informed of the sector(s) in error and the utility continues until the entire disk is copied.

From Release 3.04, only BSMII type disks may be copied using this function. The following messages are displayed by the utility if a 211 fixed disk (or any other type of disk other than BSMII) is AD'ed to the utility in the Copy function. The utility then goes to End of Job.

COPY FUNCTION DOES NOT SUPPORT THIS TYPE OF DISK

Input Disk Error Handling

NOTE

System delays may be experienced prior to the display of messages reporting input disk read errors.

If more than 100 input disk sectors with read errors are detected, the utility displays the following messages and then goes to End of Job.

MORE THAN 100 HARDWARE ERRORS DETECTED ON DISK IN DRIVE <drive id>

The utility reports the sector addresses of input disk errors as they are encountered. For CMS disks only, the utility also attempts to resolve any corrupt information which may have been copied to the output disk as a result of input errors. The utility assumes that the disk label and the sectors of the Disk File Header and Name List areas which were read successfully from the input disk contain valid information that is not corrupt.

Example:

To copy the disk "DISKA" to a second disk "DISKB"

Insert "DISKA" into drive DMA and the output disk into drive DMB
Enter: DSKUTL COPY TO DISKB
System displays:
"PLEASE RESERVE (RD) DRIVE, THEN INSERT INPUT DISK FOR COPY AND THEN ASSIGN (AD) THE DRIVE"
"<mix-number>/DSKUTL PROGRAM SUSPENSION <28>"
"ASSISTANCE REQUIRED FOR DISK FILE LIST, USE 'AD' TO ASSIGN DEVICE"

Enter: RD DMA
system displays:
"DMA RESERVED 0 FILES OPEN"

Enter: "AD <mix-number> DMA"
system displays:
"PLEASE RESERVE (RD) DRIVE, THEN INSERT OUTPUT DISK FOR COPY AND THEN ASSIGN (AD) DRIVE"
"<mix-number>/DSKUTL PROGRAM SUSPENSION <28>"
"ASSISTANCE REQUIRED FOR DISK FILE LIST, USE 'AD' TO ASSIGN DEVICE"

Enter: RD DMB
system displays:
"DMB RESERVED 0 FILES OPEN"

Enter: "AD <mix-number> DMB"

The utility now copies the disk and goes to end of job.

Recommendations For Read/Write Errors

If a read or write error is detected during the COPY function on either the input or output disk, the following procedure is recommended prior to using the output disk:

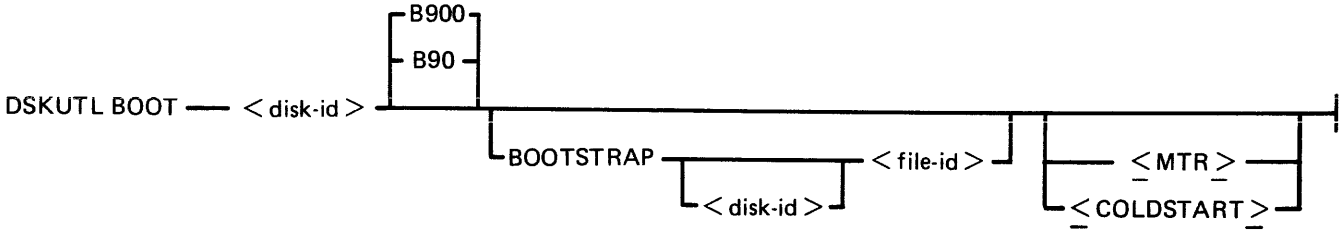
1. Produce a KA listing of the output disk.
2. From the sector addresses displayed by the utility for the read/write errors, determine which files contain the errors. These files should be removed from the output disk.

If the read/write error occurs in the disk directory area, the disk should not be used without re-initialization (or re-formatting if it is an SDI disk).

Format 5: BOOT (B 90/B 900 Only)

The BOOT function provides the facility to replace the bootstrap code in Track Zero of a CMS initialized disk without changing any other information contained by the disk. This allows the upgrading of disks with the latest level of bootstrap, without reformatting or initializing the disks with the subsequent loss of any files present on the disks.

The syntax of BOOT is:



<disk-name>	This is the name of the disk which is to have its bootstrap replaced.
BOOTSTRAP <file-name>	The bootstrap file name, if specified, is used by the utility instead of the default bootstrap file-name of 0000000/CMSBOOT on a B 90 system or 0000000/SYSINTBOOT on a B 900.
<MTR>	This option is only valid for disks which can already be used for MTR purposes. With <MTR> specified, the relevant bootstrap (as decided by the utility) is selected from the default or specified bootstrap file by the utility and copied to the specified disk. See BOOT Bootstrap Table.
<COLDSTART>	This option causes the COLDSTART bootstrap to be written into track zero. This is used to invoke the COLDSTART utility (see COLDSTART utility).

NOTE

The angled brackets around <MTR> and <COLDSTART> must be included in the initiating message.

When the appropriate bootstrap has been successfully copied to the specified disk, the utility displays the following message and terminates:

```
BOOTSTRAP ON DISK <disk-id> REPLACED
BY <bootstrap-used> VERSION <version-number>
```

where <Bootstrap-used> = CMS/WARMSTART if the CMS B 90 bootstrap was used,
or
B 900 system if a B 900 bootstrap was used,
or

MTR if either MTR-CAELUS or MTR-BSMII was used,
or
COLDSTART if the COLDSTART bootstrap was used.

<version-number> = The version number contained in the bootstrap code if one exists; otherwise the version number of the bootstrap file.
<disk-name> = The name of the disk that the bootstrap has been copied to.

NOTE

If an MTR bootstrap has been replaced by either CMS or BDS bootstrap, it is impossible to replace this new bootstrap by an MTR bootstrap. To replace the CMS or BDS bootstrap by an MTR bootstrap, the disk must be initialized or reformatted with the MTR option.

NOTE

If MTR bootstrap on 5.25 inch disk is replaced by a non-MTR bootstrap, the MTR bytes in the disk label are not reset. This does not affect subsequent use of the disk as a CMS disk.

Table 4-4. DSKUTL BOOT Bootstrap Table If MTR Specified

Disk Type	Bootstrap B 90	Selected B 900
	Default bootstrap file is 0000000/CMSBOOT	Default bootstrap file is 0000000/SYSINTBOOT
1 MB Mini	MTR – CAELUS	MTR – CAELUS
Cartridge (100 TPI and 200 TPI)	MTR – CAELUS	MTR – CAELUS
201-I Fixed (single density)	CMS	BDS
201-I Fixed (dual density)	Not supported	BDS
211 Fixed (all types)	CMS	BDS
BSMII Mini	MTR – SDI	MTR – SDI

In the case of the BSMII disk, the special test patterns required for MTR are re-written to the reserved disk areas. For Caelus disk, reserved disk areas are checked for readability before the bootstrap is copied.

Table 4-5. DSKUTL BOOT Bootstrap Table If COLDSTART Specified

Disk Type	Bootstrap B 90	Selected B 900
	Default bootstrap file is 0000000/ file is 0000000/ CMSBOOT	Default bootstrap file is 0000000/ SYSINTBOOT SYSINTBOOT
BSMII Mini and 1MB Mini	COLDSTART	Not Supported.

DSKUTL Reformat and Initialize Parameters – via Accepts

Parameters for initialization or reformatting of a disk may be entered by Accepts by specifying:

DSKUTL IN or DSKUTL RF

in the initiating message with no other parameters.

The following prompts for each parameter in both Initialize and Reformat functions are displayed by the utility, prefixed with:

DSKUTL <IN> or DSKUTL <RF>

Parameters entered by AX are checked for legality and the utility re-prompts the user until a legal response has been given for the particular parameter requested.

Prompt	Description
ENTER THE NAME FOR THE DISK (UP TO 7 LEGAL CHARACTERS)	The user is requested to enter the disk name using the "AX" command.
OR "UNIT.FDA" IF DISK FOR FIXED DISK SYSTEM	If the utility is running on a fixed disk on B 900, "UNIT.FDA" may be entered instead of the name.
ENTER THE MAXIMUM NUMBER OF FILES THAT THE DISK SHOULD CONTAIN (A NUMBER LESS THAN 2805)	The user is requested to enter the number of files that are going to be on the disk using the "AX" command.
ENTER THE SERIAL NUMBER FOR THE DISK (6 DIGITS)	The user is requested to enter a number of 6 digits. It is advised to enter the serial number as it is on the medium. Use the "AX" command to enter the number.
ENTER THE NAME OF THE DISK OWNER (UP TO 14 CHARACTERS DELIMITED BY "")	The user is requested to enter his name using the "AX" command.
IS DISK FOR COLDSTART USE? ENTER Y OR N	The user is requested to indicate whether or not this disk is going to be a coldstart disk. Enter Y or N using the "AX" command. Not applicable to B 1000.
PLEASE RESERVE (RD) DRIVE, THEN INSERT DISK FOR IN OR RF AND THEN ASSIGN (AD) DEVICE	The user is requested to reserve the drive using the RD command. For example, RD DMA, then assign the drive using the AD command. For example, AD <mix-no of DSKUTL> DMA.

DSKUTL Output Messages

NOTE

Error messages relating to options IN, COPY, BOOT or UNIT.FDA are not relevant to B 1000 systems. If such a message is displayed on B 1000, check input.

Message	Possible Causes	Suggested Action
-HARDWARE ERROR DETECTED WHILE READING FILE <file-name>	The utility has encountered an error from which it cannot recover.	Check integrity of the specified disk.
-HARDWARE ERROR DETECTED WHILE READING LABEL ON DISK <disk name>	The utility has encountered an error while reading the label of the specified disk.	Check integrity of the specified disk.
-HARDWARE ERROR DETECTED WHILE READING PSEUDO-PACK TABLE ON DISK <disk-name>	The utility has encountered an error while reading the PPIT of the specified disk.	Check integrity of the specified disk.
CANNOT RUN DSKUTL	The utility cannot run. The reason is given in the second message displayed.	Examine the reason given in the second message.
<disk name> IS NOT A PHYSICAL DISK.	An attempt was made to run DSKUTL on a pseudo-disk.	Check input and re-enter.
-<character-string> IS NOT VALID INPUT	The input to the utility is not what it expected.	Correct input.
-LABEL OF DISK IS NOT CMS FORMAT, PLEASE USE DSKUTL IN OR RF IGNORE OPTION	IGNORE was not specified in the initiating message and the disk being reformatted does not contain a CMS format label in sector 0.	Use IGNORE option.
PLEASE RESERVE (RD) DRIVE, THEN INSERT DISK FOR <function> AND THEN ASSIGN (AD)	Self-explanatory.	Use RD to reserve and AD to assign drive. For details of AD and RD see appropriate section.
-HARDWARE ERROR DETECTED WHILE READING SECTOR <number> ON DISK IN DRIVE <drive-id>	The utility has encountered an error while reading the specified sector on the specified disk.	Check integrity of the specified disk and replace media if necessary.
DISK IN DRIVE <drive-id> IS LABELLED <disk name>, PLEASE ENTER OK IF <function> IS TO CONTINUE	The utility has found a disk with the specified name, enter "OK" if DSKUTL should use the specified disk.	Enter "OK" to continue DSKUTL or nothing to stop DSKUTL.
-ONLY FIXED DISKS MAY BE USED FOR A FIXED DISK SYSTEM	UNIT.FDA was specified and a disk other than a 201-I or 211 has been AD'ed to the utility.	Check input and re-enter.
DISK IN DRIVE <drive-id> WILL BE REFORMATTED AS PHYSICAL DISK <disk name>	Self-explanatory.	None.

(continued)

Message	Possible Causes	Suggested Action
-HARDWARE ERROR DETECTED ON TRACK 0	The utility is unable to read or write to one or more of sectors 0-31.	Replace disk medium.
-HARDWARE ERROR DETECTED ON MTR SECTOR <number>	The utility is unable to read or write to an MTR sector.	Replace disk medium.
-BOOTSTRAP FILE <file-name> NOT CMS STANDARD	The bootstrap file is not the one expected by the utility.	Copy a good version of the bootstrap file.
-HARDWARE ERROR DETECTED WHILE READING DIRECTORY	The utility has encountered a read error on the directory of the disk being reformatted.	Use DSKUTL IN to initialize the disk, or on B 1000 use stand-alone PACK.INIT or CART.INIT.
-DIRECTORY IS NOT VALID, PLEASE USE DSKUTL IN OR RF IGNORE OPTION	Self-explanatory.	Re-run the utility with the IGNORE option.
-TOO MANY BAD ENTRIES IN THE AVAILABLE TABLE, PLEASE USE DSKUTL IN	The utility has found more than 90 bad entries in the disk being reformatted.	Use DSKUTL IN, or on B 1000 use stand-alone PACK.INIT or CART.INIT.
-NO SPACE ON DISK TO ALLOCATE DIRECTORY FOR SPECIFIED NUMBER OF FILES	The utility cannot find an available area large enough to accommodate the directory as specified.	Check input and re-enter with a smaller number of files.
UNABLE TO <function> DISK IN DRIVE <drive-id>	The utility is unable to IN or RF the disk in the specified drive. The reason is given in the next message displayed.	See second error message for reason.
-BOOTSTRAP IS <bootstrap-name> VERSION <version id>	Normal output to tell which bootstrap is used.	None.
-NUMBER OF BAD SECTORS IS <number>	Normal output to tell how many sectors are unusable.	None.
-PLEASE RUN CHECK.DISK ON <disk-name> WHEN READY	Normal output after an RF.	None.
DISK IN DRIVE <drive-id> REFORMATTED AS <disk-name>	Normal output to indicate the successful reformat.	None.
SECTOR(S) FROM <sector number> NOT LISTED, BEYOND END OF THE	The specified sectors are not valid for the specified disk.	Check input and re-enter.

(continued)

Message	Possible Causes	Suggested Action
DISK IN DRIVE <drive-id>		
PLEASE RESERVE (RD) DRIVE, THEN INSERT INPUT DISK FOR COPY AND THEN ASSIGN (AD) DRIVE	The utility asks for the input disk to be made available for copy.	Use RD <drive-id> to reserve and AD <mix-no> <drive-id> to assign the drive.
PLEASE RESERVE (RD) DRIVE, THEN INSERT OUTPUT DISK FOR COPY AND THEN ASSIGN (AD) DRIVE	The utility asks for the output disk to be made available for copy.	Use RD <drive-id> to reserve and AD <mix-no> <drive-id> to assign the drive.
-DISKS IN DRIVE <drive-id> AND <drive-id> ARE DIFFERENT	The AD'ed drives are of a different type.	Check correct devices have been AD'ed.
-HARDWARE ERROR DETECTED WHILE WRITING SECTOR <number> TO DISK IN DRIVE <drive id>	The utility has encountered a write error on the specified drive.	Check integrity of output disk using KA utility. If necessary, re-initialize output disk.
SPECIFIED DISK-ID FOR COPY IGNORED, INPUT DISK <diskname> IN DRIVE <drive id> HAS PSEUDO-PACK TABLE	The utility cannot handle a disk with a PPIT.	None. Utility limitation.
-MORE THAN <number> HARDWARE ERRORS DETECTED ON DISK IN DRIVE <drive-id>	The utility has encountered more than 31 sectors with errors while trying to write the directory of a 1 MB mini disk or more than 49 errors on any other type of disk.	Check output medium and replace if necessary.
-CANNOT ADD PHYSICAL DISK TO SYSTEM	The utility cannot run RF to add physical disk to existing fixed disk system.	Check integrity of existing fixed disks.
SPECIFIED DISK-ID FOR COPY IGNORED, INPUT DISK IN DRIVE <drive-id> IS NOT CMS FORMAT	An output disk-id has been specified in the initiating message and a non-CMS disk has been AD'ed for copying.	None. Information only.
<file-name> NOT COPIED TO DISK <disk-id> IN DRIVE <drive-id>	The utility has encountered an error on the disk file header of the input disk.	Check integrity of the input disk.
UNABLE TO REPLACE BOOTSTRAP ON DISK <disk-id>	The specified disk is not suitable for the requested bootstrap.	The reason for this message is given in the second message displayed.
-NO INPUT SPECIFIED	There was no reply in response to accept.	Correct input and re-enter.

Message	(continued) Possible Causes	Suggested Action
DISK IN DRIVE <drive-id> WILL BE INITIALISED AS PHYSICAL DISK <disk-id>	A fixed disk is being initialized as a part of a fixed disk system.	None.
DISK IN DRIVE <drive-id> INITIALISED AS <disk-id>	Successful completion of initialize.	None.
-HARDWARE ERROR DETECTED WHILE INITIALISING DISK	The initialize routine has encountered an error.	For details see the other messages.
-INIT.TRACK COMMUNICATE NOT SUPPORTED	There is a mismatch between the utility and the MCP.	Check that the utility and the MCP are of the right level.
ENTER THE MAXIMUM NUMBER OF FILES THAT THE DISK SHOULD CONTAIN (A NUMBER LESS THAN 2805)	Self-explanatory.	Use AX to enter number of files.
ENTER THE SERIAL NUMBER FOR THE DISK (6 DIGITS)	Self-explanatory.	Use AX to enter serial number.
ENTER THE NAME OF THE DISK (UP TO 7 LEGAL CHARACTERS) OR "UNIT.FDA" IF DISK FOR FIXED DISK SYSTEM	Self-explanatory.	Use AX to enter disk name.
ENTER THE NAME OF THE DISK OWNER (UP TO 14 CHARACTERS DELIMITED BY "")	Self-explanatory.	Use AX to enter owner's name.
-IN DOES NOT SUPPORT THIS TYPE OF DISK	A request was made to perform an IN on a disk pack.	None.
BOOTSTRAP NOT COPIED TO DISK <disk-id> IN DRIVE <drive-id>	The utility has encountered an error while copying sector 2-31 to the output disk.	Check input disk.
DISK <disk-id> IN DRIVE <drive-id> MAY NOT BE USED FOR MTR ROUTINES	The utility has encountered an error while copying the MTR areas to the output disk.	Check input disk.
AREAS MAY BE MISSING FROM DISK <disk-id> IN DRIVE <drive-id>	The utility has encountered a read error on the available table or a file header was found whose name list entry does not correspond.	Run "SQ" <disk-name> VERIFY" on the specified disk after completion of the copy.
INITIALISATION PHASE	The specified surface check	None.

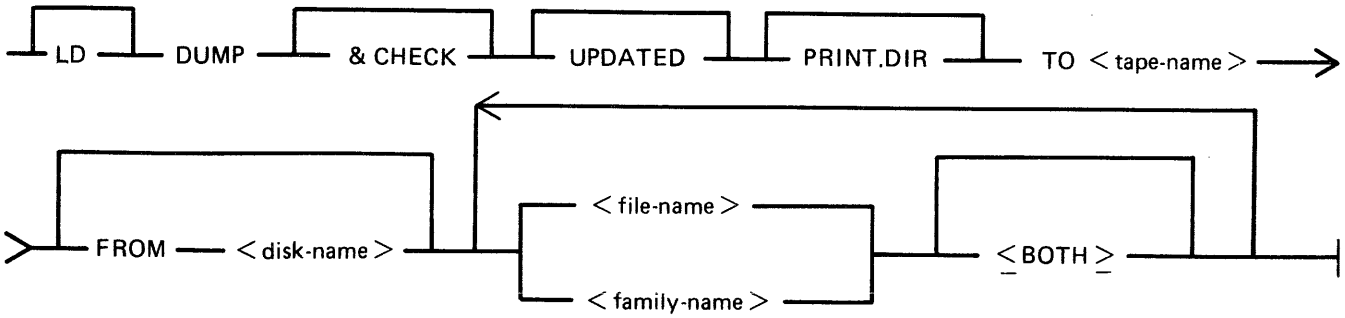
(continued)

Message	Possible Causes	Suggested Action
<number> COMPLETED ON DISK IN DRIVE <drive-id>	is completed. This message will appear three or four times depending on the type of disk to be initialized.	
-PLEASE USE A VERSION OF BOOTSTRAP FILE WHICH CONTAINS A COLDSTART BOOTSTRAP	A request has been made to initialize or reformat a disk for coldstart use, but there is no valid bootstrap on the disk.	Copy a good version bootstrap from a back-up disk.
CANNOT READ LABEL OF DISK IN DRIVE <drive-id>, WILL ASSUME DISK IS BEING INITIALISED FOR THE FIRST TIME	Sector zero of the disk being initialized is not recognized by the utility.	None. Information only.
-ONLY MINI DISK MAY BE USED FOR COLDSTART	An attempt was made to replace the bootstrap of a non mini disk by a coldstart bootstrap.	Correct input and re-enter.
-IN FUNCTION IS VALID ONLY FOR B90 AND B900 SYSTEMS	An attempt was made to use the IN function on a B 1900.	Use CART.INIT or PACK.INIT to initialize a cartridge or a pack on a B 1900.
-COLDSTART OPTION IS VALID ONLY FOR B90 SYSTEM	An attempt was made to use the coldstart option on a system other than B 90.	None. Utility limitation.
-DISK WAS NOT INITIALISED FOR USE ON A B90 OR B900 SYSTEM	An attempt was made to replace the bootstrap on a disk which was not initialized on a B 90 or B 900 system.	None.
BOOTSTRAP ON DISK <disk-id> REPLACED BY <type> VERSION <version>	Successful replace of the bootstrap.	None.
IS DISK FOR COLDSTART USE? ENTER Y OR N	Self-explanatory.	Use "AX" to enter Y or N. Initialize the specified disk with
-DISK WAS NOT INITIALISED FOR MTR USE	An attempt was made to replace the bootstrap by an MTR bootstrap but the disk is not in a suitable state to accept that.	Initialize the specified disk with the MTR option.

DUMP (Dump Files from Disk to a Library Tape) (LD – TAPE LIBRARY UTILITY)

This function is part of the utility LD. DUMP which provides the capability of copying files or groups of files from the disk specified by disk-name. If no disk-name is specified, then the system disk is assumed.

Railroad diagram:



NOTE

A purged tape must be on line and ready for use by this utility. Purged tapes can be produced using either the PG or SN intrinsics on an on-line and ready tape.

& CHECK — If & CHECK is specified, CHECKADUMP will be zipped (without pause and display) from LD when the dump operation is completed.

UPDATED — If UPDATED is specified, only updated or newly created files will be dumped, that is, files with the "Modified" bit in the DFH set. (An LR listing will identify which files have the bit set.) Refer to Archiving using UPDATE Facility for a fuller explanation.

PRINT.DIR — If PRINT.DIR is specified, an accurate tape directory is printed at the end of the dump. The list of files is the same as in the directory at the head of the tape. For each file dumped, a list of its attributes is printed. If any file is not dumped, the reason why will appear beside the name. When PRINT.DIR is specified, all messages of the form:

"<file-name> DUMPED" are suppressed.

Displays will be limited to exception conditions, for example, if a file cannot be dumped or loaded for some reason.

TO <tape-name> — This is the name which is given to the tape.

FROM <disk-name> — If the option FROM disk name is specified, the files are dumped from the disk named <disk-name>. If this option is not used, then the system disk is assumed.

<file-name> <family-name> — This is the file or family of files to be dumped to tape. Groups of files may be dumped by separating the file names or family names by a comma "," or a space " ".

<BOTH> — If <BOTH> is specified immediately following a request to dump a key file then, provided that the pertinent data file is on the same

disk, the data file following the key file on the tape will also be dumped.

Since the DUMP function is part of the utility LD, "LD" is actually what appears in a mix message. To discontinue the DUMP function, "DS <mix-number>/LD" must be used.

Examples:

Example 1. To copy all the files in the family named TEST from the system disk to a tape named TPTEST, the following initiating message is used:

LD DUMP TO TPTEST TEST=

Example 2. To copy all the key and data file pairs in the family MILK, which have been created or updated since the last time they were saved, onto tape or disk from the user disk DKMILK to the tape named TPMILK, the following initiating message is used:

LD DUMP UPDATED PRINT.DIR TO TPMILK FROM DKMILK MILK= <BOTH>

The addition of the keyword PRINT.DIR produces an accurate directory at the end of the dump.

Example 3. To copy the files ARCFILE1 and ARCFILE2 from the disk named ARCDISK to the tape named ARCTAPE, the following initiating message is used:

LD DUMP & CHECK TO ARCTAPE FROM ARCDISK ARCFILE1, ARCFILE2

The addition of the keyword & CHECK causes the utility CHECKADUMP to be executed after the dump is successfully completed.

NOTE

For details of the creation of a Coldstart tape for a B 90 system, refer to the COLDSTART utility.

Output Messages

For output messages, refer to LD utility.

DUMPADISK (Disk Dump)

The DUMPADISK utility is provided to enable periodic back-up and retrieval of user files to disk. The utility executes under MCP control and requires the open unlabelled disk facility in order to run. The B 90 MCP level 3.04 and above supports this function. DUMPADISK also requires the presence of the local language file (SYSLANGUAGE).

All reference to pseudo-pack is for information only. The B 90 system does not support pseudo-pack.

NOTE

DUMPADISK replaces the DD utility. The disk format created by DUMPADISK is unique and builds a non-standard disk directory. Disks created with the DD utility are NOT recognized by the DUMPADISK utility. As DD will not be issued for any releases after 3.03, users with disks created with DD must convert their back-up media to disks created with DUMPADISK.

The files created with the DUMPADISK utility can only be accessed using the DUMPADISK LOAD or ADD function. These disks can, however, be duplicated using the DSKUTL COPY function (see the description of DSKUTL earlier in this section). Disks used for back-up must be pre-initialized in CMS format and must not contain any user data. Any files existing on a disk written to by DUMPADISK will be lost when the DUMPADISK disk directory is written.

The utility also creates a security record when DUMPing to back-up disks. This record must be present on any disk used for a LOAD function.

DUMPADISK provides the facility to create one or a numbered sequence of back-up disks during one dump session.

Example:

In dumping files from a 2011 disk to 1MB disk, several 1MB disks may be required to accommodate all the required data. During the DUMP or UNLOAD function, replacement disks will be called for as each back-up disk becomes full. The directory of each disk will contain entries for all the files contained on that disk, and disks which have been created previously in that dump session. Thus in a series of three back-up disks created in the same session:

The directory of disk 1 has entries for all files on disk 1.
The directory of disk 2 has entries for all files on disks 1 and 2.
The directory of disk 3 has entries for all files on disks 1, 2 and 3.

When retrieving files from back-up disks, all files DUMPed may be LOADed successively, or a selection of individual or groups of files may be LOADed. In all cases, it is advisable to initiate any LOAD from the last disk of the sequence. In this way, if any files required for LOAD are not resident on that disk, the utility will call by name for the disk which contains those files.

When creating back-up disks, the utility will request a four-character prefix for disk identification purposes. Each disk created will be labelled with this prefix concatenated with a three-character decimal number, progressing from 001 to 999. During subsequent retrieval, all disks will be called for by their seven-character label (user-specified prefix plus decimal suffix). Back-up disk labels (sector 0) are created in CMS format.

Files can be DUMPed to and LOADed from any CMS supported removable disk media (except ICMD), and DUMPed or LOADed from and to any CMS supported disk.

The new "& CHECK" option can be specified, which causes the utility to compare copies of files DUMPed or LOADed with the originals. If any discrepancies are discovered, the new copy will not be made permanent.

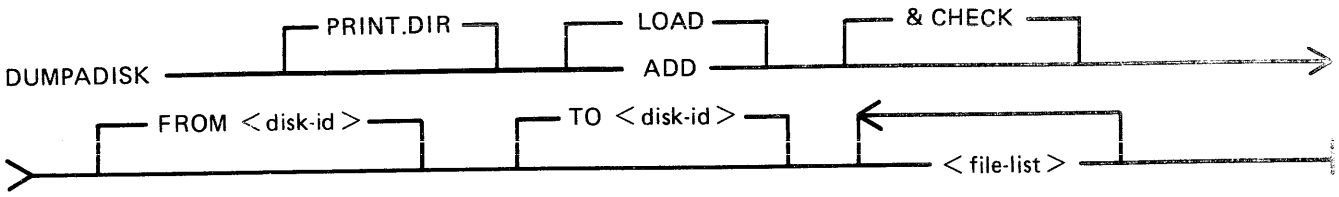
Format:

Format 1

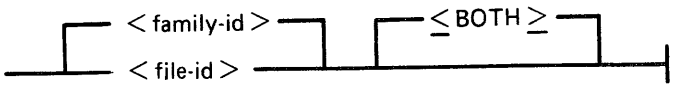
DUMPADISK PRINT.DIR

This function lists the disk directory.

Format 2



where <file-list> is defined as :



This function enables back-up disks to be created together with the option to print the directory of the complete dump session.

Format 2 dumps files, or groups of files, from a disk to a backup disk, or a series of backup disks. Each backup disk must be made available to the utility via an AD command (note the disk must have been reserved using the RD command), when the utility attempts to open an unlabelled disk file. If the option PRINT.DIR is specified, at the normal EOJ the utility prints the directory of the files dumped during this current session.

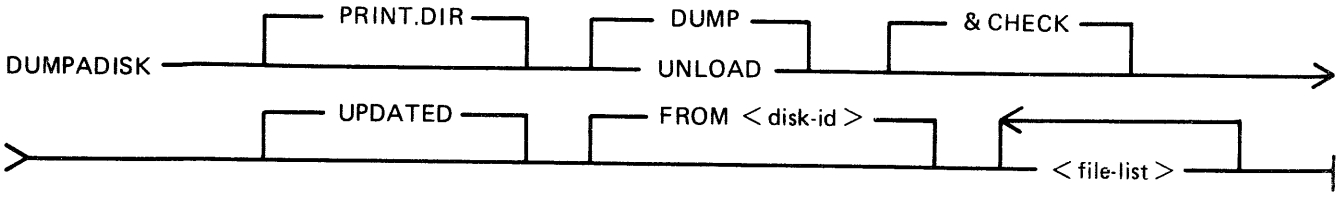
If the UPDATED option is specified, only files specified in the <file list> which have the UPDATED bit set in their Disk File Header (DFH) are dumped.

NOTE

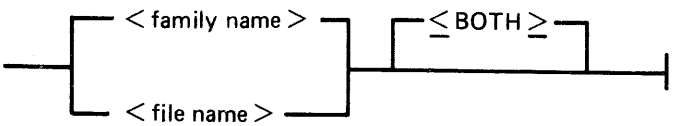
See Archiving Using UPDATE Facility for a fuller explanation.

Use of the DUMP option will copy a file or files to backup medium only. Using the UNLOAD option will copy a file or files to backup medium and remove the original copy from disk.

Format 3



where < file list > is defined as :



This function enables retrieval of backed-up files to any named disk.

Format 3 loads back files from a named disk to a named disk. The input disk must have been created previously by DUMPADISK in order for the files to be loaded. ADD only loads files if they are not already present on the output disk.

If the PRINT.DIR option is specified, a hard copy listing is produced at the end of the run, indicating whether or not the LOAD/ADD request on the file by file basis has been successful.

NOTE

Once a disk has been used as a dump disk, it must be reformatted or re-initialized before it can be used for any other purpose.

For formats 2 and 3, <file-list> is defined as a comma-list of individual file-ids, family-names (for example, ABCD=), or just = to specify all files on a disk.

NOTE

If indexed pairs are to be DUMPed or LOADed, then the <BOTH> option must be included in the file-list following the indexed pair file-name.

Formats 2 and 3

&CHECK — This option causes DUMPADISK to check each DUMPed or UNLOADed file against the original and if any data corruptions occur, the file in question will not be made permanent.

Format 1: PRINT.DIR

By specifying only PRINT.DIR in the initiating message, the utility prints the directory of the disk assigned to the utility.

Operation

The utility will open-unlabelled a disk and prompt the operator to reserve (RD) and assign (AD) the disk drive containing the relevant disk.

The utility prints the directory and goes to EOJ.

Format 2: DUMP or UNLOAD

PRINT.DIR	This option in the initiating message causes the utility to print the directory of all files DUMPed during the session when it goes to EOJ.
FROM disk-name	This option allows the operator to specify the disk-name from which the files are to be DUMPed or UNLOADed. If this option is omitted from the message, the utility searches the system disk for the specified files.
file-list	This entry specifies the individual files or groups of files to be DUMPed or UNLOADed. The form of this entry can be a comma-list of individual files, a comma-list of file families (FILE=), or just = to DUMP all files from the specified or default disk.
UPDATE	DUMPADISK will DUMP/UNLOAD only the files which have been modified if the keyword UPDATED is used in the initiating message.

Operation

On executing, the utility first builds an internal directory or list of files to be DUMPed, and then displays the message, if the name was not given in the initiating message in the TO option.

"PLEASE ENTER A FOUR CHARACTER DISK NAME WHICH WILL BE USED TO IDENTIFY YOUR DUMP DISKS"

The operator must now enter the prefix using an accept (AX) command.

The operator is now prompted to Reserve (RD) and Assign (AD) the disk drive in which the output disk is located. If the current disk is filled before the DUMP is complete, the operator is informed that it is OK to PO drive <Dxy>. The utility then prompts the operator to replace the disk by displaying the following message:

- PLEASE POWER OFF DISK IN DRIVE <drive-id>
- PLEASE INSERT AND ASSIGN (AD) NEXT DISK TO CONTINUE THE DUMP

The operator must PO the existing disk and replace it with another disk.

On completing the DUMP, if the PRINT.DIR option was entered in the initiating message, the directory for the dump session is output to the line printer before the utility goes to EOJ.

Format 3: ADD or LOAD

FROM disk-id	This entry specifies the back-up disk-id from which files are to be LOADed or ADDed. When entering this name for a sequence of back-up disks, the name of the last disk in the sequence should be used. If the specified files contained in the <file-list> option of the initiating message are not on this disk, the utility will call for the required disk by name.
TO disk-id	This option allows the operator to specify the disk to which files are LOADed or ADDed.
file-list	This entry specifies the individual files or groups of files to be LOADed or ADDed. The form of this entry can be a comma-list of individual files, a comma-list of file families (FILE=) or just = to indicate a LOAD of all files on the specified disk.

Operation

The disk from which the ADD or LOAD is to take place must be READY before entering the initiating message.

If any of the specified files for LOADing are not resident on the disk, the available files are LOADed and the utility prompts the operator to supply the required disk. When all files have been LOADed or ADDed, the utility goes to EOJ.

NOTE

Files will only be ADDed if there is no file of that name already on the disk being ADDed to. A LOAD will replace existing files.

Error Recovery

If a read/write error is detected during DUMPing, the utility automatically attempts to reallocate disk space for the file currently being DUMPed. This process is repeated up to ten times for each file on any one disk. If a parity error is detected when attempting to write to a newly re-allocated area of disk, then the utility attempts to write to another area and ignore the previous attempt. Therefore, the utility attempts to re-allocate up to ten non-contiguous areas per file on each disk. If this limit is reached, the dumping of the current file is terminated (see the following list of error messages. The utility continues by dumping the next file, if there are any more to be dumped.

If an error is detected during a multifile ADD or LOAD, the utility terminates the LOAD for that file and continues to ADD or LOAD remaining files.

If a read/write error occurs in the directory area of a disk, the utility terminates.

Output Messages

Message	Possible Cause	Suggested Action
<file-id> LOADED	Normal operation. File with name <file-id> is successfully loaded.	None.
<file-id> DUMPED	Normal operation. File with name <file-id> is successfully dumped.	None.
<file-id> NOT LOADED	Specified file is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message given.
<file-id> NOT DUMPED	Specified file is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message given.
<family> NOT LOADED	Specified family is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message.
<family> NOT DUMPED	Specified family is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message.
-ASSOCIATED DATAFILE <file-id> NOT FOUND	The <BOTH> option was used, but the datafile of an	Check the input-disk for the presence of the specified files.

(continued)

Message	Possible Cause	Suggested Action
	indexed pair cannot be found. The keyfile is not DUMPed.	Re-run the utility.
-INCONSISTENT FILE SIZE AT DUMP TIME	DFH file-size and number of records DUMPed or LOADed are not equal.	Investigate discrepancy and re-run the utility.
-FILE IS OPEN SHARED OR FOR OUTPUT USE	Another program is using the specified file.	Wait for the file to be released by the program.
-FILE HAS BEEN ALTERED	The file specifications have been changed since the utility was started.	Check status of file not dumped and re-run the utility.
-FILE HAS NOT BEEN UPDATED	The UPDATED option is specified for a file which has not been updated.	Check status of file not dumped.
- INTERMITTENT HARDWARE ERRORS DETECTED ON DISK <disk-id>	Write error which cannot be resolved has occurred during dumping of files.	Use another disk
-HARDWARE ERROR DETECTED WHILE WRITING TO DISK <disk-id>	The utility has encountered a write error on the output disk. If the utility continues, it has been able to recover.	None.
-REQUESTED FILES WERE NOT FOUND	The specified files are not in the directory of the specified disk.	Check initiating message for correct filenames.
-HARDWARE ERROR DETECTED WHILE READING DISK <disk-id>	The utility has encountered a read error on the input disk. If the utility continues, it has been able to recover.	None.
-FILE IS ALREADY ON DISK	The specified filename already exists in the directory of the disk.	None.
-ASSOCIATED DATAFILE IS ON ANOTHER DISK	The datafile of an indexed pair is not on the specified disk. The utility will continue, but the keyfile is not LOADed.	Check the initiating message and print the back-up disk directory.
-FILE WAS NOT SUCCESSFULLY DUMPED	The specified file is not dumped because the file has changed or moved since the start of the utility.	Check file status and re-run utility.
INCORRECT INPUT TO DUMPADISK	There is an error in the initiating message. The explanation is given in the second message displayed.	Look for the explanation in the next sentence displayed by the utility.

(continued)

Message	Possible Cause	Suggested Action
-INVALID CHARACTER IN <identifier>	One of the characters is outside the range A to Z, 0 to 9, "." or "-".	Correct input and re-enter.
-REQUEST IS INCOMPLETE	At least one word in the initiating message is missing or misspelled.	Correct input and re-enter.
-FILE NAME <file-id> EXCEEDS 12 CHARACTERS	The specified filename is more than 12 characters long.	Correct input and re-enter.
-<specified option> IS NOT A VALID OPTION	The UPDATED option is used in the initiating message for LOADing or ADDing.	Correct input and re-enter.
-NO FILE NAME SPECIFIED	There is no file name specified in the initiating message.	Correct input and re-enter.
-PARAMETER FOLLOWING KEYWORD <string> IS INVALID	The utility has encountered an error in the initiating message.	Correct input and re-enter.
-CANNOT HANDLE A REQUEST OF THIS LENGTH	The input message exceeds 601 characters.	Correct input and re-enter.
-IS NOT A VALID INPUT	There is an error in the initiating message.	Correct input and re-enter.
-FROM EXPECTED	The utility expects the word FROM when ADDing or LOADing.	Correct input and re-enter.
-DISK NAME <disk-id> EXCEEDS 7 CHARACTERS	The diskname <disk-id> in the initiating message is too long.	Correct input and re-enter.
-OPTION <specified option> HAS BEEN SPECIFIED MORE THAN ONCE FOR FILE <file-id>	At least one of the options occurs more than once in the initiating message.	Check input and correct.
-OPTION <specified option> HAS BEEN SPECIFIED MORE THAN ONCE FOR FILE <family-id>	At least one of the options occurs more than once in the initiating message.	Check input and correct.
-DISK <disk-id> CANNOT BE ACCESSED FOR THIS RUN	There is a diskname specified in the file list which is not the same as the name following the FROM option.	Correct input and re-enter.
-NO DISK NAME SPECIFIED	There is no diskname specified after the word "TO" or "FROM".	Correct input and re-enter.

Message	(continued) Possible Cause	Suggested Action
–FILE HAS BEEN REMOVED	The utility is unable to dump the file while it is removed.	Check status of files not dumped and re-run the utility.
–FILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	The specified file is not available because another program is using it.	Wait until the other program has finished, then re-enter.
–HARDWARE ERROR DETECTED WHILE WRITING TO FILE <file-id>	The utility has encountered a write error on the output disk from which it cannot recover.	Check output medium.
– HARDWARE ERROR DETECTED WHILE READING FILE <file-id>	The utility has encountered a read error on the input disk from which it cannot recover.	Check output medium.
–HARDWARE ERROR DETECTED WHILE READING PSEUDO-DISK TABLE ON DISK <disk-id>	An error is encountered during a read on the PPIT of the specified disk.	Check integrity of specified disk.
–DISK <disk-id> DOES NOT BELONG TO THIS LOAD	The newly inserted disk is not a part of the group of disks to be LOADed.	Check the disks to be LOADed and re-enter.
–DISK <disk-id> IS NOT RECOGNIZED BACKUP DISK	The directory of the specified disk is not of a type dumpadisk uses.	Check the disks to be LOADed and re-enter.
–HARDWARE ERROR DETECTED ON DIRECTORY OF DISK <disk-id>	The utility found an error from which it cannot recover on the directory of the specified disk.	Check integrity of specified disk.
–UNLABELLED OPEN HAS NOT BEEN IMPLEMENTED ON THIS SYSTEM	The utility is used with an incompatible level of system software.	Check the system software is of level 3.4 or higher.
<file-id REMOVED	The utility has found a duplicate filename during a LOAD, or removes the file as a result of an UNLOAD.	None.
–NO FILES TO <DUMP/LOAD>	None of the files specified in the initiating message is suitable for DUMPing or LOADing.	Check input and re-enter.
–DUMPADISK CANNOT HANDLE MORE THAN 2804 FILES, LIMIT REACHED AT FILE <file-id>	A request was made to DUMP or LOAD more than 2804 files.	Check input and divide the files to be loaded or dumped into two groups.
–FAMILY MAY BE INCOMPLETE	Some of the requested files are on a pseudo-disk,	Ready the disk and re-enter.

Message	(continued) Possible Cause	Suggested Action
	pertaining to a physical disk which is not ready.	
-FAMILY FOR <family-name> MAY BE INCOMPLETE	Some of the requested files are on a pseudo-disk pertaining to a physical disk which is not ready.	Ready the disk and re-enter.
COMPARISON ERROR DETECTED WHILE CHECKING <file-id>	The "& CHECK" option has been specified and a discrepancy has been found.	Look at second sentence of message to find reason for error.
-ERROR OCCURRED AT RECORD <number>	Self-explanatory. Further explanation is given in next sentence of message.	Check file integrity.
-DISK FILE HEADERS DO NOT MATCH	A comparison error has been found on the directory of the file.	Check disk integrity.
-PLEASE POWER OFF DISK IN DRIVE <drive-id>	The current disk is full during DUMPing.	Use "PO" to power off the specified drive.
-PLEASE INSERT AND ASSIGN (AD) NEXT DISK TO CONTINUE THE DUMP	Self-explanatory.	Insert new disk and use "AD <mix-no> <drive-id>" to continue.
-PLEASE POWER OFF DISK <disk-id>	The current disk has been LOADed.	Use "PO" to power off the specified drive.
-PLEASE INSERT DISK <disk-id> TO CONTINUE	Self-explanatory.	Insert requested disk.
DISK <disk-id> IS NOT REQUIRED FOR THIS LOAD	The files to be loaded do not reside on this disk.	Look at the second message given for the disk name requested by the utility.
-PLEASE MAKE DISK <disk-id> AVAILABLE AND ENTER OK, OR MAKE ANOTHER DISK AVAILABLE AND ENTER ITS NAME	Information only. An additional disk is required to continue the LOAD.	Insert requested disk.
-PLEASE ENTER A FOUR CHARACTER DISK NAME WHICH WILL BE USED TO IDENTIFY YOUR DUMP DISKS.	The name of the dump disk has not been entered in the initiating message.	Use "AX <mix-no> <4 char. disk-id>" to inform the utility of the name of the dump prefix.
-INSUFFICIENT AREA TO DUMP TO ON DRIVE <drive-id>	After dumping the directory of files there is no room left to dump anything else. This disk cannot be used for dump.	Use a larger type of disk to dump the specified files.
-DISK LABEL OF <disk-id> IS NOT CMS FORMAT	The label of the specified disk is not in a recognized CMS	Check the integrity of the specified disk. dump the

(continued)

Message	Possible Cause	Suggested Action
DUMP TO ON DRIVE <drive-id>	format. of files, there is no room left to dump anything else. This disk cannot be used for dump.	specified files.
-DISK LABEL OF <disk-id> IS NOT CMS FORMAT	The label of the specified disk is not in a recognized CMS format.	Check the integrity of the specified disk.
-FILE HAS NOT BEEN CHECKED FOR COMPARISON ERRORS	The file is split over two back-up disks and cannot be compared.	None.
END DUMPADISK	Normal message when DUMPADISK is at End Of Job.	None.

DUMPANALYSE

The DUMPANALYSE utility is for use by CMS COBOL, RPG and MPL programmers. It provides a method of obtaining B 90 and B 1000 program dumps from programs that have been abnormally terminated and gives an analysis to be used as an aid to locating logic errors.

The dump may be analyzed remotely if the interactive dump analyzer is executing in a datacomm environment via a menu interface. If it is executed in a non-datacomm environment, a complete analysis to the printer will be performed.

To understand a program dump, the user must be familiar with the basic concepts of CMS and with the s-code of the language in which the program is written. This is described in the CMS MCP Reference Manual (form number 2007555).

NOTE:

1. This DUMPANALYSE utility is common to B 90 and B 1000 systems. For B 900 systems, refer to section 9 "B 900/CP 9500 Dependent System Software".
2. To understand a program dump, a compilation listing of the program must be available.
3. Setting certain dollar options during compilation helps in reading program dumps. These dollar options are discussed later.

Obtaining the Dumpfile

When a program encounters an execution time error, a DS/DP message occurs and a dump file can be created by the DP command:

DP <mix-number>/<program-name>

Alternatively, if a program is known to be looping, a DP of the program can be forced in order to identify the conditions associated with the loop.

On B 1000 systems only:

The DM command may be used to create a dump of a running program. To do this, enter:

DM <mix-number>/<program-name>.

When the dump file has been created, the program is suspended. To re-start the program, enter:

GO <mix-number>

The dump file created is named DMFILnn, where nn is a two digit decimal number representing the mix-number of the DP'd task.

The dump file contains the program parameter block (PPB) of the code file of the executing program, the overlayable segments of code and data, and certain memory tables used by the MCP for this task (the locked slice).

Initiating The Analysis

Initiating Interactive Analysis

If the analyzer is run in a datacomm environment and remote analysis is required, the initiating message will be:

[illegible] PN DUMPANALYSE

[illegible] the control character.

[illegible] screen displayed, the user is required to enter the dump file name. If an invalid dump file name is given, an error message will be displayed and the user has the opportunity to re-enter his information.

If the dump file to be analyzed is found to be a COBOL or RPG program dump file, the next screen displayed will then allow the codefile and/or namefile to be identified, so that COP Table Analysis and/or Namefile Analysis respectively may be performed.

[illegible] Systems

[illegible] the codefile and namefile fields are blank, then no COP table or NAMEFILE analysis is available, but the analysis continues.

B 1000 Systems

Codefile field:

The second screen that is displayed prompts the user to enter the codefile name. This field must not be left blank.

[illegible] analyzer displays the default codefile name that was recorded in the PPB when the program was compiled. If the codefile name has changed, enter the new name in this field.

If the specified codefile name cannot be found, a message is displayed and the analyzer remains suspended until a valid name is entered. The codefile contains information which is used by DUMPANALYSE, and must be on-line during the analysis.

Namefile field:

If this field is left blank, no namefile analysis is available for COBOL/RPG dumps, but the analysis continues.

The entire interactive analysis is menu driven, so specific sections of the analysis may be examined remotely by selecting the appropriate menu option. Prints of various sections or specific pages may be obtained, as well as a complete printed analysis, if required.

Any error messages, resulting from invalid entries in any of the forms fields, are displayed between the menu list and the "SELECT MENU OPTION" entry, and the user may re-enter his information.

Depending on the dump type, either an MPL menu page or COBOL/RPG menu page is displayed. Unless otherwise specified, screens and analyses are the same for both dump file types.

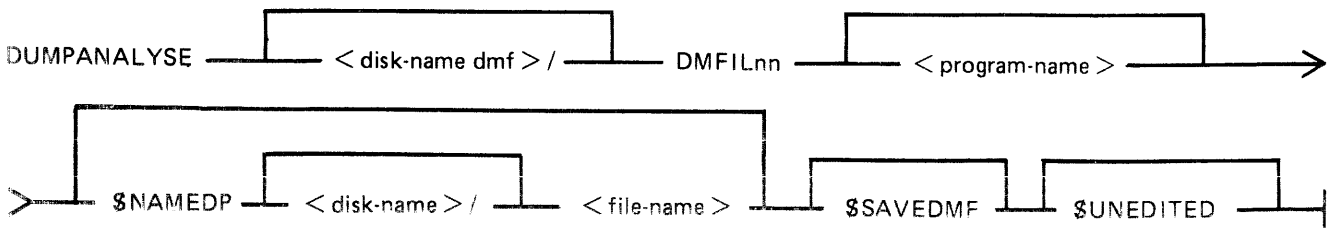
The menu screens for B 90 and B 1000 systems are similar. However, where differences exist they are specified in the following table.

Menu Option	Explanation
1	Terminates the session and displays the termination screen on the user's terminal.
2	Displays the screen containing the Program Parameter Block Information

Menu Option		Explanation
3		Displays the screen containing the information of the S-Interpreter Work Area.
4	COBOL/RPG	Displays information about the PERFORMS and/or subroutines that were current at the time the dump was taken.
	MPL	Displays the lexical level and the values of DISP containing a pointer to the data stack.
5	COBOL/RPG	Displays Current Operand Table (COP) Analysis including data item numbers and values.
	MPL	Displays the data segments, starting from segment 0, in hexadecimal and ASCII.
6	COBOL/RPG	Displays list containing the namefile and Data Values if the namefile is supplied by the user.
	MPL	Displays the Control Stack list.
7	COBOL/RPG	Allows the user to specify a particular FIB: For B 90 systems enter external filename. For B 1000 systems enter internal filename. If no FIB is specified, the analysis is given in the data segment table order.
	MPL	Allows the user to examine the Data Stack Contents for a specified segment and procedure number. The default for PSN/SPN is the segment and procedure for the level at which the dump was taken.
8	COBOL/RPG	Allows the user to specify a particular data segment number. The default is "ALL" which starts analyzing the data segment number at segment zero.
	MPL	Allows the user to specify a particular FIB. For B 90 systems enter the external filename. For B 1000 systems enter the internal filename. If no FIB is specified, the analysis is given in the data segment table order.
9		Starts display of the unedited dump.
10	COBOL/RPG	Explains the Communicate Responses of the dump file.
	MPL	Explains the Fetch Value of the dump file.
11		Prints the complete dump in a format similar to that appearing on the screens.

Initiating Non-Interactive Analysis

DUMPANALYSE produces a printed analysis of this dump file. This is achieved by using the following initiating message:



disk-name dmf	This is the name of the disk on which the dump file resides.
DMFILnn	This is the name of the dump file.
program-name	This is the name of the program which was DP'ed, it is required for COP Table Analysis of COBOL/RPG programs. For analysis of B 1000 system dumps, either this filename or the default name, must be on-line.
$NAMEDP	This dollar option is used with COBOL or RPG programs to provide source related dump analysis. Note that the COBOL or RGP program must be compiled with the relevant dollar options (see COBOL and RPG Dependent Dump Analysis). The name of the file which the compiler produced must be specified if this option is used.
$SAVEDMF	This dollar option will prevent DUMPANALYSE from removing the dump file when DUMPANALYSE has finished with it.
$UNEDITED	This dollar option will cause DUMPANALYSE to generate only an unedited dump.

NOTE

If the program-name (codefile) is not specified and one is required, the default is assumed. The default is set when the program is compiled. If the program-name has been changed, the new name must be specified.

If the default codefile is not found:

- On B 90 systems a warning message is displayed and the analysis continues.
- On B 1000 systems a warning message is displayed and the utility terminates.

Examples:

The following sample initiating messages relate to a program called OBJECT.A, with an internal program-name of CODE.A. The mix-number of the program was 02 and the name of the file created by the dump is DMFIL02.

For each example, the response for B 90 and B 1000 systems is given.

1. DUMPANALYSE DMFIL02

B 90 systems	– DUMPANALYSE produces the printed analysis of DMFIL02 and removes the dumpfile after use.
B 1000 systems	– DUMPANALYSE terminates before any analysis is performed, because the default program-name was not found. The dump file is not removed.

2. DUMPANALYSE DMFIL02 OBJECT.A $SAVEDMF

B 90/B 1000 systems	– DUMPANALYSE produces the printed analysis of DMFIL02 including COP Table Analysis for COBOL/RPG dumps, for the file specified (OBJECT.A) The dumpfile is not removed after use because the $SAVEDMF was set.

3. DUMPANALYSE DMFIL02 $NAMEDP COB.A.NAMES

B 90 systems	– DUMPANALYSE produces the printed analysis of DMFIL02 including analysis of the file COB.A.NAMES. If the namefile is available, namefile analysis is listed. COP Table Analysis is not performed because the default program-name was not found.

B 1000 systems – DUMPANALYSE terminates before any analysis is performed because the default program-name was not found.

4. DUMPANALYSE DMFIL02 OBJECT.A $NAMEDP COB.A.NAMES

B 90/B 1000 systems – DUMPANALYSE produces the printed analysis of DMFIL02, including COP Table Analysis for the specified program-name (OBJECT.A). Analysis of the file COB.A.NAMES is also given.
DUMPANALYSE removes the dumpfile after use.

Analysis Common to All Program Languages – Interactive/Non-interactive Analysis

The dump analysis is a reflection of the state of the interpreter and the state of the task itself. Some methods of representing the information on the listing are explained here.

Each line of a hex memory dump is printed in the following format:

<offset> <text> "<ASCII representation>"

The <offset> applies to the first byte and is relative to the base of the hex memory dump being printed. It is given in four digit hex and decimal representation.

@10BC@/00444

The <text> will consist of 16 hex byte values, divided into groups of four, for display on the screens, or 32 hex byte values when full printed analysis, is required. The <ASCII representation> is the ASCII character representations of the hex byte values in the line. Blanks are inserted where no such equivalent exists. For any sequence of lines which contains the same values, the first two lines are printed, followed by a single line with the text "SAME AS LINE ABOVE".

The analysis listing is headed by a title the run date and the release level. Where source related dump analysis is used, the name of the related source file is also listed. The various sections are described as follows:

Reason for Dump

This is the decoded reason for the abnormal termination of the task.

Program Parameter Block Information

The fields of the Program Parameter Block (PPB) are decoded with the hexadecimal equivalent of the values printed beside these values. For a further explanation of the contents of the PPB refer to the CMS MCP Reference Manual (form number 2007555).

Also contained in this section are several other fields of information.

Dump File Name

This gives the user the pack name on which the file resides and the file name of the dump file for later reference.

Locked Location Start

This is a pointer to the location of the code file's locked slice. The value of this field is the dump file's logical record number in which the start of the locked slice is located.

Task Name

This is only available on the B 90 analysis

This is a task originator identifier of the aborted task. A breakdown of this byte could be useful to the user.

This byte is arranged as follows:

most significant 3 bits : SOURCE field
next 4 bits : TASK field
least significant bit : TASK-FLAG

1. The source field can have the following values and meanings:

   100 – The task was loaded as the result of a keyboard input message.

   001 – The task was loaded as the result of a ZIP from another task.

2. If the TASK.FLAG is set, the TASK field contains the task-name of a task which is to be re-started when the current task reaches end-of-job. If the TASK.FLAG is reset, the TASK field is ignored and no task is to be restarted when the current task reaches end-of-job.
3. If a task performs a ZIP with PAUSE of another task, the TASK field for the zipped task is loaded with the TASK field of the zipping task and the TASK-FLAG of the zipped task is set. If a task performs a ZIP without PAUSE, the TASK-FLAG of the zipped task is reset and the TASK field is ignored.

Interpreter Preset Area

Also listed for COBOL and RPG dumps are the contents of the EDIT table used to contain the editing constants for the program. This is an 8-byte field.

File Information

For each file associated with the task, the File Information Block (FIB) and File Parameter Block (FPB) are analyzed. The file-name is printed in the heading for each analyzed FIB.

NOTE

For B 90 systems the external file-name is printed. For B 1000 systems the internal file-name is printed.

(Details of files and file attributes can be found in the CMS MCP Reference Manual, form number 2007555).

Unless otherwise specified, the FIB information is available for both B 90 and B 1000 systems.

On the menu screen, option 7 or 8 may be used to request a specific FIB. If these options are used, specify the external file-name for B 90 systems and the internal file-name for B 1000 systems.

NOTE

Data comm is not handled on a file basis.

Closed Files

If the file was closed, or half closed, at the time of the dump, the following information is provided:

1. File Parameter Block Analysis.

2. FIB segment number.
   This is the data segment number for the File Information Block.

3. FPB segment number.
   This is the data segment number for the File Parameter Block.

4. File state
   This specifies if the file was closed or half closed.

Open Files

If the file was open at the time of the dump, the following information is provided:

1. File Parameter Block Analysis

2. FIB segment number
   This is the data segment number for the File Information Block

3. FPB segment number
   This is the data segment number for the File Parameter Block

4. File state
   OPEN/CLOSE
   For OPEN files on the B 90 further information about the state is given as follows:

MYUSE	(input, output or both)
OTHERUSE	(free, shared, lockaccess or lock)
ORGANIZATION	(sequential or indexed)
ACCESS	(sequential or random)

   – On B 1000 systems, this information is given for OPEN files in the File Parameter Block under the adverb for OPEN.

5. Work area
   If a work area exists, the following information is provided:
   – the number of the data segment assigned to the work area
   – the offset into that segment of the work area base
   – the length of the work area (B 90 systems only)

6. Records per buffer
   This is the number of records contained in a buffer or block.

7. Record size
   This is the size of records in bytes.

8. Block size
   The size of the buffer or block in bytes.

9. Buffer pointer (B 90 systems only)
   This holds an offset into the current buffer to the next available record or, for stream access mode, to the next available byte.

10. Device kind
    As defined in the CMS MCP Reference Manual (form number 2007555).

11. Adverb for close (B 1000 systems only)
    This gives the close attribute type.

12. Adverb for open (B 1000 systems only)
    This gives the open attributes of the file.

13. Flags (B 1000 systems only)
    This gives the FPB flags description
    For example, MUSTBACKUP.

14. File type (B 1000 systems only)
    This gives the CMS file type, for example
    DATA, KEYFILE, PRINTERBACKUP

15. Number of buffers (B 1000 systems only)
    This gives the number of buffers requested for this file.

Analysis of disk files provides the following extra information:

16. Sectors per buffer (B 90 systems only)
    This is the number of sectors contained in each buffer.

17. Maximum record written (B 90 systems)
    Highest record number (B 1000 systems)
    This is the actual number of records in the file.

18. Maximum record declared (B 90 systems only)
    Maximum file size (B 1000 systems)
    This is the maximum number of records the file can contain.

19. Disk address details (B 90 systems only)
    Unit and sector numbers (absolute sector address).

20. Area number log being processed (B 90 systems only)
    A disk file can have a single area, or be spread over a maximum of 16 areas.
    This number relates to the entry in the disk file header for the particular disk area in use when the dump was taken.

21. For stream access files (B 90 systems only)
    - spare bytes in the last record:
      If the last record of this file was not completely filled, the number of these spare bytes is given.
    - record number at start of current buffer:

    For other types of access

    - current record number
      This is the logical record number at the start of the current buffer.

22. External file identification (B 1000 systems only)
    this gives the filename on disk as diskname/file-name

Finally, a dump of all the file buffers which exist can be examined or printed. The current buffer is dumped first and then the remainder in sequence up to the last buffer.

Data Segment and Unformatted Dump

These sections appear in the analysis listing after the language dependent section, but are themselves language independent.

The data segments are printed as hex dumps of each segment. If a particular data segment is an FPB or FIB, it is noted as such in the analysis.

The unformatted dump is a listing of the locked slice of the task.

For B 90 systems the locked slice contains:
- Task Control Block (TCB)
- Interpreter work area
- Control stack
- Data segment table
- Task FIBs
- All locked segments

For B 1000 systems the locked slice contains:
- Program segment table
- Data segment table
- Task control block preset area
- Task control block extension
- Control stack
- Code control block (COP table in COBOL/RPG)
- Internal file-name block

The following paragraphs describe those parts of a COBOL program dump that relate specifically to the COBOL s-language, RPG s-language and MPL s-language.

COBOL Dependent Dump Analysis

This section describes those parts of a COBOL program dump that relate specifically to the COBOL-S language.

Dump-Related Dollar Options

To obtain full information from a COBOL dump, it is necessary to set certain of the dollar options at compile time. These are:

LIST

Gives a listing of source and corresponding line number, identifies a COP number for each data-name and prints the COP table.

CODE

Lists Standard COBOL-S

OPTCODE

Lists COBOL-S and other code file information.

COP-TABLE

Lists the COP-TABLE transformation, the PPB and the segment table numbers in COBSVERTER.

LINE-MAP

Provides a mapping between object code addresses and source line numbers.

LINE-CODE

The compiler generates code to update the COBOL-S LINE-NUMBER register.

DUMP disk-name/file-name

Causes the compiler to generate a workfile. This file is processed by COBSVERTER to produce a name-file which DUMPANALYSE uses to output, in place of the COP-TABLE listing, a listing of the source-names and their current values at the time the program goes DS/DP. To use this name file with DUMPANALYSE, include the $NAMEDP option in the initiating message of DUMPANALYSE. More information about this name-file is included at the end of the section on language related dump analysis.

For a complete description of these dollar options, refer to the CMS COBOL Reference Manual Relative to Release 3.05 form number 2033007).

To locate the line number on which the dump occurred, it is necessary to set either the LINE-MAP or LINE-CODE dollar options when the program is compiled. When LINE-CODE is set, the line-number is indicated in the LINE-COUNT area of the Interpreter Work Area. LINE-MAP causes COBSVERTER to list the mapping of code addresses to line-numbers. When a dump is produced the line-number of the failure point can be deduced from the Current Code Segment Pointer.

To obtain information about the contents of data-names, it is useful to set the dollar options LIST CODE and OPTCODE (or COP-TABLE). By setting CODE and LIST it is possible to identify the STANDARD COBOL-S code generated for a particular statement. COBSVERTER now changes the significance of COP-TABLE numbers: some COP-entries are removed from the STANDARD COBOL-S COP-TABLE and replaced by in-line code; most COP-TABLE entries are re-ordered. The mapping of STANDARD COBOL-S COP-TABLE entries to the COP numbers found in the dump file is printed by setting the options OPTCODE or COP-TABLE. To find the values contained in data items which do not appear as COP entries, the appropriate segment/displacement must be located. COP-TABLE entries can be replaced by actual source data names by:

- compiling the program with the dollar option DUMP in the program source
- and using the related name file in the initiating message of the DUMPANALYSE, or specifying the namefile on the screen prompt.

COBOL Segment Allocation

Data Segments

Segments 0 and 1 and the last segment are used as run-time compiler work segments. For each file declared, an FPB segment, FIB segment and file buffer segment must be allocated (except that the file buffer is not necessary for console files). The specification of SAME RECORD AREA allocates a common buffer segment. WORKING-STORAGE data items are allocated segments depending on the dollar option DSSIZE (or default).

Code Segments

Code segments are allocated sequentially by the compiler.

If segment numbers in section headers are not used, all code will go in segment 0. If segment numbers appear in section headers, each time a new number is encountered, it is assigned a new code segment. For modular compilations, segment numbers in the source are local to the module. When bound, each module's segment numbers are mapped to consecutive code segment numbers in the order that the modules are bound.

COBOL S-Interpreter Work Area

This contains information about:

the version of the interpreter used
current address where the program was dumped
the last communicate message issued by the program
the state of the Overflow Register and the Line Count Register
the contents of the table of characters used for editing
the reason for the dump
the last Communicate response
the error number.

Control Stack

This contains information about the PERFORMS and/or subroutine calls that were current at the time the dump was taken.

K is the value issued by the COBOL compiler to a particular PERFORM statement.

SEGNO is the segment number of the calling code segment.

DISPL is the displacement in that code segment of the call or PERFORM.

LINE COUNT is the contents of the line count register at the time of the call or PERFORM.

COP Table (Current Operand Table)

This is a table of descriptors of data used by the program. Note that some descriptors are in-line within the code segments.

The descriptor is printed symbolically and, for descriptors which are not subscripted or indexed, the contents of the field are listed in hexadecimal.

Most of the listing is self explanatory; however, the meaning of the data type codes is as follows:

UNS8 – Unsigned 8-bit alphanumeric
UNS4 – Unsigned 4-bit Binary coded decimal
SGN8L – Signed 8-bit numeric overpunched leading sign
SGN8R – Signed 8-bit numeric overpunched trailing sign
SGN4L – Signed 4-bit BCD numeric leading sign
SGN4R – Signed 4-bit BCD numeric trailing sign

These data type codes can also contain an indication of whether the field uses an alternate alphabet. For example:

UNS8 – Unsigned 8-bit
UNS8E – Unsigned 8-bit with reverse escapement

UNS8A – Unsigned 8-bit with alternate graphics
UNS8EA – Unsigned 8-bit with reverse escapement and alternate graphics

Namefile Table

This is a table of descriptors of data used by the program. It gives the same information about the descriptors as the COP Table, except that the actual field names used in the program are given instead of COP numbers.

Any compiler generated COP Table entries will not, however, appear in this table as they cannot be identified by name.

In the COP table analysis only COP entries which appear in the COP Table can be analyzed. Any COPs which have been moved in-line in the S-code by COBSVERTER cannot be analyzed. In the namefile, however, all user generated descriptors will appear in this table even although they may be moved in-line by COBSVERTER.

RPG Dependent Dump Analysis

This section describes those parts of an RPG program dump that relate specifically to RPG. The S-language is the same as the COBOL-S language.

The COP table information provided under COBOL Dependent Dump Analysis is also relevant to RPG.

Dump-Related Source Constructs

The following RPG source constructs and dollar options are available to aid the user in the search for the cause of the dump:

1. The use of XMAP, MAP or PARMAP options of the COBSVERTER phase of the compiler produces a mapping of line numbers to segment/address for calculation, input and output lines.
2. The use of column 15 of the H-specification (DEBUG) generates code to set a line-number register, the contents of which are available in the dump analysis.
3. To interpret the data in terms of source field names, set the dollar options, NAMES and MAP. The field denoted by COP in the NAMES listing is used as the OLD COP in the COP TABLE of the MAP listing, and the position and attributes of the actual data can be obtained.
4. To produce a full commented code listing, use the dollar option XMAP. This produces a code listing which includes line number indications, subroutine names and other comments in the logic cycle.
5. Using the dollar option(s) NAMEDP and/or PACKDP in an RPG source causes the generation of a namefile. This namefile is used by DUMPANALYSE to replace COP TABLE references with actual source declared data names. NOTE : not all data names are placed into a COP TABLE. To use the name-file with DUMPANALYSE, include the dollar option $NAMEDP in the initiating message of DUMPANALYSE, or, on the screen, enter the namefile when prompted.

For more detailed information on these dollar options, refer to the CMS RPG Language Reference Manual Relative to Release 3.05 (form number 2033130).

Object Program Organization

The following points should be noted about RPG object program organization:

1. For each file (except data communications and console files) three data segments are allocated, numbered in the range 1 to as many as needed. These are used as the file buffer, FPB and FIB. For data comm, there are only two segments; the file buffer, and the CD areas. For console files, the FPB is segment 2, the FIB is segment 3, and the file buffers are allocated separately.
2. Literals are generally allocated to read-only segments.
3. Data segment zero contains edit strings and routine system work-fields.

Further Information

For additional information common to COBOL, RPG and the interpreter, refer to "ANALYSIS COMMON TO ALL LANGUAGES" earlier in this section. Refer also to the CMS COBOL Language Reference Manual Relative to Release 3.05 (form number 2033007) CMS RPG Language Reference Manual Relative to Release 3.05 (form number 2033130). Interpreter information is to be found in the CMS MCP Reference Manual (form number 2007555).

MPL Dependent Dump Analysis

These paragraphs describe those parts of an MPL program that relate specifically to the BIL S-language. The CMS compilers (written in either BIL or MPLII) also produce dumps which can be analyzed with the aid of these paragraphs.

MPL S-Registers

Refer to the CMS MCP Reference Manual (form number 2007555) for a description of the MPL S-language. The S-registers at the time of the dump are analyzed in the program dump, as follows:

1. PSN

   Program Segment Number; contains the number of the current code segment.

2. SPN

   Segment Procedure Number; contains the number of the procedure within the segment PSN, that was executing at task termination.

3. PCA

   Program Current Address; holds an offset into the segment PSN of the current S-instruction.

4. Offset Within Procedure

   This is the offset of the current S-instruction relative to the start of the code of procedure number SPN.

5. NLD

   Number of Local Descriptors; contains the number of occurrences of descriptors declared local to the currently active procedure.

6. VSN

   Virtual Segment Number; one or more virtual segments can exist whose pages map onto a set of data segments with VSN holding the segment number corresponding to the first page

of the first virtual memory segment. Virtual segments may be declared in MPL by the SEGMAP construct in which a set of segments can be referenced as one segment (see the CMS MPLII Language Reference Manual, form number 2007563).

7. MODE

This register reflects the state of the operation of the S-machine. It can be in one of three possible states:
– PROCESS, DATA.DECL or REMAP.DECL
If the state of operation is either DATA.DECL or REMAP.DECL then the following registers are analyzed.

– SEGN : holds the data segment number of the data which is being declared or remapped.

– SOL : holds the offset into SEGN of the start of the area allocated to the previously declared descriptor.

– EOL : holds the offset into data segment zero (the data stack), for the location of the next descriptor,

STA

This register holds an offset into data segment zero for the next available byte on the data stack. This is used as a pointer to administer the allocation of space on the data stack.

8. FETCH VALUE

This is a 24-bit register which holds the result of the previous MCP communicate (the FCM).

9. COMMUNICATE PARAMETER AREA

This is the last external communicate sent by the interpreter to logical Input/Output on behalf of the user's program. The communicate consists of three parts – VERB, OBJECT and ADVERB. The fetch value described above reflects the success or failure in the communicate.

10. PARTIAL STACK LENGTH

This gives the length in bytes of the control stack (this information can also be found in the PPB).

11. DISPLAY

The vector DISP is printed. There are two columns: one indicating the lexical level; the other the values of DISP containing a pointer to the data stack.
At most, 16 regions or areas of descriptors are accessible to a task during execution. DISP contains offsets into data segment zero to the bases of the procedure parameter descriptors and the locally declared descriptors for each of the active procedures which are currently in scope. The LVL register indicates the current lexical level of the S-machine.

12. REGION 1 AND REGION 2

These reflect the most frequently used lexical levels in the procedure which was active at the time of abnormal termination. This information is useful only when the code segment has to be decoded.

13. LEVEL

    This gives the current lexical level at which the next instruction resides.

14. CARRY

    This reflects the value of the CARRY 16-bit field used by the MPL compiler in arithmetic operations.

Formatted Code and Data Stack

This is an analysis of each of the active procedures at the time of task termination.

Volatile Registers

During the dynamic invocation of a procedure, all the volatile registers: PSN, SPN, PCA, LVL, NLD, REG1, REG2, and also the DISP (of the new value of LVL), are saved on the control stack. The functions of all these registers have already been explained. The functions of the registers REG1 and REG2 are similar to those of register LVL, except that they indicate the lexical level of the "most referenced level" and the "next most referenced level", respectively.

Descriptor Analysis

For each descriptor, the following information is printed:

1. Descriptor address :

   The offset within data segment zero of the base of the descriptor.

2. Descriptor :

   A hexadecimal representation of the descriptor.

3. Occurrence Number :

   The descriptor occurrence number as encountered on the pass 3 listing, as determined by the order of declarations in the source program.

4. Data Type :

   The decoded type of descriptor.

5. Seg/Origin :

   The data segment number in which the actual data resides and the offset of the base of the data from the segment base. The value of the @3F@ for the segment number applies to self-relative descriptors.

6. Data :

   A hexadecimal dump of the data. In the case of fixed or self-relative data type, this is the hexadecimal value and the converted decimal value (plus ASCII representation or blank). If the segment number or the value of the offset within a segment is too large for the segment, an error message is printed.

Relation to MPL Compilation Listings

During the compilation of an MPL program, various listings can be obtained, namely the source statement listing, the XMAP listing and the Procedure Correspondence listing.

1. Pass ONE Listing :

   This listing is of the source statements. At the beginning of each record there are three columns of figures. These are:

   - The procedure occurrence number of the record.
   - The nested level of the record.
   - The descriptor occurrence number; that is, the number of parameters and local descriptors so far encountered within the current procedure.

2. Pass TWO Listing :

   From pass two of the compiler, two listings can be obtained:

   1) XMAP

   This contains the occurrence number of the procedure, the procedure name, the code segment number and the procedure number within this code segment (that is, PSN/SPN). For each procedure, the sequence number of each source statement is printed. The offset of the code generated by this source statement from the base of the procedure, is printed in parenthesis at the left of the sequence number.

   2) The Procedure Correspondence Listing

   This lists all procedures, giving the procedure occurrence number, the procedure name, the code segment number of the procedure's S-instructions, the number of the procedure within that segment (that is, PSN/SPN) and the length in bytes of the S-Code generated by the procedure statement.

Location of Program Failure Point

1. Using PSN/SPN Registers:

   The values of these registers can be used with either the XMAP listing or the Procedure Correspondence Listing to determine the name of the procedure and its location on the pass three listing.

2. Using PCA Register:

   From this register value, the offset of the current S-instruction relative to the base of the procedure code can be calculated.
   At the base of each code segment there is a procedure table with one two-byte entry for each procedure.
   If the number of bytes in this table is added to the number of bytes taken up by the preceding procedures in the code segment (obtained from the Procedure Correspondence listing), and the result is subtracted from the value in the PCA, then the offset of the current S-instruction from the base of the procedure code is the final result.
   This offset can be used to locate the source listing line with the aid of the XMAP listing.

Description of Namefile for Use in Source Related Dump Analysis

This namefile can be used by DUMPANALYSE to replace the COP-NUMBER entry in the COP-TABLE listing with the respective data-names used in the related source. (not all names appear in the COP-TABLE).

The namefile produced by the compiler is a sequential file of 60 byte records. It consists of several record types as follows:

1. Control Record

   (Provided in first release)

Bytes	Content
0	@01@ for CONTROL RECORD
1	@00@ reserved for implementation level if required
2	@01@ if the file has been updated by by COBSVERTER
3-14	The program name as in bytes 1-12 of the PPB
15-20	Compile date as in bytes 51-61 of the PPB
21-23	Compile time (HHMMSS)
24-29	Compiler version and user version
30	@01@ COBOL @02@ RPG
31-32	Record number of the start of entries for symbolic names
33-34	Number of records containing entries for symbolic names
35-39	Reserved for future use and will initially contain binary zeros

2. Data Item Record (COBOL/RPG)

Bytes	Content
0	@02@ for Data Item record
1	Declaration level 1-99 in binary
2-31	User specified name (this may be blank for compiler specified fields)
32-33	Line number
34-45	Record offset of fathers entry from the start of the symbolic entry table. If no father this entry will contain @FFFF@
36-51	COBOL-S descriptor
52-53	Cop number. If no cops exist this will contain @FFFF@
54-59	Reserved and will contain binary zeros

3. File Entry (COBOL/RPG)

Bytes	Content
0	@03@ for a file entry
1-30	User specified file name
31	Segment number of FPB
32-59	Reserved and will contain binary zeros

Output Messages

The following messages from the utility may appear on the SPO when running the non-interactive analysis. When running the interactive analysis, these and other messages which are self-explanatory may appear on the screen to inform the user of an exception condition.

Message	Possible Causes	Suggested Action
WARNING BIL DUMP FL, SECOND PARAMETER IN INIT.MESS IGNORED	This message only appears on B 90 when program-name has been specified for a dump file from an MPL program. On B 1000, MPL dumps assume a default program-name if one is not specified.	None.
PACK ID TOO LONG	The disk name specified for the disk on which the namefile is to reside is invalid because it contains too many characters.	Check input and re-enter.
FILE ID TOO LONG	The file name specified as the namefile has been entered with too many characters.	Check input and re-enter.
INVALID CHARACTERS IN FILE	Invalid characters have been used in the initiating message (for example in a filename).	Check input and re-enter.
IN USE	A specified file is already in use by another program.	None.
NOT FOUND <reason>	A specified file could not be accessed. <reason> is an interpretation of the fetch value returned to DUMPANALYSE specifying why the file could not be accessed.	None.
MFID NOT PRESENT FILE IN USE, OTHERUSE CONFLICTS BAD FILE TYPE DISK NOT FOUND DUPLICATE PACK DISK ERROR DISK LOCKED PARITY ERROR	These are reasons for the FILE NOT FOUND condition.	
INTERP. NOT SUPPORTED	The DP'd file cannot be recognized by DUMPANALYSE. This is probably due to an object file not being COBOL object, RPG object or MPL object but could also be due to a corrupt file being used.	Check file.
ATTEMPTING UNEDITED DUMP	This message is generated after "INTERP. NOT	None.

(continued)

Message	Possible Causes	Suggested Action
	SUPPORTED". DUMPANALYSE will attempt to generate an unedited dump.	
NO SPECIFICATION GIVEN	This is output when the initiating message given is incomplete.	Check input and re-enter.
INVALID CODEFILE – NOT COMPATIBLE WITH DUMPFILE	This is output when the dumpfile supplied is not a dump of the program-name.	Check input and re-enter.

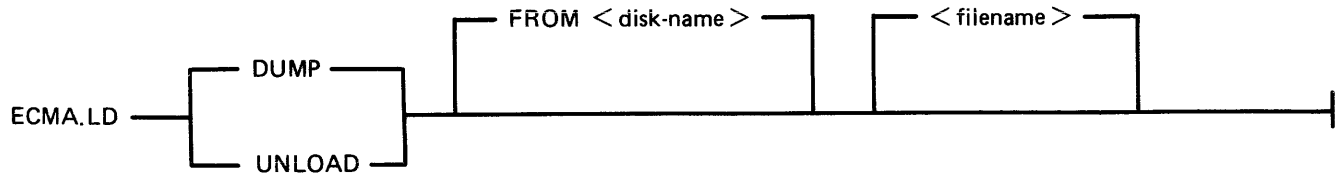
ECMA.LD (Load/Dump of ECMA Tape Files – B 90/B 900 Only)

This utility allows the operator to structure tape files according to ECMA BASIC and ECMA COMPACT systems as specified in the STANDARD ECMA-41 publication.

The ECMA tapes are treated as unlabelled tapes in the CMS system. The utility is initiated in two different ways for BASIC system and COMPACT system.

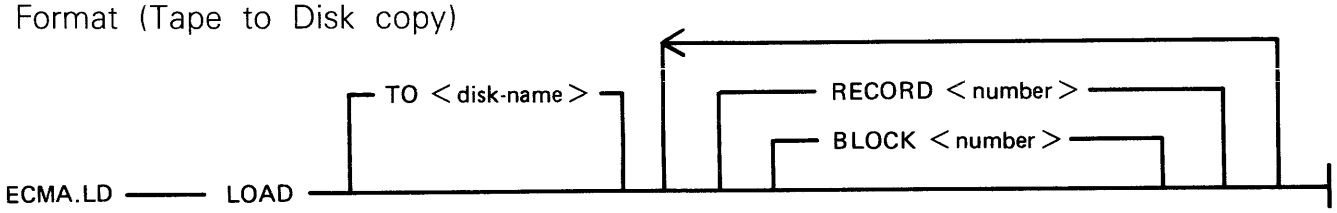
Basic Initiation

Format (Disk to Tape copy)



The files are copied from the disk to a purged tape. If the option UNLOAD is specified, the files copied are removed from the disk.

Format (Tape to Disk copy)



The files are copied from an ECMA BASIC tape to the disk specified by disk-name. The option RECORD followed by a number specifies the record size in bytes on the tape and BLOCK followed by a number is the blocking factor; that is, number of records per block.

If the RECORD and BLOCK options are used, the first file on tape is read according to the first attributes, the second file is read according to the second attribute etc. If the block size (that is, record size x number of records per block) exceeds 256, then an error message is issued and loading is not performed.

If RECORD and BLOCK options are not used, tape files are loaded as 256 byte records blocked 1.

The names of the ECMA tape files loaded to the disk become ECMA001, ECMA002,....ECMA00n.

Examples:

To copy files IN001, IN002 from the systems disk to a tape in ECMA BASIC format:

```
ECMA.LD DUMP IN001, IN002
```

To copy file AR030 from the disk ARDISK to tape and remove after copy:

```
ECMA.LD UNLOAD FROM ARDISK AR030
```

To copy files PR200, record size 80 block 160, PR120, record size 60 block 180, to the system disk from ECMA BASIC tape on device CTA:

ECMA.LD LOAD RECORD 80 BLOCK 2, RECORD 60 BLOCK 3

The utility will display the following message:

mix no/ECMA.LD <14> WAITING UNLAB ECMATAP/NONE AT DEVICE REQUIRED

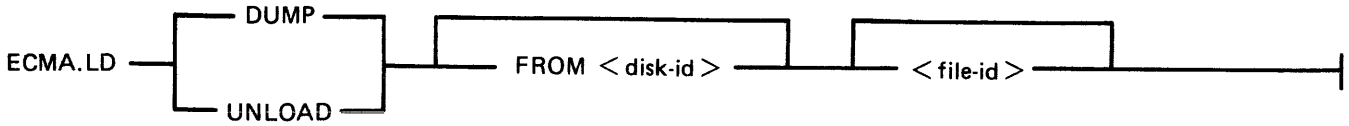
Then enter:

AD mix number CTA

The files PR200 and PR210 will be called ECMA001, ECMA002 on the disk.

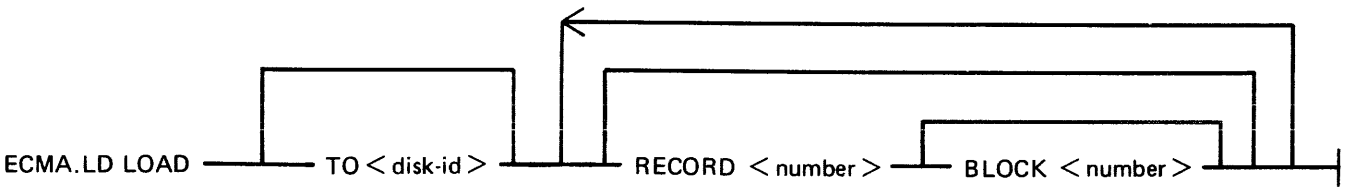
Compact Initiation

Format (Disk to Tape copy)



The files are copied from the disk to tape. If the option UNLOAD is used, files copied are removed from the disk.

Format (Tape to Disk copy)



The files are copied from an ECMA COMPACT tape to the disk specified by disk name. If any file is already present on the disk, it will be removed before the same named file is copied. The option RECORD followed by a number specifies the record size in bytes on the tape and BLOCK followed by a number is the number of records in a block.

If RECORD and BLOCK options are used then the first file on the tape is read according to the first attribute and the second file is read according to the second attribute. If block size (that is, record size x number of records per block) exceeds 256 then an error message is given and utility will terminate.

If RECORD and BLOCK options are not used, then tape files are loaded as 256 byte records blocked 1.

The ADD option will copy a file to the disk only if that file is not already present.

Examples:

To copy files IN001, IN002 from the disk INDISK to a tape FRED in ECMA COMPACT format:

ECMA.LD DUMP TO FRED FROM INDISK IN001, IN002

To copy files IN010 from the system disk to a tape FRED and remove from the disk after copy:

ECMA.LD UNLOAD TO FRED IN010

To copy files IN001, IN002 from tape FRED on CTA, which is created in ECMA COMPACT format, to the system disk:

ECMA.LD LOAD FROM FRED IN001 RECORD 80 BLOCK 2, IN002

The utility will display the following message:

Mix number/ECMA.LD <14> WAITING UNLAB ECMATAP/NONE AT DEVICE REQUIRED

The response should be:

AD mix number CTA

Output Messages

Message	Possible Causes	Suggested Action
NOT A BASIC ECMA TAPE	Loading is requested from a BASIC tape but the tape is not a BASIC format.	Check the tape and re-input.
NOT A COMPACT ECMA TAPE	Loading is requested from a COMPACT tape but the tape is not a COMPACT format.	Check the tape and re-input.
NO FILES IN THE FAMILY <group-name> ON TAPE <tape-name> FOR DUMP/ LOAD ADD/UNLOAD	Specified files are not present on the tape.	Check the group name and/or the tape, and re-input.
NO FILES IN THE FAMILY <family-name> ON DISK FOR DUMP/LOAD/ADD/ UNLOAD	The specified group of files is not present on disk	Check the group name and re-input
NO FILE <file-name> ON TAPE <tape-name>/DISK <disk-name> FOR LOAD/ DUMP/ADD/UNLOAD	The specified file is not present on tape or on disk.	Check the file-name and re-input.
<file-name> REMOVED	This message is displayed for each file removed by a LOAD or UNLOAD.	None.
<file-name> LOADED	This message is displayed for each file added or loaded.	None.
<file-name> DUMPED	This message is displayed for each file dumped or unloaded.	None.

(continued)

Message	Possible Causes	Suggested Action
<file-name> NOT LOADED – ALREADY ON DISK	Self-explanatory	
<file-name> NOT FOUND	The file is not found on disk.	Check input and re-enter.
file-name HAS BEEN REMOVED – DUMP ABANDONED – TAPE BEING PURGED	The specified disk file to be copied to tape has been removed before the utility was able to copy it.	None.
<file-name> HAS BEEN ALTERED – DUMP ABANDONED – TAPE BEING PURGED	The contents of the file to be copied to tape were altered after the start of the dump and before the utility was able to copy it.	None.
<file-name> IN OUTPUT USE – DUMP ABANDONED – TAPE BEING PURGED	The file cannot be dumped to tape as it is being used as an output file.	None.
BLOCK COUNT ERROR	This is displayed when the BLOCK COUNT entry in the end-of-file label, end-of-track label, or end-of-volume label does not match with the actual number of blocks read during a LOAD or ADD.	Possible bad tape: retry with a different drive or with a backup tape.
INCORRECT TAPE NAME	For a COMPACT format tape, the names in the initiating message and the tape label do not agree.	Check the tape label and re-input.
NOT FIRST PART OF FILE	A section number other than 1 was found when reading the first label on tape.	
DATA BLOCK TOO LARGE	The block size on tape for a LOAD or DUMP is greater than 256 bytes.	None.
<file-name> DATA FILE NOT FOUND ON TAPE FOR LOAD	File specified for COMPACT format LOAD cannot be found.	Check file name and re-input if necessary.

FILEUTL (FILE MAINTENANCE UTILITY)

FILEUTL runs only under datacomm on a TD 830 type terminal. FILEUTL is interfaced to GEMCOS using forms mode and requires a screen size of 80 characters by 24 lines.

The utility name must be in the TCL and must have its interface set to TCMS.

The initiating message for FILEUTL is :

<cc> RN FILEUTL

where <cc> is the control character

FILEUTL provides the facilities to create a new file and update or remove an existing file. Records may be displayed, edited or added to a file. These facilities are restricted to source and data type files (#00-#0F).

A "FIND" command is provided to search for records containing specifiable key values.

It is possible to copy records from an existing file to the file currently being created or modified.

To delete records from an existing file, it is necessary to create a new file and then copy only the records which are to be maintained from the existing file to the new file.

All record input and updating is done directly to the actual file.

All user entry fields except those for disk-names, file-names and record data are numeric or hexadecimal – no English key words are used.

Indexed files are not supported in this release of FILEUTL.

The file SYS.FILEUTL contains the screen formats and messages for the utility. The filetype of this file is #47 and the first record contains a check-string and version number which are checked by the utility when the file is opened.

Screen Formats and Functions

Screen 1.	SELECT function – CREATE new source or data file UPDATE existing source or data file DISPLAY any existing type of file REMOVE existing source or data file HELP – program information TERMINATE program.
Screen 2.	Entry of attributes and options – CREATE
Screen 3.	Entry of attributes and options – UPDATE
Screen 4.	Entry of attributes and options – DISPLAY
Screen 5.	Selection of record sections – CREATE, UPDATE or DISPLAY
Screen 6.	Record input or edit – CREATE or UPDATE
Screen 7.	Record copy or display from another file – CREATE or UPDATE
Screen 8.	Record Display – DISPLAY

9.	Specify keys
10.	Find record information – CREATE, UPDATE or DISPLAY
11/12/13.	Program information – HELP

After utility has been successfully attached to a station, screen 1 is displayed.

The user is requested to enter an option number and/or parameters in entry fields on each screen display. When valid input has been entered and transmitted, the next logical screen is displayed – see routing diagram, figure 4-4.

Key to Screen Formats

fffffffffff	File identifier
ddddddd	Disk identifier
sssssss	System disk identifier
rrr	Record size
bbbbb	Block size
nnnnnnn	Actual no. of records in file
mmmmmmm	Maximum possible no. of records in file
xxxxxxx	Current record no.
yyyyyyy	Current record no. + 1
zzzzzzz	Actual no. of records in file + 1
aaaaaaaa	Data from record (ASCII)
hhhhhhhh	Data from record (hexadecimal)
‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘‘	Display line reserved for error message
date.and.time	Example – FRI 14 AUG 81 AT 11.04.56.4

These fields are protected and cannot be over-written unless enclosed in unprotected field delimiters > <.

> < Unprotected data entry field delimiters.
> 180< Unprotected Data Entry Field Delimiters with default.
Screen top and bottom.

Cursor Management

If a screen is transmitted with the cursor sitting in the first character position of the first unprotected entry field, the data in all unprotected entry fields as displayed is saved by the program.

If a screen is transmitted with the cursor not sitting in the first character position of the first unprotected entry:

1. The data, as displayed in any unprotected data fields preceding the field where the cursor is currently situated, is saved.

2. If the cursor is in the first character position of an entry field, the data in this field as originally displayed is saved. If not, the contents of the field up to the cursor are saved and the rest of that field is assumed to be blank filled.

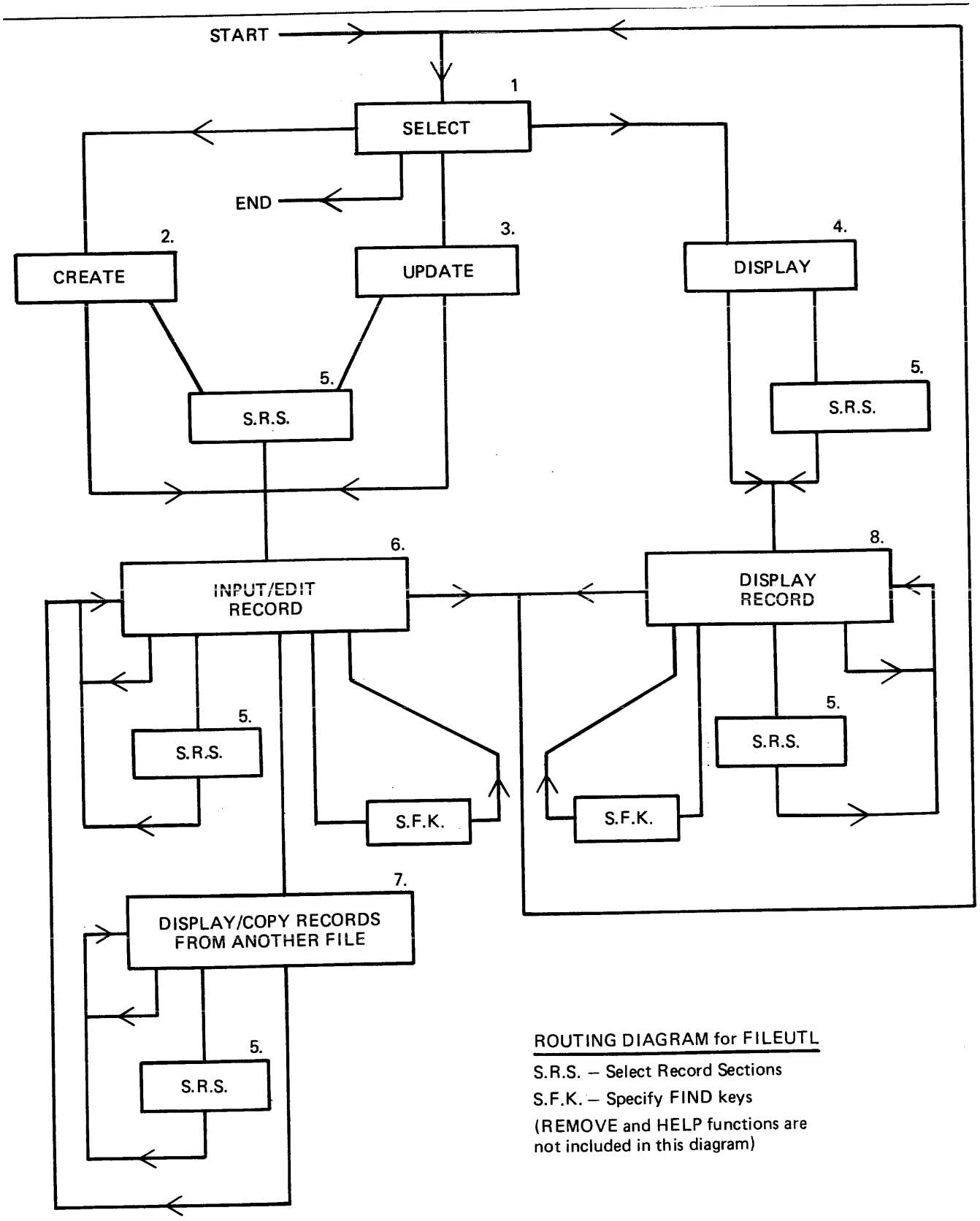


Figure 4-4. Routing Diagram for FILEUTL

3. The data as originally displayed in any following fields is saved – any modifications done to these fields are not saved.

NOTE

Record Data Entry fields in the RECORD INPUT/EDIT screen are handled differently as explained in under Screen 6.

When a screen is displayed, the cursor is positioned at the start of the most convenient entry field for ease of use on that particular screen.

When invalid data is entered and transmitted, the same screen is re-displayed and the first entry field with invalid data is highlighted with the cursor positioned at the start of that field. A message identifying the error is also displayed on the bottom line of the screen.

NOTE

The screen may be refreshed by pressing the "HOME" key, entering "0" and transmitting. For further details, see Screen 13 – HELP 3.

Screen 1 – Select

```
CMS UTILITY - FILEUTL          VERSION 3.04.01          date.and.time


ENTER "1" TO CREATE A NEW FILE
   OR "2" TO UPDATE AN EXISTING FILE
   OR "3" TO DISPLAY AN EXISTING FILE
   OR "4" TO REMOVE AN EXISTING FILE
   OR "5" IF HELP IS REQUIRED
   OR "6" TO FINISH THIS PROGRAM RUN                              >1<


IF "5" OR "6" IS ENTERED, ALL OTHER INPUT FROM THIS SCREEN WILL BE
                                                                  IGNORED.
IF CREATING A NEW FILE, SELECT THE TYPE REQUIRED.
ENTER "1" FOR SOURCE - INPUT IN ASCII
   OR "2" FOR DATA   - INPUT IN ASCII
   OR "3" FOR DATA   - INPUT IN HEXADECIMAL                       >1<

ENTER FILE IDENTIFIER                                             >          <

ENTER NAME OF DISK ON WHICH FILE RESIDES OR WILL RESIDE           >sssssss<
```

If function 1, 2, 3 or 4 is selected, a file identifier must also be entered.

The duplicate file condition is checked for at this stage in the CREATE function.

The maximum record size which can be supported by the utility is 2000 bytes in either hexadecimal or ASCII Input/Display mode. The record size of the entered file-name is checked to be within these limits in the UPDATE and DISPLAY functions.

If function 4 is selected, the utility attempts to remove the entered file-name. Only files with filetypes of #00 – @0F (data or source) and record size less than or equal to 2000 may be removed. This screen is re-displayed with a message to report whether the request was successful or not.

If function 5 is selected, three screens of useful information on this utility are displayed before screen 1 is re-displayed (see screens 11, 12 and 13).

For this screen and all other screens, if invalid input is entered and transmitted or any error condition (for example, duplicate file or record size too large) is detected, the same screen is re-displayed with a message in the bottom line identifying the error. The user should then attempt to re-enter valid input.

Screen 2 – Create

```
FILE IDENTIFIER                                    fffffffffff

NAME OF DISK ON WHICH FILE WILL RESIDE             ddddddd

CREATION DATE OF FILE WILL BE                      FRI 14 AUG 81

FILE TYPE/INPUT MODE                               <00> DATA/HEXADECIMAL

ENTER RECORD SIZE OF FILE                          >180 <

ENTER BLOCK SIZE OF FILE                           >180  <
ENTER MAXIMUM NO. OF RECORDS FILE WILL CONTAIN     >2048   <

ENTER RECORD FILLER - ONE CHARACTER FOR ASCII
                    - TWO CHARACTERS FOR HEX.      >00<

ENTER "1" TO COMMENCE RECORD INPUT
   OR "2" TO SELECT RECORD SECTIONS
   OR "3" TO RETURN TO FUNCTION SELECTION          >1<

IF "3" IS ENTERED THIS FILE WILL NOT BE CREATED.
```

Entering a zero in the first column of "RECORD SIZE OF FILE" causes the screen to refresh (see Screen 13 – HELP 3).

This example is for a data file using hexadecimal input mode.

The maximum record size permitted for a source or data file is 2000 bytes.

Defaults for source file:

The "ENTER RECORD FILL CHARACTER" option entry field is protected and contains a blank character.

Record size entry field – 80 (bytes)
Block size entry field – 720 (bytes)

"FILE TYPE/INPUT MODE" display is "<01> SOURCE/ASCII".

Defaults for data file with ASCII input:

The "ENTER RECORD FILL CHARACTER" entry field is one byte long with "0" default entered.

"FILE TYPE/INPUT MODE" display is "<00> DATA/ASCII".

When valid file attributes have been entered and transmitted, the utility opens the new file and closes it as soon as it is full. This may allow recovery of a new file in the event of a system failure while the file is being created.

Screen 3 – Update

```
FILE IDENTIFIER                                          fffffffffff

NAME OF DISK ON WHICH FILE RESIDES                       ddddddd

CREATION DATE/LAST ACCESS DATE OF FILE                   11 JUL 81/14 AUG 81

RECORD SIZE/BLOCK SIZE OF FILE                           rrr/bbbbb

NO. OF RECORDS IN FILE                                   nnnnnnn

MAXIMUM NO. OF RECORDS FILE CAN CONTAIN                  mmmmmmm

FILE TYPE                                                <00> DATA

ENTER "1" TO COMMENCE RECORD DISPLAY
   OR "2" TO SELECT RECORD SECTIONS
   OR "3" TO RETURN TO FUNCTION SELECTION                >1<

ENTER "1" FOR ASCII INPUT MODE
   OR "2" FOR HEXADECIMAL INPUT MODE                     >1<

ENTER RECORD FILLER - ONE CHARACTER FOR ASCII
                      TWO CHARACTERS FOR HEX.            >0 <
```

The "ENTER INPUT MODE" and "ENTER RECORD FILL CHARACTER" option entry fields are protected for source files.

The "FILE TYPE" display is "<01> SOURCE" for source files.

As new records are added to a file being created in the CREATE or MODIFY functions, the utility half closes and re-opens the file after every nth new record entered, where n is the smallest of the three values:

1. 1000/record size (if 0 then 1)
2. ((10/block factor) x block factor) (if 0 then 1)
3. 10

This invokes an update of the "actual number of records in file/End of File pointer" field in the Disk File Header.

In the event of a system failure while a file is being created or modified, no more than the last n new records (1000 bytes of record information or 10 records maximum) entered should be lost, provided no directory or file corruptions have occurred.

Screen 4 – Display

```
FILE IDENTIFIER                                          fffffffffff

NAME OF DISK ON WHICH FILE RESIDES                       ddddddd

CREATION DATE/LAST ACCESS DATE OF FILE                   11 JUL 81/11 JUL 81

RECORD SIZE/BLOCK SIZE OF FILE                           rrrr/bbbbb

NO. OF RECORDS IN FILE                                   nnnnnnn

MAXIMUM NO. OF RECORDS FILE CAN CONTAIN                  mmmmmmm

FILE TYPE                                                <10> CODE

ENTER "1" TO COMMENCE RECORD DISPLAY
   OR "2" TO SELECT RECORD SECTIONS
   OR "3" TO RETURN TO FUNCTION SELECTION                >1<

ENTER "1" FOR ASCII DISPLAY MODE
   OR "2" FOR HEXADECIMAL DISPLAY MODE                   >1<
```

This example is for a code type file.

Any type of file may be selected for display in either ASCII or hexadecimal mode.

Screen 5 – Select Record Sections

```
ENTER "1" TO SELECT RECORD SECTIONS
   OR "2" TO DISPLAY COMPLETE RECORD          >1<

IF "1" IS NOT ENTERED, ALL OTHER INPUT FROM THIS SCREEN WILL BE IGNORED.

UP TO 12 SECTIONS OF A RECORD MAY BE SELECTED FOR    DISPLAY     SECTION
ON A SINGLE SCREEN BY ENTERING IN ORDER, THE START  POSITION      START
OF EACH SECTION IN THE TABLE ON THE RIGHT. THE LAST POSITION    POSITION
MUST BE LESS THAN OR EQUAL TO THE RECORD SIZE OF THE    FILE
                                                                  >    <
WHICH IS xxxx.                                                    >    <
                                                                  >    <
A RECORD SECTION  CONSISTS OF ALL  THE CHARACTERS IN A RECORD     >    <
BETWEEN  THE ENTERED START POSITION    AND   THE NEXT SECTION     >    <
START POSITION OR  THE END OF THE RECORD .  IF THIS NUMBER OF     >    <
CHARACTERS EXCEEDS  yy,   THEN ONLY  yy  CHARACTERS  WILL  BE     >    <
                                                                  >    <
DISPLAYED.                                                        >    <

IN RECORD INPUT/EDIT SCREENS, THESE SELECTED SECTIONS MAY  BE     >    <
INPUT OR EDITED AS IF THE COMPLETE RECORD WAS DISPLAYED.   IF     >    <
THE CURRENT RECORD IS NEW, ALL OTHER SECTIONS WILL BE  FILLED     >    <
WITH  THE APPROPRIATE FILL CHARACTER   OTHERWISE  THEY   WILL     >    <
                                                                  >    <
REMAIN UNALTERED.
```

yy – 25 for Hexadecimal character input
50 for ASCII character input

This option, which may be selected from the Record Input/Edit, Record Copy or Record Display function, allows the user to select up to 13 specific sections of a record to be displayed on one screen and is self explanatory. Complete Record display is described in the next section.

If invalid start positions are entered, this screen is re-displayed with a message identifying the error.

If creation or modification of a file (file-a) is completed and a new file is created or an existing file is selected for modification or display (file-b); provided the input type and record sizes of file-a and file-b are the same, record sections for file-b default to those last set for file-a if the select record section screen is not requested. Otherwise, the complete record display mode is the default.

Screen 6 – Record Input/Edit

```
FILE fffffffffff              MAX. POSSIBLE SIZE OF FILE IS mmmmmmm RECORDS
DISK ddddddd                    ACTUAL SIZE OF FILE IS nnnnnnn RECORDS
ENTER "1" TO INPUT,EDIT OR DISPLAY CURRENT RECORD
   OR "2" TO COPY RECORDS FROM ANOTHER FILE
   OR "3" TO RE-SELECT RECORD SECTIONS
   OR "4" IF WORK ON THIS FILE IS COMPLETED      >1<
   OR "5" TO SPECIFY KEYS FOR FIND COMMAND
IF "1" IS NOT ENTERED, ALL OTHER INPUT FROM THIS SCREEN WILL BE IGNORED.

CURRENT RECORD NUMBER IS xxxxxxx                              >rec.check<
   1  >0000000000000000000000000000000000000000000000000000<
  33  >0000000000000000000000000000000000000000000000000000<
  63  >00000000000000000<
  71  >0000000000000000000000000000000000000000000000000000<
 103  >00000000000000000<
 111  >00000000000000000<
 119  >0000<
 121  >0000<
 123  >000000<
 126  >00000000<
 130  >0000000000<
 135  >00000000000000000000000000000000<

ENTER NEXT RECORD TO BE DISPLAYED OR "FIND"  >yyyyyyy<
```

The >rec.check< field is protected. When this screen is transmitted back to the program, the content of the field is verified as unchanged. If not, the screen for the current record or part(s) of this record displays the message, "INVALID INFORMATION RECEIVED – PLEASE RE-ENTER".

Since record sizes up to 2000 bytes are permitted and it is impossible to display a complete record on a single screen format, it may be necessary for the utility to display a record on more than one successive screen, but on as few as possible, depending on the input type (ASCII or Hex.) and record size. This is required for all options of the utility which display record information – Record Input/Edit, Record Display and Record Copy.

For ASCII input, a maximum of 500 bytes is displayed per screen, 50 bytes by 10 lines; up to four screens may be necessary to display a complete record.

For Hex. Input, a maximum of 250 bytes is displayed per screen, 25 bytes by 10 lines; up to eight screens may be necessary to display a complete record.

These display formats of 500 bytes/ASCII and 250 bytes/Hexadecimal per screen are fixed.

When more than one screen is required to display a record, the "NEXT RECORD TO BE DISPLAYED" field contains the record number of the current record in all screens except the last, where it is incremented to the current record number + 1.

The user may select another option or alter the above field before all the screens of the current record have been displayed. If a new record is being displayed for input by the Record Input/Edit option, those parts of the record described by screens which were not displayed are filled with appropriate fill characters. For an old record, the un-displayed/un-edited parts of the record remain unaltered.

This method of displaying a complete record is the same for screens 7 and 8, Record Copy and Record Display.

If the current record number is the next new record, the record entry field(s) are filled with the pre-defined or default fill character.

If an already written record is selected, that record is displayed in the record entry field(s) and may be edited as required.

A "NEXT RECORD TO BE DISPLAYED" field entry which is greater than the current size of the file + 1 is invalid.

"FIND" may be entered if keys have been specified for the find command. If requested, the utility will search from the current record + 1. If a record is found containing the specified keys, that record will be displayed, if not, the previous current record will be re-displayed.

Cursor management: If this screen is transmitted with the cursor sitting in a record data entry field, all the data up to the cursor is saved. Any modifications done after the cursor in entry fields following the cursor (including the field the cursor is sitting in) are not saved; the data as originally displayed is saved.

The example for screen six shows record entry fields for a data file with record sections selected at 1, 33, 63, 71, 103, 111, 119, 121, 123, 126, 130, 135 and 140; record size is 150, input mode is hexadecimal, Hex. 00 is the fill character and the screen is waiting for information for new record number xxxxxxxx to be entered.

Screen 7 – Copy Records From Another File

```
ENTER "1" TO DISPLAY OR COPY RECORDS
   OR "2" TO RE-SELECT RECORD SECTIONS              >1<
   OR "3" IF NO OTHER FILES ARE REQUIRED
ENTER IDENTIFIER OF FILE CONTAINING THE REQUIRED RECORDS  >           <
ENTER NAME OF DISK ON WHICH THIS FILE RESIDES              >sssssss<
ENTER REQUIRED RECORD NUMBER, LAST IN FILE IS |||||||     >1        <
   1     aaaaaaaaaaaaaaaaaaa<
  21     aaaaaaaaaa<
  31     aaaaaaaaaaaaaaaaaaaaaaaa<
  56     aaaaaaaaaaaaaaaaaaaaaaaa<
  81     aaaaaaaaaaaaaaaaa<
  99     aa<



TO COPY THIS REC. OR RANGE FROM THIS REC, ENTER END REC. NO.>      <
FILE fffffffffff    MAX. POSSIBLE SIZE IS ttttttt RECORDS
DISK ddddddd          SIZE OF FILE NOW IS ttttttt RECORDS
ENTER STARTING NUMBER FOR THESE RECORD<S> TO BE COPIED TO   >      <
```

In this example, record sections selected at 1, 21, 31, 56, 81 and 99 are for a source file or data file in ASCII mode with a record size of 100 bytes.

The secondary file (from which records are displayed or copied) must be of the same type and have the same Record Size as the primary file (being created or updated).

Record sections are the same as for the primary file.

The "End of Range record" entry field and the "Starting record in the primary file" entry field are always blank by default. This allows records in the secondary file to be listed without any record copying being performed.

To copy records, the user must enter valid record numbers in both the above fields.

NOTE

To copy only one record, the same record number as that requested for display must be entered in the "End of Range record" field.

Secondary File record display fields are write protected in this screen.

Until a valid file-name has been entered and transmitted to the program and that file has been successfully opened, no record information is displayed. Also, the "LAST IN FILE IS" display, which refers to the last record in the Selected Secondary file, has "....." in the record number field until that file has been successfully opened.

Screen 8 – Record Display

```
FILE fffffffffff          MAX. POSSIBLE SIZE OF FILE IS mmmmmmm RECORDS
DISK ddddddd                    ACTUAL SIZE OF FILE IS nnnnnnn RECORDS

ENTER "1" TO DISPLAY RECORD
   OR "2" TO RE-SELECT RECORD SECTIONS
   OR "3" IF NO MORE RECORDS ARE TO BE DISPLAYED  >1<
   OR "4" TO SPECIFY KEYS FOR THE FIND COMMAND
IF "1" IS NOT ENTERED, NEXT RECORD NUMBER WILL BE IGNORED.

CURRENT RECORD NUMBER IS xxxxxxx
   1  >hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh<
  21  >hhhhhhhhhh<
  26  >hh<
  27  >hhhhhhhhhh<
  32  >hh<
  33  >hhhhhhhhhh<
  38  >hhhhhh<
  41  >hh<
  42  >hh<
  43  >hh<
  44  >hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh<
  60  >hh<

ENTER NEXT NUMBER TO BE DISPLAYED OR "FIND"  >yyyyyyy<
```

In this example, record sections have been selected for a file with a record size of 80 and the display mode is in hexadecimal.

Record information display fields are write protected in this screen.

Any type of file is permitted in the display function including "SYSMEM".

Screen 9 – Specify Keys

```
ENTER "1" TO SPECIFY KEYS FOR FIND COMMAND
   OR "2" TO SET FIND COMMAND FOR X-FILLED RECORDS >1<
   OR "3" IF FIND COMMAND IS NOT REQUIRED
IF "2" IS ENTERED, KEY SPECIFICATION ON THIS SCREEN WILL BE IGNORED.
LENGTH   KEY VALUE                                          TEST  POS.N
 >  <   >                                                  < >  < >    <
 >  <   >                                                  < >  < >    <
 >  <   >                                                  < >  < >    <

UP TO THREE KEYS MAY BE SPECIFIED.

LENGTH      < ENTER EXACT LENGTH OF KEY.
              MAXIMUM FOR THIS FILE IS xx IN yyyy MODE.
KEY VALUE   < ENTER VALUE OR STRING TO BE FOUND
TEST        < ENTER "= ", "<=", "< ", ">=" OR "> "
POS.N       < ENTER POSITION IN RECORD ONLY IF KEY IS TO
              BE TESTED AT A SPECIFIC LOCATION.
              IF BLANK, THE WHOLE RECORD WILL BE SEARCHED.

A RECORD WILL BE FOUND ONLY IF ALL SPECIFIED KEY TESTS ARE SUCCESSFUL.
THE FIND COMMAND CAN BE REQUESTED ON THE RECORD DISPLAY SCREENS.
IF REQUIRED, A SEARCH CAN BE INITIATED NOW ON FILE fffffffffffff
ON DISK ddddddd BY ENTERING THE STARTING RECORD NUMBER.    >yyyyyyy<
```

Specification of keys for the FIND command may be requested from the record input/edit and record display screens.

Key values should be entered in ASCII or Hex. depending on whether the file is being edited or displayed in ASCII or Hex. mode.

Once keys have been specified, or the FIND command has been set to search for "X" filled records, that setting will remain unchanged until a file of different record size is selected, the intput/edit/display mode is altered (ASCII/Hex), or the option to specify keys is requested again.

The maximum length of key value which can be specified is 28 if in hex. mode, 56 if in ASCII mode, or the record size of the file, whichever is smaller. This appears at field xx on the screen. "ASCII" or "HEX" will appear in field yyyyy.

If the starting record number field is left blank, the utility will re-display the last record displayed. The "FIND" command can be requested from the record input/edit screens when required.

If a starting record number is entered, the FIND command will commence searching the file from this record number (see screen 10).

Screen 10 – Find Information

```
THE UTILITY IS SEARCHING

FILE fffffffffff
ON DISK ddddddd

FOR A RECORD CONTAINING THE xxxxx KEY<S>

LENGTH    KEY VALUE                                        TEST POSITION

  (keys as specified)

or
FOR A RECORD FILLED WITH "X" ASCII CHARACTERS <HEX. 58>
  (if in Hex. mode)

or
FOR A RECORD FILLED WITH "X" ASCII CHARACTERS
  (if in ASCII mode)

FIND COMMAND MAY BE TERMINATED BY PRESSING XMIT KEY ONCE.
```

This screen, which is for information only, is displayed when the FIND command has been requested from the record input/edit display screens, or initiated from the specify keys screen.

The command will search from the appropriate record for a record containing the specified keys or "X" filled records. When the end of file is reached, the command will search from the start of the file, but only as far as the record from which the search was initiated.

If a record is found, that record will be displayed with a message identifying the position of the specified keys within the record.

If the FIND command is terminated or is unsuccessful, the screen from which the command was initiated will be re-displayed.

"ASCII" or "HEX" will appear in the xxxxx field.

A message will be displayed on the bottom line of the screen which will inform the user of the progress of the search. See paragraphs on Screen Messages.

Screen 11 – Help 1

THIS PROGRAM CAN BE USED TO CREATE, UPDATE AND REMOVE SOURCE OR DATA TYPE FILES. ANY TYPE OF FILE CAN BE DISPLAYED.

THE MAXIMUM RECORD SIZE WHICH CAN BE HANDLED IS 2000 CHARACTERS.

IN THE CREATE AND UPDATE FUNCTIONS, NEW RECORDS MAY BE INPUT INTO THE FILE AND OLD RECORDS MAY BE EDITED USING THE RECORD INPUT/EDIT SCREEN BY ENTERING OR ALTERING THE RECORD INFORMATION IN THE RECORD ENTRY FIELDS. RECORDS MAY BE SELECTED BY ENTERING THE REQUIRED RECORD NUMBER IN THE "NEXT RECORD NUMBER TO BE DISPLAYED" FIELD.

RECORDS MAY BE COPIED FROM ANOTHER FILE BY SELECTING THE COPY OPTION FROM THE RECORD INPUT/EDIT SCREEN. RECORDS CAN BE DISPLAYED AND COPIED FROM ANY SELECTED FILE WHICH HAS THE SAME RECORD SIZE AND IS OF THE SAME TYPE AS THE FILE BEING CREATED OR UPDATED. COPIED RECORDS MAY BE ADDED TO THE END OF THE FILE OR MAY OVER-WRITE EXISTING RECORDS.

TO DELETE RECORDS FROM AN EXISTING FILE, A NEW VERSION OF THE FILE MUST BE CREATED USING THE CREATE FUNCTION. THE REQUIRED RECORDS CAN THEN BE COPIED TO THE NEW VERSION OF THE FILE USING THE COPY OPTION.

THE DISPLAY FUNCTION ONLY ALLOWS RECORDS OF A FILE TO BE DISPLAYED.

PRESS "XMT" TO CONTINUE

Screen 12 – Help 2

"RECORD FILLER" FIELD - WHEN CREATING OR UPDATING A DATA TYPE FILE, THE USER MAY ENTER A CHARACTER WHICH WILL FILL RECORD ENTRY FIELDS IN RECORD INPUT/EDIT SCREEN FOR NEW RECORDS.
IF A SINGLE CHARACTER IS ENTERED WITH A LEADING OR TRAILING BLANK, THAT SINGLE ASCII CHARACTER WILL BE USED AS THE FILL CHARACTER.
IF TWO CHARACTERS ARE ENTERED, THEY WILL BE REGARDED AS THE HEXADECIMAL CODE FOR A SINGLE CHARACTER AND SHOULD BE IN THE RANGE 0-9/A-F.
IF IT IS NOT IN THE RANGE HEXADECIMAL 20-5D-60-7E XXXXXXX, THE FILL CHARACTER WILL APPEAR AS A ? WHEN DISPLAYED IN RECORD ENTRY FIELDS IN THE RECORD INPUT/EDIT SCREEN IF INPUT MODE IS ASCII.

NON-DISPLAYABLE CHARACTERS IN RECORDS OF DATA FILES (ASCII MODE).
WHEN USING ANY FUNCTION ON A DATA FILE IN ASCII MODE, ANY CHARACTER IN A RECORD TO BE DISPLAYED WHICH IS NOT WITHIN THE RANGE HEX. 20-5D/60-7F WILL APPEAR AS A ?. IN THE CREATE AND UPDATE FUNCTIONS, ? CHARACTERS MAY BE ALTERED IN THE SAME WAY AS DISPLAYABLE CHARACTERS IF THEY APPEAR IN THE RECORD ENTRY FIELDS IN THE RECORD INPUT/EDIT SCREEN.
IF THE INPUT/EDIT SCREEN IS TRANSMITTED WITH ? CHARACTERS IN THE RECORD ENTRY FIELDS, EACH CHARACTER CORRESPONDING TO A ? WILL REMAIN UNALTERED IN THE ACTUAL RECORD.

IN SOURCE TYPE FILES, THE ? WILL BE TREATED AS A NORMAL CHARACTER.
NOTE. UNDERSCORE, UP-ARROW AND DEL CHARACTERS ARE CONVERTED TO A ?

PRESS "XMT" TO CONTINUE

Screen 13 – Help 3.

```
FOR THIS PROGRAM TO RUN EFFICIENTLY, IT IS VITAL THAT THE USER WAITS FOR
A COMPLETE SCREEN TO BE DISPLAYED ON THE TERMINAL BEFORE ENTERING AND/OR
TRANSMITTING INFORMATION BACK TO THE PROGRAM. TO TRANSMIT, PRESS THE
"XMT" KEY ONCE ONLY.

IF AN INVALID REQUEST IS ENTERED IN AN ENTRY FIELD ON A SCREEN,THAT SAME
SCREEN WILL BE RE-DISPLAYED BY THIS PROGRAM. THE FIRST INVALID ENTRY
WILL BE HIGHLIGHTED AND A MESSAGE IDENTIFYING THE ERROR WILL BE DISPLAY-
ED ON THE LAST LINE OF THE SCREEN. THE INVALID ENTRY SHOULD BE CORRECTED

IF THE INFORMATION DISPLAYED BY THE PROGRAM BECOMES OVER-WRITTEN OR
ERASED, THE USER CAN REFRESH THE SCREEN BY DOING THE FOLLOWING.

 PRESS "HOME" KEY,   ENTER "0",    PRESS "XMT" KEY.

WHEN THE SCREEN IS RE-DISPLAYED, THE ENTRY FIELDS WILL BE FILLED WITH
THE INFORMATION WHICH WAS ORIGINALLY DISPLAYED BY THE PROGRAM ON THIS
SCREEN. IF THE REFRESH OPTION IS REQUESTED WHILE USING THE RECORD INPUT/
EDIT SCREEN, THE ACTUAL RECORD INFORMATION WILL RE-APPEAR IN THE RECORD
ENTRY FIELDS.


 PRESS "XMT" TO RETURN TO FUNCTION SELECTION
```

If the HELP function is selected in screen 1, the utility displays screens 11, 12 and 13 and then re-displays screen 1.

The characters UNDERSCORE/£5E and UP-ARROW/£5F are converted because it is possible for the TD 830 type terminals to be configured to interpret these characters as unprotected data entry and terminate forms delimiters when they are received in forms mode.

Screen Messages

If invalid input is entered in the entry fields on any screen or any error condition is detected, the same screen is re-displayed with one of the following messages displayed in the bottom line. All data as originally transmitted is re-displayed.

Field Entry Error Messages May be displayed on screens:

1. ERROR – HIGHLIGHTED ENTRY MUST NOT BE SPLIT	1-9
2. ERROR – HIGHLIGHTED ENTRY MUST NOT BE BLANK	1-9
3. ERROR – HIGHLIGHTED ENTRY MUST BE NUMERIC	1-9
4. ERROR – HIGHLIGHTED ENTRY SHOULD NOT BE GREATER THAN xxx	1-9
5. ERROR – HIGHLIGHTED ENTRY SHOULD NOT BE LESS THAN xxx	7
6. ERROR – HIGHLIGHTED ENTRY MUST NOT BE ZERO	1-9
7. ERROR – BLOCK SIZE MUST BE A MULTIPLE OF RECORD SIZE	2
8. ERROR – INVALID CHARACTER IN IDENTIFIER	1,7

(continued)

Field Entry Error Messages May be displayed on screens:

9. ERROR – START POSITIONS MUST BE ENTERED IN ORDER	5
10. ERROR – NO START POSITIONS HAVE BEEN ENTERED	5
11. ERROR – BLANKED OUT CHARACTERS WERE NOT HEXADECIMAL	6
12. ERROR – INVALID INFORMATION RECEIVED – PLEASE RE-ENTER	6
13. ERROR – HIGHLIGHTED ENTRY MUST BE NUMERIC OR "FIND"	6, 8
14. ERROR – NO KEYS HAVE BEEN SPECIFIED FOR FIND COMMAND	6, 8, 9
15. ERROR – TEST MUST BE "=", "<", ">", "<=" or "=>"	9

Message 11 may only be displayed in the Record Input/Edit function screen (6) if the input mode is hexadecimal and characters not in the range 0-9/A-F have been entered in record data entry fields. These non-hexadecimal characters are replaced by blanks when the screen is re-displayed.

Message 12 will be displayed if the <rec.check> field does not contain the expected string when transmitted back to the program. This may happen if the user depresses the "XMT" key more than once when transmitting.

File Error Messages

The following messages may be displayed on screen 1 (SELECT function) and screen 7 (COPY records) only.

Message May be displayed on screens:

1. CANNOT FIND FILE fffffffffff ON DISK ddddddd	1,7
2. FILE fffffffffff ON DISK ddddddd IS INCOMPLETE	1
3. FILE fffffffffff ON DISK ddddddd IS IN USE	1,7
4. DISK ddddddd IS NOT AVAILABLE	1,7
5. DISK ddddddd IS LOCKED	1,7
6. FILE fffffffffff ALREADY EXISTS ON DISK ddddddd	1
7. FILE fffffffffff ON DISK ddddddd IS NOT SOURCE OR DATA TYPE	1
8. FILE fffffffffff ON DISK ddddddd IS NOT SOURCE TYPE	7
9. FILE fffffffffff ON DISK ddddddd IS NOT DATA TYPE	7
10. RECORD SIZE OF FILE fffffffffff ON DISK ddddddd IS NOT rrr	7
11. FILE fffffffffff ON DISK ddddddd HAS NO RECORDS	1,7
12. RECORD SIZE OF FILE fffffffffff ON DISK ddddddd IS GREATER THAN 2000	1
13. DIRECTORY OF DISK ddddddd IS FULL	1

Remove Function Message

1. FILE fffffffffff REMOVED FROM DISK ddddddd

This message is displayed when a file has been successfully removed from a disk.

Copy Messages

1. >>>>>RECORD<S> BEING COPIED – PLEASE WAIT.....

This message flashes until copying of records is complete; the following message is then displayed.

2. xxxxxxx RECORDS COPIED TO FILE fffffffffff ON ddddddd STARTING AT REC. yyyyyyyy

Copy Error Messages

1. NO RECORDS COPIED-BAD RECORD xxxxxxx IN FILE fffffffffff ON ddddddd
2. ONLY xxxxxxx RECORD<S> COPIED-BAD RECORD yyyyyyy IN FILE fffffffffff ON ddddddd
3. RECORD xxxxxxx IN FILE fffffffffff ON ddddddd IS BAD

A read error has been detected on the Secondary file. Message 3 may also be displayed on Screen 8 (record display).

4. NO RECORDS COPIED-MAX. SIZE OF FILE fffffffffff ON ddddddd CANNOT BE EXCEEDED

Find Command Messages

1. >>>>> SEARCHING FROM RECORD xxxxxxx – PLEASE WAIT.....
2. >>>>> END OF FILE REACHED – NOW SEARCHING FROM START – PLEASE WAIT
3. RECORD FOUND
4. NO RECORD FOUND
5. FIND COMMAND TERMINATED

Message 5 will be displayed if the user has pressed XMT key to terminate search.

The above messages may be displayed on the FIND information screen (10).

The following messages may be displayed on the record input/edit (6) and the record display (8) screens when the FIND command has been terminated, has found a record, or has failed to find a record.

1. FIND COMMAND TERMINATED AT RECORD xxxxxxx:
2. THIS RECORD CONTAINS THE SPECIFIED KEY(S) AT POSITION(S) a b c
3. NO RECORD FOUND IN FILE WITH SPECIFIED KEY(S)
4. THIS RECORD IS FILLED WITH "X" CHARACTERS
5. NO "X" FILLED RECORD FOUND IN FILE

No Disk Space Message

1. >>>>> WAITING FOR SPACE FOR FILE fffffffffff ON DISK ddddddd.....

This message may appear on screen 4 (record Input/Edit) or screen 7 (Record Copy) if an area for the file being created or modified cannot be allocated. Operator intervention is required to resolve the situation. This message continues to flash until space is made available for the file (or another disk is AD'ed to the utility).

Input Or Output Parity On A File Being Created Or Modified

If an error is detected by the utility when inputting or editing a record via screen 6 (Record Input/Edit) in the CREATE or UPDATE functions, or when copying records from a file using the COPY function, the utility terminates the function and displays the following screen.

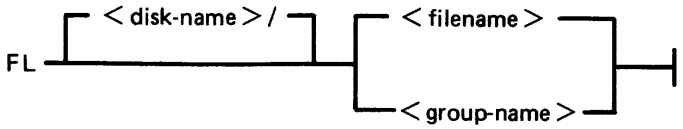
```
PERMANENT ERROR DETECTED ON FILE fffffffffff ON DISK ddddddd.

RECORD xxxxxxx IS BAD.


RECOVERY OF THIS FILE MAY BE POSSIBLE.

CREATE A NEW VERSION OF THE FILE BY USING THE CREATE FUNCTION.  COPY
GOOD SECTIONS OF THE OLD FILE TO THE SAME LOGICAL RECORD POSITION IN
THE NEW FILE USING THE COPY OPTION.

EACH BAD RECORD IN THE OLD FILE WILL HAVE TO BE REPLACED IN THE  NEW
FILE BY INPUTTING A NEW RECORD WITH THE SAME RECORD NUMBER  INTO  THE
NEW FILE USING THE RECORD INPUT/EDIT SCREEN.

THE NEW FILE SHOULD BE BUILT SEQUENTIALLY - A REPLACEMENT FOR A  BAD
RECORD SHOULD BE INPUT INTO THE NEW FILE BEFORE  A  SUBSEQUENT  GOOD
SECTION OF THE OLD FILE IS COPIED TO THE NEW FILE.

THE NEXT SCREEN TO BE DISPLAYED IS FOR FUNCTION SELECTION.
```

The utility then displays Screen 1.

FL (Display File Attributes on Self-Scan – B 90 Only)

This utility allows the operator to display detailed information about particular files or groups of files on disk. The information given is similar to the LR utility, and is displayed on a console screen, or console printer if no console screen is available. This utility supports the starfile option *<file-name> in the initiating message.

Format:



The utility uses the following PKs when more than one file is specified:

PK1	PK6
NEXT	END

Examples:

To display information about all entries on the system disk:

```
FL =
```

To display information about a file called PR200 found on disk called PR2:

```
FL PR2/PR200
```

To display information about a group of files beginning with the letters "PR" found on the system disk:

```
FL PR =
```

Output Format:

Information from this utility can be output in one of two forms depending on the size of screen being used. For a CRT configured to be 40 or more characters wide, or for a console printer, the six-line format is as follows:

```
Line 1 FILE ddddddd/fffffffffff filetype @ft@
Line 2 ACTUAL SIZE xxxxxxx : RECORD SIZE xxxxx
Line 3 MAXIMUM SIZE xxxxxxx : RECS/BLOCK xxxxx
Line 4 DATE CREATED yyddd : ACCESSED yyddd
Line 5 AREA MAP : **********
Line 6 OVERFLOW ON DISK : ddddddd
```

For a CRT configured to 32 characters wide or for a SELF-SCAN, the format is as follows:

```
Line 1 FILE NAME: ddddddd/fffffffffff
Line 2 FILE TYPE: filetype @ft@
Line 3 SIZE: ACTUAL xxxxxxx, MAX xxxxxxx
Line 4 CREATED ACCESSED REC.SZ RECS/BLK
```

Line 5 yyddd yyddd xxxxx xxxxx
Line 6 AREA MAP: **********
Line 7 OVERFLOW ON DISK: ddddddd PSEUDO)
Line 8

Note that the AREA MAP entries will show * for unallocated areas, B for areas allocated on this disk and O for areas allocated on the overflow disk.

If the file is dual-pack and the overflow disk is not available, the last line of each format will include "FILE INCOMPLETE"

Note that the OVERFLOW ON DISK will not be displayed if the file has no overflow areas allocated.

The first line contains the disk name specified by DDDDDDD on which the file specified by FFFFFFFFFFF resides.

The filetype entry will contain one of the following:

Filetype		Entry in Listing/Display
@00@	DATA	@00@
@01@ – @0E@	SRCLANG	@0x@
@0F@	SRCLIBR	@0F@
@10@ – @13@	CYYMMDD	@1x@
@21@	SYSLANG	@21@
@22@	SYSDATA	@22@
@30@	VIRTMEM	@30@
@31@	SYSLOG	@31@
@81@	KEY	@81@
@A0@	PRNTBKP	@A0@
Any other	SYSTEM	@xy@

ACTUAL SIZE	The actual file size specified for the file.
MAXIMUM SIZE	The maximum file size specified for the file.
RECORD SIZE	Number of characters per second.
RECS/BLOCK	Number of records per block.
DATE CREATED	Date of creation.
ACCESSED	Date the file was last accessed.
AREA MAP	16 characters to show the allocation of the 16 areas into which a file may be broken.

Entries for AREA MAP are as follows:

Entries	Area Allocated On:	Possible Disk Types
*	Unallocated	1 2 or 3
B	Base disk	1
O	Overflow disk	1 or 2
U	Physical unit	2 or 3
A	Another physical unit	3
	Disk Types:	

Entries	(continued) Area Allocated On:	Possible Disk Types
	1 = non-pseudo disk 2 = restricted pseudo disk 3 = unrestricted pseudo disk	

The two component disks of a dual-pack file can be:

Two non-pseudo disks
or
One non-pseudo disk and one restricted pseudo disk.

"UNIT : uuuuuuu"	is only displayed if ddddddd is a pseudo disk; uuuuuuu being the physical unit on which this file directory information was found.
"UNRESTRICTED PSEUDO"	is displayed only if ddddddd is an unrestricted pseudo disk.
"FILE INCOMPLETE"	is only displayed if the file is a dual-pack or an unrestricted pseudo disk file, and one or more of the disk or units on which areas and/or directory entries for this file are not available.
"OVERFLOW DISK : ooooooo"	is only displayed if the file is a dual-pack.

Output Messages

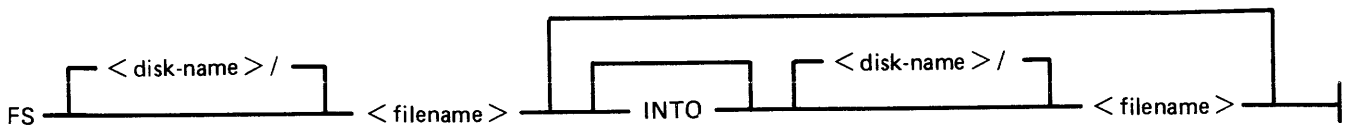
Message	Possible Cause	Suggested Action
<file name> NOT LISTED	The utility could not list the specified file. The reason is given in the second message.	See the second error message given for the reason.
<family name> NOT LISTED	The utility could not list the specified file. The reason is given in the second message.	See the second error message displayed for the reason.

Note: Refer to "Common Utility Output Messages" for additional aid.

FS (File Squash)

This utility allows the operator to remove all deleted records from a data file. Records are normally "deleted" (that is, hexadecimal @FF@ is written over the records) through an appropriate application program. The FS utility will remove these previously deleted records, allowing additional records to be added to the file.

Format:



The "file-name" identifies either a data file or a keyfile. If a keyfile has been specified, the name of the data file is obtained from the information held in the keyfile.

If a keyfile is specified, then the utility will reconstruct this keyfile so that it relates to the modified data file.

If an index file is specified for squashing, the utility will display the message:

<keyfile-name> IS ASSOCIATED WITH <datafile-name>

If an INPLACE squash is being done, the keyfile will be removed and the message:

<keyfile-name> REMOVED

is displayed.

While the utility is processing, no other program may access the data file (or the keyfile if one is specified).

If only one file-name is specified, and no other options are used, the file squash will be carried out in place, and no new data file will be created. If a keyfile was specified, then a new keyfile with the same "file-name" will be recreated by the SORT.

If two file-names are specified, the data file will be squashed into a new file, and the keyfile (if specified) will be recreated by the SORT. If the first file-name specifies a keyfile, then the new keyfile will have the name indicated by second file-name, and the new data file will have the name of the new keyfile name, with the letters "QQ" attached to the end of the name.

Examples:

To squash the file, PR200:

```
FS PR200
```

To squash the file, PR200 and create a new file, PR200B:

```
FS PR200 INTO PR200B
```

Output Messages

Message	Possible Cause	Suggested Action
UNABLE TO ZIP SORT	SORT utility is not initiated. The reason is given by the second message displayed by the utility.	See the second error message displayed for the reason.
-RECORD SIZE LIMIT IS 1000	An attempt was made to squash a file with a record size greater than 1000 bytes.	None. Utility limitation.
<file name> SQUASHED FROM n RECORDS TO m RECORDS	FS successful.	None.
KEYFILE <file name> RECONSTRUCTED	FS successful. The data file associated with specified keyfile has squashed successfully and SORT reconstructed the keyfile.	None.
-HARDWARE ERROR DETECTED WHILE READING KEYFILE <file name>	The utility has encountered an error while reading the specified keyfile.	Run FS on the datafile then SORT the datafile to obtain a new keyfile.
-KEYFILE <filename> WAS NOT CREATED FOR DATAFILE <filename>	SORT was not successful.	Refer to the output messages of SORT to detect the reason for the failure.
-NO RECORDS IN KEYFILE <filename>	The keyfile to be FS'ed is empty.	Run SORT on the related datafile to obtain a new keyfile.
-<filename> IS NOT A DATAFILE	The specified datafile is not of type data or source.	Check input for correct file name.
UNABLE TO ACCESS <filename>	The specified file is not on the specified disk.	Make file available and re-enter.
-UNABLE TO ACCESS ASSOCIATED DATAFILE <filename>	The datafile with the filename found in the keyfile is not available to the system.	Make file available and re-enter.

Note: Refer to "Common Utility Ouput Messages" for additional aid.

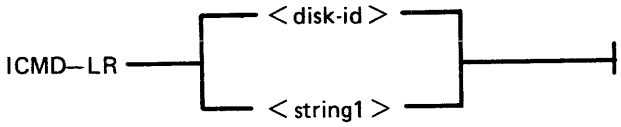
NOTE

It is strongly recommended that FS, once loaded, is not DSed.

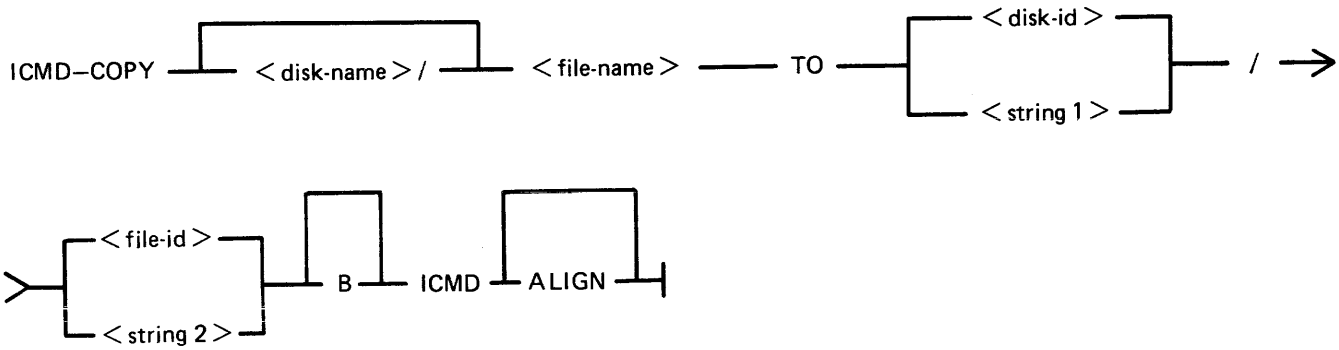
ICMD

The ICMD utility allows the operator to access industry compatible mini disks.

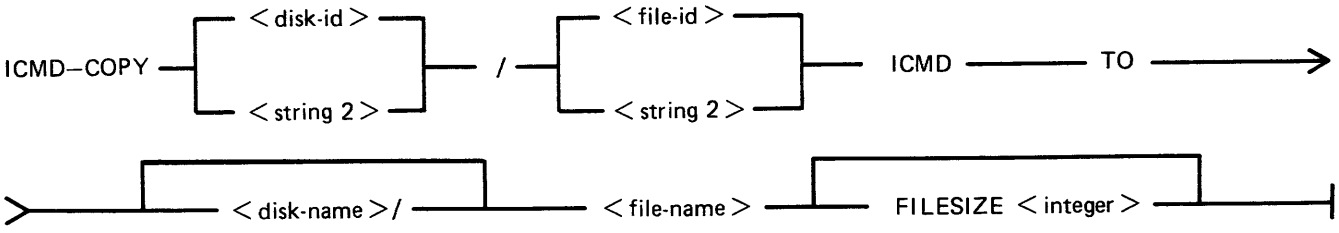
Version 1



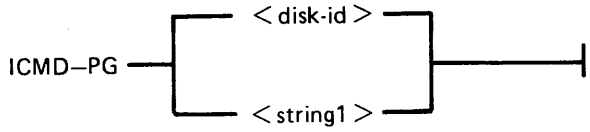
Version 2



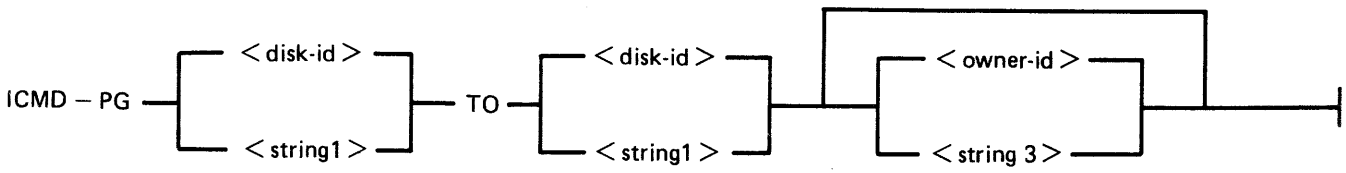
Version 3



Version 4



Version 5



<disk-id>	Is the volume identifier of an Industry Compatible Mini Disk (ICMD) and does not exceed six valid CMS characters.
<file-id>	Is the name of a file on an ICMD and does not exceed eight valid CMS characters.
<owner-id>	Is an identifier not exceeding 14 valid CMS characters.
<string1>	Is a sequence of characters bounded by quotes, not exceeding six characters. Alternative to <disk-id>.
<string2>	Is a sequence of characters bounded by quotes, not exceeding eight characters. Alternative to <file-id>.
<string3>	Is a sequence of characters bounded by quotes, not exceeding 14 characters. Alternative to <owner-id>.
<disk-name>	Is a CMS standard disk name.
<file-name>	Is a CMS standard file name.

NOTE

A quote may be included in the string by substituting a pair of quotes.

The first version allows the operator to list the directory of an ICMD. The utility will print a line of information for each file on the disk.

The second version allows the operator to copy a CMS file to an ICMD file. If B is specified, any file may be copied without loss of attributes but it may only be copied back by this utility. The file may be accessed by other systems but it must be a data or source file with a record size not exceeding 128. If the keyword ALIGN is specified, the file will be copied to the ICMD so that the start of the file is at the beginning of a track.

The third version allows the operator to copy an ICMD file to a CMS file. Files originally copied by this utility using the B option will be restored as they were. Other files will be restored as data files with the largest possible block not exceeding 180. If the FILESIZE of the CMS file is not specified, it will be calculated from the header of the ICMD file.

NOTE

When copying a Multi-volume file which was generated without the B option, the utility will insist on this attribute.

The fourth version allows the operator to purge all files on an ICMD. The utility will replace all files by a single zero-length file named DATA to which all the disk space available to the user is allocated.

The fifth version is similar to the fourth but re-labels the ICMD. The new label will record whatever owner-id is specified.

Examples

To print the disk directory of an ICMD disk called PR2:
ICMD LR PR2

To purge all the files from the ICMD called PR2 and name it as PR3:
ICMD PG PR2 TO PR3

To copy a file called PR200 from the CMS disk called PR1 to an ICMD disk called PRBU:
ICMD COPY PRI/PR200 TO PRBU/PR200 ICMD

To copy a file called PRFILE from the ICMD called PRBU to the CMS disk called PRI:
ICMD COPY PRBU/PRFILE ICMD TO PRI/PR200

To copy a file called !! from the ICMD called FRED* to the file called CMSFILE on the CMS disk called CMSDISK:
ICMD COPY "FRED*"/"!!" ICMD TO CMSDISK/CMSFILE

Output Messages

Message	Possible Causes	Suggested Action
<filename> NOT COPIED	ICMD could not copy the specified file. The reason is given in the second message.	Look for the reason in the second error message displayed.
INVALID CHARACTER IN IDENTIFIER <string> – WARNING	The specified string does not comply with ICMD standards.	Check input for correct characters used in identifier.
– <filename> IS ALREADY ON DISK	The specified file cannot be copied to ICMD as a file of the same name already exists.	Copy this file to another name.
– PARAMETER FOLLOWING KEYWORD <string> IS INVALID	There is an error in the initiating message.	Correct input and re-enter.
<filename> REMOVED	A CMS file was removed by the utility to make way for another file being copied from ICMD.	None.
– HARDWARE ERROR DETECTED WHILE WRITING TO FILE <filename>	The utility has detected an error while writing to the specified file.	Check integrity of output medium.
<filename> COPIED TO <filename>	ICMD COPY successful.	None.
– NO SPACE ON DISK	Self-explanatory.	Select a disk with enough unused space to copy to.

(continued)

Message	Possible Causes	Suggested Action
-RECORD SIZE EXCEEDS 128	The file cannot be copied to the ICMD as its record size is greater than 128.	Use CMS COPY utility to change record/block sizes. (For example, COPY CMSFILE TO CMSFILE RECORD 128 BLOCK 128) Then use ICMD COPY to complete transfer.
-BAD INDEX TRACK ON <diskname> SECTOR <number>	ICMD has an error on track 0 and cannot be used by this utility.	Select another disk for this function.
VOLUME n OF <filename> OUT OF SEQUENCE	The utility expected another disk as the next disk in a multivolume file copy.	Remove inappropriate disk and supply the disk with the correct sequence number.
NEXT SEQUENCE NUMBER OF <filename> EXCEEDS 99	No more disks may be used, as the maximum for an ICMD is 99.	Use CMS COPY to break file. Then re-run ICMD.
ENTER DISKNAME FOR <filename> n	The utility requests the multi-volume filename and the sequence number to be used.	Enter appropriate diskname and sequence number.
FILESIZE IS REQUIRED FOR <filename>	The attribute FILESIZE must be specified when copying a multi-volume ICMD file if it was created without the "B" option.	Re-enter with filesize option in initiating message.
<diskname> PURGED	Successful completion of an ICMD PG function.	None.
"DATA" IS A RESERVED FILENAME	An attempt was made to create a file with the filename "DATA", which is a reserved word.	Use another name and re-run the function.
-SPECIFIED FILESIZE IS TOO SMALL	The filesize specified in the FILESIZE attribute is too small.	Rerun the function with a larger filesize.
NO FREE LABEL ON <disk-name>	File cannot be copied to ICMD as its directory is full.	Select another disk which has directory space.

Refer to Common Utility Output Messages for additional aid.

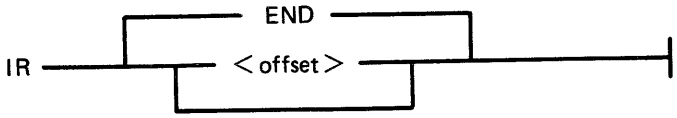
IR (Initiate Log Recall)

(a function of SYS-SUPERUTL)

This function will initiate recall and go back in SYS-LOG files after skipping the number of entries specified by the operator (that is, 5 digit "offset") and display the required message.

This function, once loaded, will remain "active" until another SYS-SUPERUTL function is used. When another function is used, the message "END IR" will appear before any output from the function just requested. It is now possible to enter "IR END" to terminate the function, which will cause "END IR" to be displayed in response.

Format:



where <offset> is a 5-digit number signifying the number of log entries to be skipped before displaying logged messages.

The characters in the range hexadecimal @00@ to @1F@, @7F@ to @9F@ and @FF@ inclusive will be replaced by spaces.

Examples:

To initiate recall after 12 entries and display the message on the console:

IR 12

To initiate recall of the message just given:

IR 1

NOTE

Only log entries which originated at the SPO requesting recall can be recalled, unless it is the controlling SPO, in which case all log entries can be recalled.

Output Messages

Message	Possible Causes	Suggested Action
CANNOT LOCATE DESIRED ENTRY	The log files have been transferred using TL utility, or offset is greater than number of entries in log file.	Decrease the value of offset and re-enter.

For additional information, refer to LD function.

KA (Analyze Disk Space Assignment)

This utility provides the operator with a map of all space used on disk by specific files, or available for other use. The printout is in ascending disk address order in terms of areas and their assignment.

KA is capable of analyzing space assigned to one or more files, one or more groups of files, or the available areas.

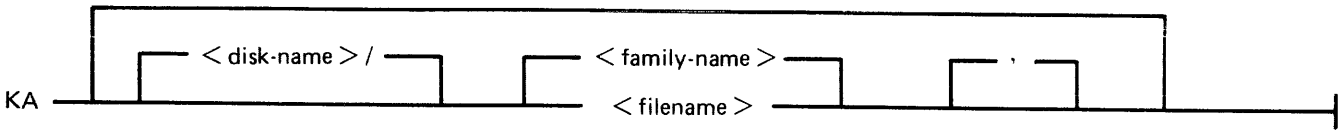
Special reporting is given if the group identifies all files on the disk (that is disk-name/=). In addition to an analysis of the areas allocated to each file, this report will show the space assigned to the disk directory, temporary, available, bad, and missing areas. The temporary areas are those which are allocated either to temporary files or to the virtual memory.

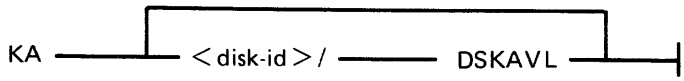
If files are created, extended or deleted by the system during the processing of KA, the map will not be accurate. It is therefore necessary that KA be run only when no other programs are in the mix.

The analyzed output will be to a line or console printer, and will print the areas in ascending disk address order associating with each area its first sector address, its length in sectors and its status. The status will be either allocated, available, temporary, bad or missing. If the area is allocated, the file name of the file to which the area is assigned will also be listed. If a particular file or family is not on-line, then this is indicated on the printout.

If the option DSKAVL is selected, then an analysis of the available areas on the disk specified by "disk-name" (or system disk if no "disk-name" was specified) will be printed.

Format:





Examples:

To analyze disk space assignments of all files on system disk:

KA =

To analyze disk space assignments of all files on the disk called PR2:

KA PR2/=

To analyze disk space assignments for a group of files beginning with the letters "PR" on the system disk, and a file called PR200 on a disk called PR2:

KA PR= PR2/PR200

To analyze available areas on the disk called PRBU:

KA PRBU DSKAVL

Output Format

Display of Bootstraps

Depending on the settings of the initialization system field, MTR field and Coldstart flag in the label and also the disk type, the Disk/Bootstrap types will be identified by one of the following entries which will be printed in the banner heading of KA:

"CMS/WARMSTART DISK"	Removable disks only CMS – B 90
"B900SYSTEM DISK"	Removable disks only B900SYSTEM – B 900
"MTR DISK"	Removable disks only
"CMS/WARMSTART&MTR DISK"	Fixed disks – B 90
"B900SYSTEM&MTR DISK"	Fixed disks (and packs on B 900)
"COLDSTART DISK"	Mini disks – B 90 only

The Bootstrap version number will only be displayed if the Bootstrap located in sectors 2-31 contains its own version number and the correct checkstring. The text

"BOOTSTRAP VERSION <version number>"

will be printed following the Disk/Bootstrap type.

For disks which have been initialized on a B 1900 system, no Disk/Bootstrap type or version number will be displayed.

Six columns of information are output. The column headings, the format of the values these columns contain, and the significance of these values are as follows:

Heading	Value	Significance
AREA ADDRESS	8 digits 6 hex. digits	Sector address of start of area
AREA LENGTH	8 digits 6 hex. digits	Number of sectors in this area
STATUS	9 characters	See Note 1.
FILE NAME	12 characters	See Note 2.

Note 1: The status will be one of AVAILABLE, TEMPORARY, BAD or *MISSING*, depending on whether the area is available, allocated to a file, denoted as temporary, unusable or lost.

Note 2: If the area is ASSIGNED, then this field will contain the identifier of the file residing in the area. If a file belongs to a pseudo disk, its disk name is also listed. Otherwise it will be blank.

The status *MISSING* occurs if an area is not referenced from anywhere within the file directory or available table. This may be because the area is in fact lost, or because existing files have been opened, have had further areas allocated to them and are still open during the processing of KA.

If fixed disk is being used, three areas are reserved for MTR purposes with the status marked as "BAD". The area lengths are 256, 128, and 128 sectors respectively.

Output Messages

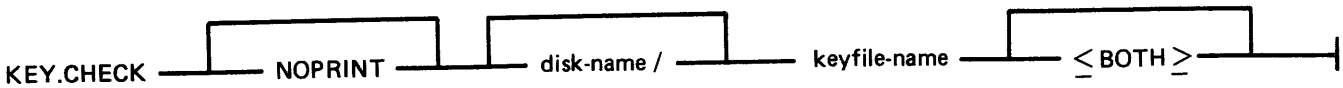
Message	Possible Causes	Suggested Action
<filename> NOT LISTED	The utility could not list the specified filename. The reason is given in the second message displayed.	Refer to the second error message to find the reason.
FAMILY <family name> NOT LISTED	The utility could not list the specified family. The reason is given in the second error message displayed.	Refer to the second error message to find the reason.
– <diskname> IS NOT A PHYSICAL DISK	An attempt was made to run KA on a pseudo disk.	Run KA on physical disk containing the pseudo disk.
AREA APPARENTLY ASSIGNED TWICE	Area is contained partly or completely in an area previously listed.	Run SQ <diskname> VERIFY. If, after successful completion of the SQ, the problem still exists, ask for technical assistance.
AREA ASSIGNED BEYOND MAXIMUM ADDRESS	Area is assigned beyond the addressable space.	Run SQ <diskname> VERIFY. If, after successful completion of the SQ, the problem still exists, ask for technical assistance.
THIS AREA IS MISSING	Area is not referenced from anywhere within the file directory or the available table.	Run SQ <diskname> VERIFY. If, after successful completion of the SQ, the problem still exists, ask for technical assistance.

Note: Refer to "Common Utility Output Messages" for additional aid.

KEY.CHECK

This utility allows the operator to check and print the information on the validity of keys in an indexed pair of files.

Format:



This utility does not provide the *<file-name> option in the initiating message.

During the execution of the utility, the operator is informed if the data file has been updated via another key file or parity errors have occurred on the data file since creation of the specified key file. The checking is performed in two ways, Key file to Data file check, and Data file to Key file check.

In Key file to Data file checking, the key value field of each entry in the key file is compared with the key field in the corresponding record of the data file. This comparison will detect any changes to the keys in the data file records and disused entries in the key file.

In Data file to Key file checking, the key field of every non-deleted record in the data file will be checked to have an entry in the key file. The record written to the data file via another key file and records with invalid keys will be detected, as they will have no entry in the key file.

The NOPRINT option is provided to permit execution of the utility on a system with no printer. It is invoked by specifying NOPRINT at the beginning of the initiating message.

SPO messages are provided for resulting output from KEY.CHECK with the NOPRINT option specified. A message is displayed for only the first discrepancy the utility finds between the key file and the data file (see Output Messages).

If the <BOTH> option is specified in the initiating message, then Key file to Data file and Data file to Key file will be checked, otherwise only Key file to Data file will be checked.

The utility will terminate if a parity error is encountered on the key file.

If the Generation number of the key file differs from that of the data file, then a warning is printed. (Generation number is a field in the File Parameter Block, refer to MCP manual.)

The Generation number of the key file will be modified to that of the associated data file on completion of a Key file to Data file check provided that the following conditions are satisfied:

1. Every key entered in the key file matches the key field in the associated data file record.
2. Every non-deleted data file record has a key entry in the key file.
3. The number of matched key entries in the key file is equal to the number of data file records with a key entry in the key file.
4. There are no parity errors on either the key file or the data file.

On completion of a printer listing execution of this utility, the total number of discrepancies found on checking each file is printed on completion of the key file and data file check.

Keys printed by the utility will be displayed both in ASCII and hexadecimal.

KEY.CHECK will open the keyfile and datafile with OTHERUSE of LOCK.ACCESS. This means that no other program can use the keyfile and datafile while KEY.CHECK is using these files.

The following message will now be displayed (if NOPRINT was specified) or will appear in the summary section of the printer listing if the number of matched keyfile entries is not equal to the number of datafile records with a key in the keyfile.

"THE NUMBER OF MATCHED KEYFILE ENTRIES IS NOT EQUAL TO THE NUMBER OF DATA RECORDS WITH A KEY IN THE KEYFILE"

The files will not be consistent if the message is displayed.

The following totals will also be included in the summary section of the printer listing.

"NUMBER OF MATCH KEYFILE ENTRIES – x"

"NUMBER OF DATAFILE RECORDS WITH A KEY IN KEYFILE – y"

Output Format

The output format is self-explanatory.

Examples

Check the key file PQR, performing Key file to Data file check only:
KEY.CHECK PQR

Check the key file PQR, performing Key file to Data file check as well as Data file to Key file check:
KEY.CHECK PQR <BOTH>

Output Messages

Message	Possible Causes	Suggested Action
<filename> NOT CHECKED	The utility could not check the specified file. The reason is given in the second message displayed.	See the second error message displayed by the utility for the reason.
–FILE IS NOT A KEYFILE	Self-explanatory.	None.
–FILE IS A NULL-KEY KEYFILE	The specified file is a tag-file, not a keyfile.	See section 5 of SOG on SORT for meanings; this utility cannot check tagfiles.
–HARDWARE ERROR DETECTED WHILE READING KEYFILE <filename>	The utility has encountered an error in the keyfile.	Remove keyfile and use the SORT utility to produce a new keyfile.
–KEYFILE <filename> WAS NOT CREATED FOR DATAFILE <FILENAME>	The key offset and length of the keyfile are beyond the end of the data record.	Use the SORT utility to produce a new keyfile.

(continued)

Message	Possible Causes	Suggested Action
-NO RECORDS IN KEYFILE <filename>	Self-explanatory.	None.
-<FILENAME> IS NOT A DATAFILE	The file identified in the KFPB of the specified keyfile as the associated datafile does not have a filetype @00@.	Check the filename and re-enter.
GENERATION NO. MISMATCH MATCH BETWEEN <keyfilename> AND DATAFILE <filename>	The generation numbers of the files are inconsistent. After the KEY.CHECK, the generation number will be corrected.	None.
FIRST INDEX DISCREPANCY FOUND BETWEEN KEYFILE <keyfilename> AND DATAFILE <filename>	The first error has been encountered between the keyfile and the datafile.	None.
-<NUMBER> INDEX DISCREPANCIE(S) FOUND WITH <filename>	The check on the keyfile has been completed and "X" number of discrepancies have been found.	None.
<filename> CHECKED	Normal output message.	None.
CHECK COMMENCING ON DATAFILE <filename>	The check on the keyfile has been completed and the datafile check has been invoked.	None.
KEYFILE <filename> CURRENTLY INDEXES DATAFILE <filename>	Shows which datafile is related to the specified keyfile.	None.
-GENERATION NO. OF KEYFILE <filename> MODIFIED TO <number>	A generation number mismatch was found and a correction has been applied.	None.
-UNABLE TO MODIFY GENERATION NO. OF KEYFILE <filename>	Self-explanatory.	None.
KEYFILE <filename> DOES NOT CORRECTLY INDEX DATAFILE <filename>	Discrepancies have been found on one or both files.	None.
KEYFILE <filename> IS ASSOCIATED WITH <filename>	Shows relation between keyfile and datafile.	None.
UNABLE TO ACCESS <filename>	The utility cannot run. The reason is given the second message displayed.	Look for the reason in the second message displayed by the utility.

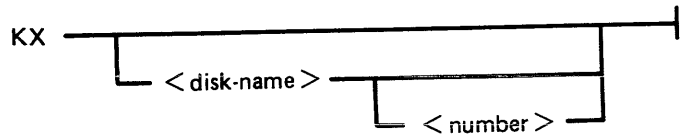
(continued)

Message	Possible Causes	Suggested Action
-UNABLE TO ACCESS ASSOCIATED DATAFILE <FILENAME>	The utility cannot access the datafile related to the keyfile entered in the initiating message.	Look for the reason in the second message displayed by the utility.
-NO. OF VALID KEYS IS INCONSISTENT WITH THE NO. OF RECORDS CONTAINING DATA	Discrepancies have been found on one or both files.	None.

KX (Disk Allocation Information)

(a function of SYS-SUPERUTL)

This function will allow the operator to display the names of all the files found on the disk specified by "disk name" (or on the system disk if no "disk-name" is specified) whose total number of sectors allocated is equal to or greater than "number" (assumed zero if not specified). The disk concerned may not be a pseudo pack.



After each display, which will include the information of the current numbers of temporary and available sectors, the KX function of SYS-SUPERUTL remains available, waiting for one of the following input responses:

A call on any other function of SYS-SUPERUTL: this will terminate KX.

KX or KX NEXT To display the next file name, if any, otherwise KX will go to END.

KX RM or KX REMOVE To remove the file whose name has just been displayed.

KX END To terminate KX.

NOTE

The <number> option specifying the size of the files in KX can only be changed by ending KX and starting it again with another value in the number.

Examples:

To display the name of the first file on a disk called "SYS" whose size is equal to or greater than 250 sectors:

```
KX SYS 250
```

To display the name of the next file whose size is equal to or greater than 250 sectors:

```
KX NEXT
```

To remove the file just displayed:

```
KX RM
```

To terminate KX:

```
KX END
```

Output Messages

Message	Possible Causes	Suggested Action
<filename> OCCUPIES <integer> SECTORS - TOTAL AVAILABLE SECTORS = n, TOTAL TEMPORARY SECTORS = <integer>	Normal output.	None.
- <disk> IS NOT A PHYSICAL DISK	KX may not be run on a pseudo disk.	None.

LB (Look Back in Log)

(a function of SYS-SUPERUTL)

This function will Look Back to continue recall in the direction of earlier messages with a screenful of messages. If the serial printer (SPA) is used as the console, then the function will display a number of messages calculated by the length of messages and width of console.

It is possible to terminate this function by entering "LB END", causing "END LB" to be displayed.

Railroad diagram:

Example:

To look back and display the messages:

```
LB
```

LB can be initiated only after IR, LB, and LF.

Output Messages

Message	Possible Causes	Suggested Action
HARDWARE ERROR DETECTED BY <utility> RECALL MAY BE INCOMPLETE	Self-explanatory.	None.
ENTRIES NOT RECALLED	Self-explanatory.	None.
PREVIOUS FUNCTION WAS NOT <utility>	An invalid END command was entered.	None.
-IR REQUIRED TO INITIATE LOG RECALL	An attempt was made to do an LB or LF without doing a IR before it.	Initiate IR, then re-input LB or LF.
-CANNOT LOCATE DESIRED ENTRY	The offset in the IR command is too high.	Enter IR with a smaller offset.
-NO FURTHER ENTRIES FOR <utility>	LB has reached the beginning of this log file, or LF has reached the end of the log file.	None.
-<file> NOT A RECOGNIZED LOG FILE	The filetype of the logfile has changed.	Error in SYS-SUPERUTL. Copy good SYS-SUPERUTL from backup disk and re-start the system.

LD – TAPE LIBRARY UTILITY (ADD, LOAD, DUMP, UNLOAD)

The LD utility has two formats:

Format 1. Loading files from TAPE onto DISK (ADD, LOAD)

LOAD and ADD provide the capability of copying files or groups of files from a library tape to the disk specified by name. If no disk name is specified, the system disk is assumed.

LOAD	If LOAD is specified, the files are copied to the disk and any duplicate files are removed.
ADD	If ADD is specified, then only files which do not already have copies on the disk are loaded.

Format 2. Dumping files from DISK to TAPE (DUMP, UNLOAD)

DUMP and UNLOAD provide the capability of copying files or groups of files from the specified named disk to a library tape. If no disk name is given, the system disk is assumed.

UNLOAD	If UNLOAD is specified, then the files are deleted from the disk after they have been copied to the tape.
DUMP	If DUMP is specified, then the files are not deleted from the disk after they have been copied to the tape.

If the utility LD resides on the system disk, only the function name has to appear in the initiating message. For example, the input

```
DUMP TO ARTAPE AR=
```

causes the same action as

```
LD DUMP TO ARTAPE AR=
```

If the LD utility does not reside on the system disk, then the initiating message must contain "<disk-name containing LD>/LD".

Detailed descriptions of ADD, LOAD, DUMP and UNLOAD and associated output messages are provided under the name of the function.

"UPDATED" Bit Options

1. DUMP	All "UPDATED" bits will be reset.
2. DUMP & CHECK	LD will not alter the "UPDATED" bits. At the end of the dump, LD will Close Release the tape and zip CHECKADUMP without pause before going to EOJ. CHECKADUMP will RESET the "UPDATED" bit of any file which compares successfully.

	NOTE: If CHECKADUMP is executed independently, it will not alter the "UPDATED" bits.
3. LOAD	All "UPDATED" bits will be SET automatically by the MCP as a result of closing a new output file.
4. LOAD & CHECK	This is identical to LOAD with regard to the "UPDATED" bits. The only difference is that LD will Close Release the tape at the end of the load, and zip CHECKADUMP without pause before terminating.
	CHECKADUMP will not alter the "UPDATED" bits.
5. DUMP UPDATED and DUMP & CHECK UPDATED	These operate like DUMP and DUMP & CHECK except that only files with the "UPDATED" bit SET will be candidates for dumping. The "UPDATED" bit set indicates that the file has been created or updated since the last dump was taken.
6. ADD	All "UPDATED" bits will be SET automatically by the MCP as a result of closing a new output file.

Output Messages From LD – ADD/DUMP/LOAD/UNLOAD

Message	Possible Causes	Suggested Action
<filename> REMOVED	The utility has successfully loaded a file which was already on disk, or the utility has removed the file as a result of an UNLOAD request.	None.
–ASSOCIATED DATAFILE <filename> NOT FOUND	The utility cannot find the datafile belonging to a keyfile it has to LOAD/ ADD/DUMP/UNLOAD.	Make file available and re-enter.
–FILE NAME <filename> EXCEEDS 12 CHARACTERS	Self-explanatory	Check input and re-enter.
<filename> NOT LOADED	The utility could not load the specified <filename>.	Refer to the second error message displayed to find the reason.
–<tapename> IS NOT A RECOGNIZED LOAD/ DUMP TAPE	The specified tape is not created using the LD utility.	None.
<file-id> LOADED	Normal operation. File with name <file-id> is successfully loaded.	None.
<file-id> DUMPED	Normal operation. File with name <file-id> is successfully dumped.	None.
<file-id> NOT LOADED	Specified file is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message displayed by the utility.

(continued)

Message	Possible Causes	Suggested Action
<file-id> NOT DUMPED	Specified file is not loaded. The reason is given in the second message.	Look for the reason in the second message displayed by the utility.
<family> NOT LOADED	Specified family is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message displayed.
<family> NOT DUMPED	Specified family is not loaded. The reason is given in the second message displayed.	Look for the reason in the second message displayed.
-ASSOCIATED DATAFILE <file-id> NOT FOUND	The <BOTH> option was used but the datafile of an indexed pair cannot be found. The keyfile is not DUMPed.	Check the input disk for the presence of the specified files. Re-run the utility.
-INCONSISTENT FILE SIZE AT DUMP TIME	DFH file-size and number of records DUMPed or LOADed are not equal.	Investigate discrepancy and re-run the utility.
-FILE HAS BEEN REMOVED	The utility is unable to dump the file while it is removed.	Check status of files not dumped and re-run the utility.
-FILE IS OPEN SHARED OR FOR OUTPUT USE	Another program is using the specified file.	Wait for the file to be released by the program.
-FILE HAS BEEN ALTERED	The file specifications have been changed since the utility was started.	Check status of file not dumped and re-run the utility.
-FILE HAS NOT BEEN UPDATED	The UPDATED option is specified for a file which has not been updated.	Check status of file not dumped.
-HARDWARE ERROR DETECTED WHILE READING PSEUDO-PACK TABLE ON <disk-id>	The utility has encountered a read error on the PPIT of the input disk.	None.
-REQUESTED FILES WERE NOT FOUND	the specified files are not in the directory of the specified disk.	Check initiating message for correct filenames.
-HARDWARE ERROR DETECTED WHILE READING DISK <disk-id>	The utility has encountered a read error on the input disk. If the utility continues it has been able to recover.	None.
-FILE IS ALREADY ON DISK	The specified filename already exists in the directory of the disk.	None.

(continued)

Message	Possible Causes	Suggested Action
-ASSOCIATED DATAFILE IS ON ANOTHER DISK	The datafile of an indexed pair is not on the specified disk. The utility will continue but the keyfile is not LOADed.	Check the initiating message and print the back-up disk directory.
-FILE WAS NOT SUCCESSFULLY DUMPED	The specified file is not dumped because the file is changed or moved since the start of the utility.	Check file status and re-run the utility.
INCORRECT INPUT TO LD	There is an error in the initiating message. The explanation is given in the second message displayed.	Look for the explanation in the second error message given by the utility.
-INVALID CHARACTER IN <identifier>	One of the characters is outside the range A to Z, 0 to 9, "." or "-".	Correct input and re-enter.
-FILE NAME <file-id> EXCEEDS 12 CHARACTERS	The specified filename is more than 12 characters long.	Correct input and re-enter.
-<specified option> IS NOT A VALID OPTION.	The UPDATED option is used in the initiating message for LOADing or ADDing.	Correct input and re-enter.
-NO FILE NAME SPECIFIED	There is no file name specified in the initiating message.	Correct input and re-enter.
-CANNOT HANDLE A REQUEST OF THIS LENGTH	The input message exceeds 601 characters.	Correct input and re-enter.
-FROM/TO EXPECTED	The utility expects the word FROM when ADDing or LOADing.	Correct input and re-enter.
-DISK NAME <disk-id> EXCEEDS 7 CHARACTERS	The diskname <disk-id> in the initiating message is too long.	Correct input and re-enter.
-OPTION <specified option> HAS BEEN SPECIFIED MORE THAN ONCE FOR FILE <file-id>	At least one of the options occurs more than once in the initiating message.	Check input and correct.
-OPTION <specified option> HAS BEEN SPECIFIED MORE THAN ONCE FOR FILE <family-id>	At least one of the options occurs more than once in the initiating message.	Check input and re-enter.

(continued)

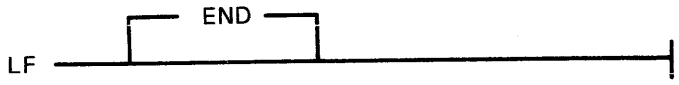
Message	Possible Causes	Suggested Action
-FILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	The specified file is not available because another program is using it.	Wait until the other program has finished, then re-enter.
<file-id> REMOVED	The utility has found a duplicate filename during a LOAD or removes the file as a result of an UNLOAD.	None.
-NO FILES TO <DUMP/LOAD>	None of the files specified in the initiating message is suitable for DUMPing or LOADing.	Check input and re-enter.
-LD CANNOT HANDLE MORE THAN 2804 FILES, LIMIT REACHED AT FILE <file-id>	A request was made to DUMP or LOAD more than 2804 files.	Check input and divide the files to be loaded or dumped into two groups.
-FAMILY MAY BE INCOMPLETE	Part of the requested files is on a pseudo-disk pertaining to a physical disk which is not ready.	Ready the disk and re-enter.
-FAMILY FOR <family-name> MAY BE INCOMPLETE	Part of the requested files is on a pseudo-disk pertaining to a physical disk which is not ready.	Ready the disk and re-enter.
TAPE <tape-id> SHOULD HAVE BEEN PURGED	The information on the tape is such that LD cannot handle it.	Recreate a dumptape after purging the tape.
-PARAMETER FOLLOWING KEYWORD <string> IS INVALID	The utility has encountered an error in the initiating message.	Correct input and re-enter.
THE UTILITY HAS ATTEMPTED TO ISSUE A NON-IMPLEMENTED COMMUNICATE	There is a mismatch between the MCP and the LD utility.	Copy both MCP and LD utility from a good backup disk. Re-enter.
-<tapename> IS NO LONGER A VALID LOAD/DUMP TAPE, AND SHOULD BE PURGED	The utility has encountered a parity error on the specified tape.	Replace medium.
-FILE IS INCOMPLETE	The specified file is a part of a dual pack file.	Make the other part available and re-enter.
ZIP FAILURE WHEN ZIPPING CHECKADUMP FROM LD	LD has tried to zip a checkadump as & CHECK was specified in the initiating message.	Make CHECKADUMP available and enter Y.
-PLEASE ENTER Y TO RETRY THE ZIP OR N TO TERMINATE	Self-explanatory.	Enter y or N using the AX command.

LF (Look Forward in Log)

(a function of SYS-SUPERUTL)

This function will look forward to continue recall in the direction of later messages with a screenful of messages. If the serial printer (SPA) is used as the console, then the function will display a number of messages calculated by the length of the messages and the width of the console.

This function may be terminated by entering "LF END" causing "END LF" to be displayed in response.



Example:

To look forward from last recall and display messages:

```
LF
```

LF can be initiated only after IR, LB, and LF.

Output Messages

For output messages refer to LB command.

LIST (File List)

This utility allows the operator to list, in whole or in part, files on any CMS device. Output will be either to the line printer or to the console printer.

LIST will attempt to open SYSMEM on all PPIT listed units for directory scanning. It will check the PPIT entry for System Pseudo disk-name.

See figure 4-5 for railroad diagram of LIST.

If the "A" option is chosen, the file will be listed in alpha characters. The "N" option will list the file entirely in hexadecimal. If neither the "A" nor "N" options IS selected, the file will be listed in both alpha and hexadecimal.

If the file to be listed is a keyfile, the utility will list the associated data file in the order of the keyfile unless the <KEY> option is specified. When the <KEY> option is used, the utility will list the keyfile itself.

The operator may also list selected parts of a file by specifying the relative record number at which printing should begin and the number of records to be printed from that point.

Examples:

To list the records of a file called PROGSRC as alpha:

LIST PROGSRC A

To print the first record only of a file called PR200 in hexadecimal:

LIST PR200 N 1 1

To list records 100 through 149 of PROGSRC as alpha:

LIST PROGSRC A 100 50

To list keyfiles:

Assume there is a keyfile called PR200K which refers to a data file called PR200.

The statement

LIST PR200K N <KEY> will list all the records of the keyfile PR200K in hexadecimal.

The statement

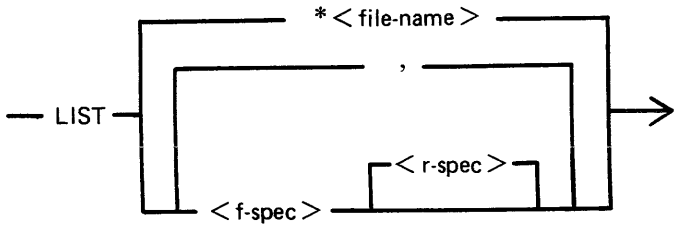
LIST PR200K N will list all records of the data file, PR200 in keyfile order in hexadecimal.

Additional Capabilities

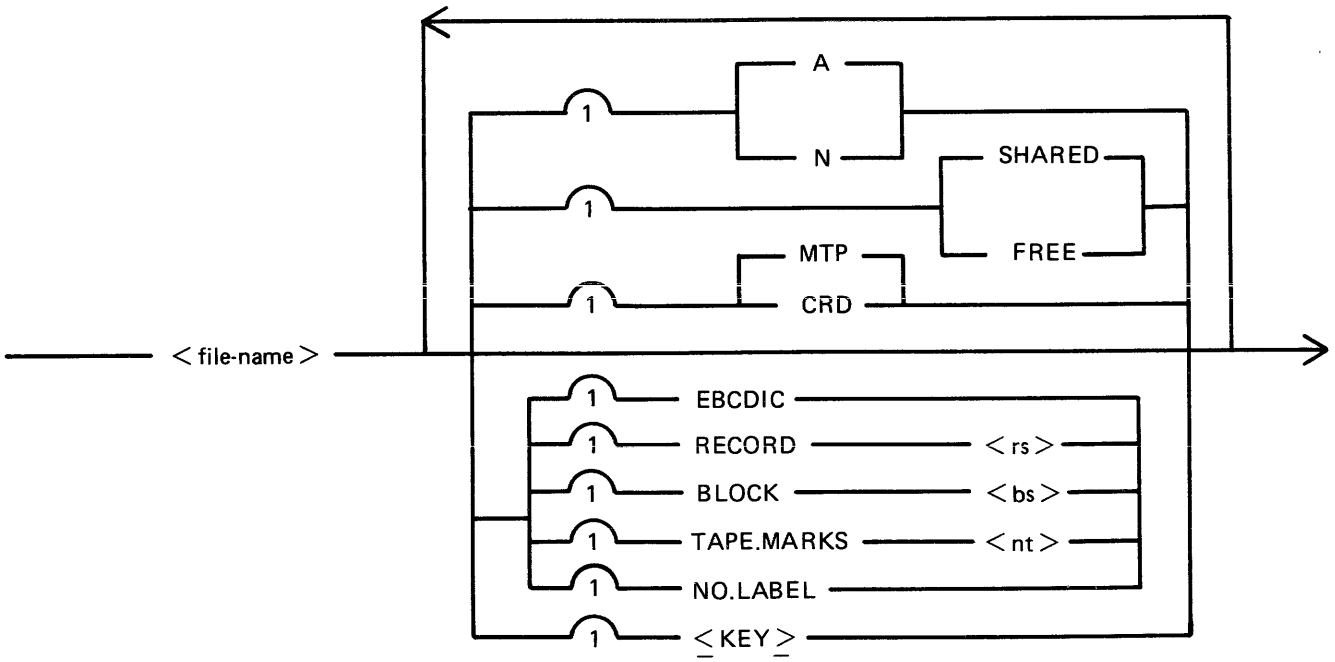
Non Disk Files

Files on media other than disk may be listed. Abbreviations for valid devices are as follows:

MTP – magnetic tape or cassette CRD – punched cards



<f-spec> is defined as :



<r-spec> is defined as :

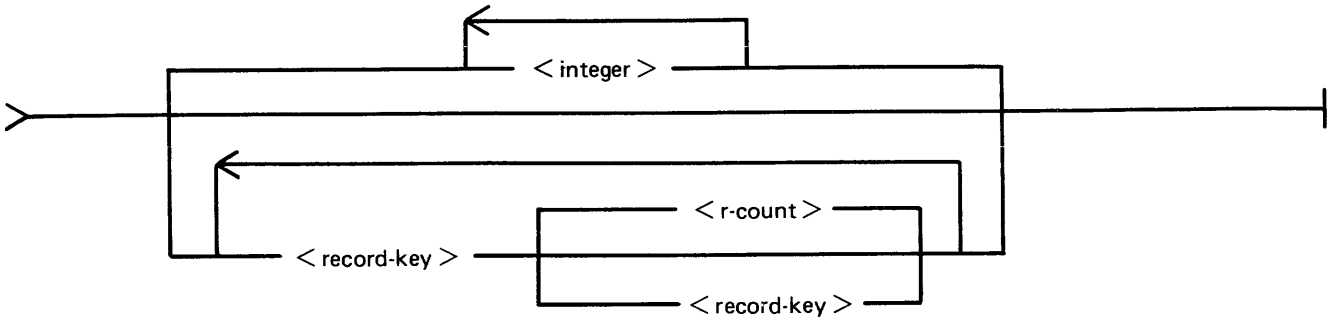


Figure 4-5. Railroad Chart for LIST Utility

Examples:

To list a cardfile called PRFILE in alpha:

```
LIST PRFILE CRD A
```

To list the first 10 records of a CMS labelled magnetic tape called PRTAPE:

```
LIST PRTAPE MTP 1 10
```

(Note: this assumes record size of 180 bytes.) The tape or cassette to be listed should be a tape created by the COPY utility. Library tapes and non-CMS tapes should be treated as unlabelled (see below).

Unlabelled Tapes

Input tapes having no CMS labels ("unlabelled" tapes) may be accessed by the LIST utility.

The NO.LABEL option allows the listing of unlabelled files. On recognizing an unlabelled file, the MCP will print a "DEVICE REQUIRED" message. The operator must then respond with an appropriate "AD" input message (see "AD") to identify the unlabelled file.

The end of file recognition for unlabelled files is determined by tapemark count. The TAPE.MARKS option allows the operator to specify the total number of tapemarks which will indicate end of file to the utility when listing an unlabelled file. The default value is 2. Each tape mark which is encountered will contribute to this total. Therefore, a standard labelled CMS file will be listed up to, but excluding, the trailing label if NO.LABEL and 2 tapemarks are specified. (A labelled CMS file consists of "label, tape mark, data, tape mark, label".) The operator must be aware of the format of any file which is to be listed when using the NO.LABEL option.

If the RECORD size is not 180 bytes, refer to the section on Record/Block modifications.

Example:

To list the first file of a magnetic tape with non-standard label (the format being: label, tapemark, data, tapemark):

```
LIST TP MTP NO.LABEL TAPE.MARKS 2
```

Note: MCP will issue a message asking for unlabelled tape TP. Operator must respond with "AD" input. Additionally, the first line of the listing contains a list of the non-standard label.

Record and Block Sizes

The listing is record-oriented. The following record sizes are assumed:

Disk = (Disk File Header) from file itself

Labelled tape/cassette = from tape label

Unlabelled tape = 180 bytes

Cards = 80 or 96 bytes depending on device.

If different values are required, Record and Block sizes may be specified.

Example:

To list an unlabelled tape containing 10-byte records with 10 records per block:

```
LIST TP MTP NO.LABEL TAPEMARKS 2
RECORD 100 BLOCK 1000
```

If EBCDIC is specified, the input will be translated from EBCDIC coding, otherwise ASCII is assumed.

For magnetic tape or cassette files the record size must be specified if it is greater than 1024 characters, otherwise the utility will not be able to read this file and therefore no list will be produced. If the record size is specified and no block size is specified then the block size will be set to the same as the record size. For unlabelled files the default record and block sizes are 180 each.

Note: Care should be taken to ensure that the record and block sizes specified are compatible with the physical block size on the tape. The block size specified must be an integer multiple of the record size. The utility will attempt to identify inconsistencies when using labelled CMS files. Any inconsistency not isolated by the LIST will cause MCP to discontinue (DS/DP) the utility.

Selected File List

More than one selected portion of the input file may be listed. Pairs of numbers may be specified within each pair. The first number specifies a relative record number and the second specifies the number of records to be listed. If an extra number is specified, the last number specifies listing from that record to the end of file.

Example:

To list records 100 to 149, 300 to 499, and 1000 to end of file.

```
LIST FILE1 100 50 300 200 1000
```

Selected Indexed File List

For indexed files, listing or records can be selected based on content of the key. There are two options: the number of records can be specified or an ending key value can be given.

Examples:

PQR is a keyfile containing personnel records. To list 15 records from the corresponding data file starting from the record with personnel number 01786:

```
LIST PQR 01786 15
```

Using the same keyfile to list all data records from personnel number 01786 to 18000:

```
LIST PQR 01786 - 18000
```

Note: the second option is specified by the hyphen in the LIST statement. Note that at least one space is required before and after all key values (personnel numbers in this case).

Note: when using this option to list from a specified key to another specified key, care must be taken to enter the whole key value including leading zeros. In the case of an incomplete key, the data is left justified and space filled on the right.

Output Messages

Message	Possible Causes	Suggested Action
<filename> NOT LISTED	The utility could not list the specified file. The reason is given in the second message displayed.	See explanation for the error in the second error message displayed by the utility.
-NO RECORDS TO LIST	The specified file is empty	Check input for correct filename.
- INVALID FILE SPECIFICATION FOR <filename>	The specification, or combination of specifications given is not permitted.	Check input for correct specifications.
- FILE IS CURRENTLY USED BY ANOTHER PROGRAM	The specified file is in use by another program.	Use "SHARED" or "FREE" option to list the file.
- RECORD SIZE EXCEEDS MAXIMUM SET FOR THIS RUN, PLEASE RESUBMIT	LIST has encountered a file with a record size greater than expected. This can happen if a magnetic tape file with record size greater than 1024 chararcters is submitted to the utility without the record size being properly specified in the initial input.	Check input for correct record size.
- NO RECORDS IN KEYFILE <file-name>	<BOTH> option was used. Utility was not able to access data file through some failure in the keyfile.	Check input for correct syntax or use SORT to create a new keyfile.
- KEYFILE OR DATAFILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM.	Another program has opened the file for output use.	Use "FREE" or "SHARED" option in initiating message if other program is not using the file with LOCK-ACCESS.
- KEYFILE OR DATAFILE NOT FOUND	Specified file is not in the directory of the specified disk.	Check input for correct disk and filenames.
- KEYFILE OR DATAFILE IS ON A DISK WHICH IS CURRENTLY LOCKED BY ANOTHER PROGRAM	Specified keyfile or datafile is on a disk whose SYSMEM is opened with lock-access by another program	Wait until the other program has finished and re-enter.
- ASSOCIATED DATAFILE <file-name> NOT FOUND	The datafile which belongs to the specified keyfile is not on line.	Bring associated datafile on line and re-enter.
- HARDWARE ERROR DETECTED WHILE READING KEY <key>	Error encountered when reading specified key.	Check datafile for errors or use "RECOVER" for copying the readable records. Then LIST again.

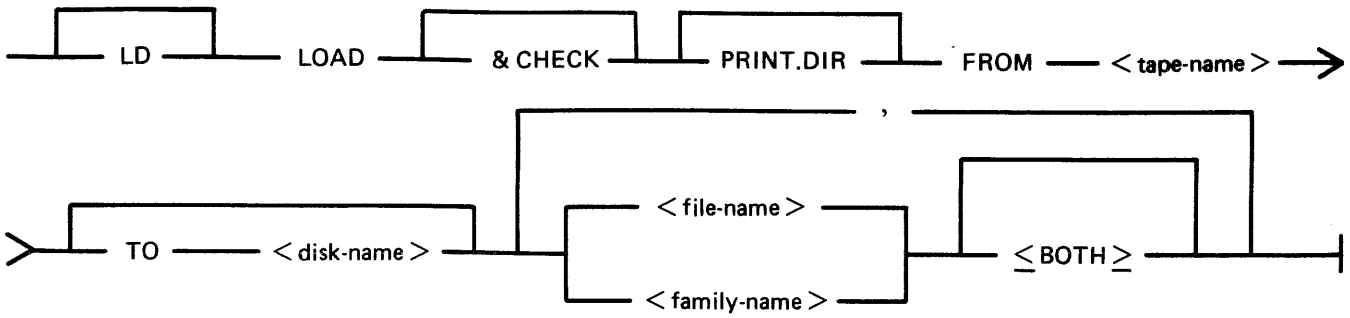
(continued)

Message	Possible Causes	Suggested Action
– HARDWARE ERROR DETECTED WHILE READING RECORD <record number>	Error encountered when reading specified record.	Check datafile for errors or use "RECOVER" for copying the readable records. Then LIST again.
– FILE IS NOT A SOURCE OR DATA FILE	A file has been specified for listing with translation to EBCDIC which is not source or data type.	None.
– PRINTER BACKUP FILE <filename> NEEDS TO BE INCREASED	LIST was printing the printer backup but the filesize of the PB file is too small.	Use SO PRBK <number> to increase the PB filesize.
– FILE IS NOT AN EBCDIC FILE	A disk file has been specified for translation from EBCDIC which is not an EBCDIC data type file.	None.
FAMILY MAY BE INCOMPLETE	A family is specified for listing from a unrestricted pseudo-disk and at least one of the physical disks is not available.	None.
FAMILY FOR LIST MAY BE INCOMPLETE	A family is specified for list from an unrestricted pseudo-disk and at least one of the physical disks is not available.	None.
END OF FILE REACHED ON <file-name> DURING RANGE FROM RECORD <record-number>	End-of-file encountered while section of file indicated by record-number is being listed.	None.
END OF FILE REACHED ON <file-name> DURING RANGE FROM KEY <key>	End-of-file encountered while section of the file indicated by key is being listed.	None.
<file-name> RANGE FROM RECORD <record-number> IGNORED SINCE OUT OF SEQUENCE	A non-disk file or null-key keyfile is being listed with record selection, the selected ranges must be in sequential order.	None.
<file-name> RANGE FROM RECORD <record-number> IGNORED SINCE BEYOND END OF FILE	Once the end of a null-key keyfile or a non-disk file is reached, no further selections can be listed.	None.
–RECORD SIZE LIMIT IS 10000	The utility cannot handle record sizes greater than 10000.	None. Utility limit.

Note: Refer to "Common Utility Output Messages" for additional aid.

LOAD (LD – TAPE LIBRARY UTILITY)

This function is part of the utility LD.LOAD. It provides the capability of copying files or groups of files and removing duplicate files from the disk, if present, from a library tape to the disk specified by disk-name or, if no disk-name is specified, to the system disk.



& CHECK	If & CHECK is specified, CHECKADUMP will be zipped (without pause or display) from LD when the LOAD is complete.
PRINT.DIR	If PRINT.DIR is specified, a printout of all the files loaded will be produced at the end of the LOAD. When PRINT.DIR is specified, the messages "<file-name> REMOVED" "<file-name> LOADED" will be suppressed. Displays will be limited to exception conditions, that is, if a file is not loaded for some reason.
FROM <tape-name>	This is the name of the tape from which the files are to be loaded.
TO <disk-name>	If the TO option is specified, the name of the disk to which the files are to be loaded must be given. If the TO option is not used, then the files are loaded onto the system disk.
<file-name> <family-name>	This is the file or family of files to be loaded onto the disk. Groups of files may be loaded by separating the file names or family names by a comma ",".
<BOTH>	If <BOTH> is specified immediately following a request to load a key file then, provided that the pertinent data file does not precede the key file on the library tape, the data file will also be loaded and a suitable amendment will be made to the key file so that it points to the disk which now holds the data file (rather than the disk from which the data file was dumped).

Since the LOAD function is part of the utility LD, "LD" appears in a mix message. To discontinue the LOAD function, "DS <mix-number>/LD" must be used.

Examples:

Example 1. To copy all the files from the tape named UPDATE to the system disk, the following syntax is used:

LD LOAD & CHECK FROM UPDATE =

With the addition of the keyword "& CHECK", the LD utility will zip the utility CHECKADUMP after all the files have been loaded.

LD LOAD & CHECK FROM UPDATE =

Example 2. To copy the keyfile named QUARCK and its associated data file from the tape called QACK to the disk called Q3, the following syntax is used:

LD LOAD FROM QACK TO Q3 QUARCK <BOTH>

Example 3. To copy all the files from the tape called BACKUP in the family JOB. to the disk named DISK1, the following initiating message is used:

LD LOAD PRINT.DIR FROM BACKUP TO DISK1 JOB.=

With the addition of the option PRINT.DIR, a printout of all the files which LOAD loaded will be produced.

Output Messages

For output messages refer to LD utility.

LOAD.VFU (Load Vertical Format Unit – B 90/B 900 Only)

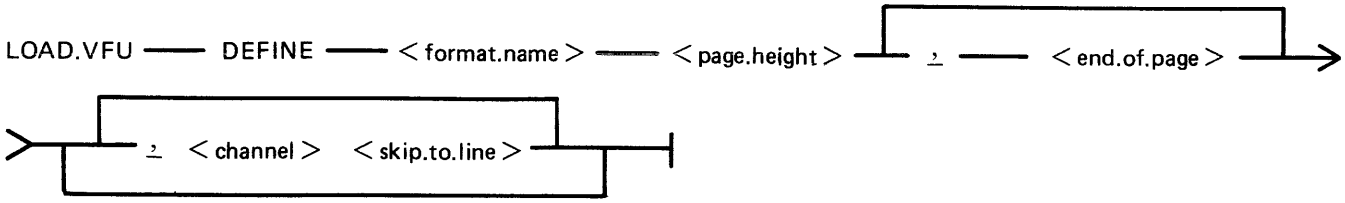
This utility allows the operator to define the page format on a line printer that contains a soft vertical format unit.

NOTE

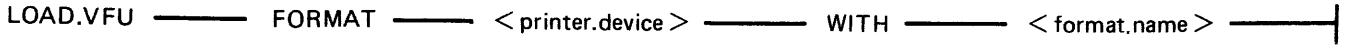
LOAD.VFU cannot be used on B 9252 line printers as it expects input in a format that is incompatible with B 9252.

The utility has 4 possible formats:

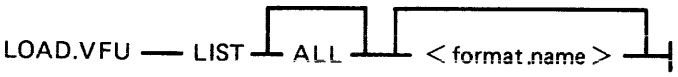
Format for DEFINE:



Format for FORMAT:



Format for LIST:



Format for DELETE:

The utility may be used to define vertical format unit formats and store them in a library file SYSVFU-.LIB. These formats can be subsequently selected by name to be loaded to the specified printer (type A 9249-30/50 or B 9251-1). The utility zips the VF intrinsic which performs the actual loading of a format string.

Define

The define format is used to enter a format in a library file called SYSVFU.LIB. If this file does not exist on the system disk, it will be created.

The FORMAT.NAME you want to create must have a unique name. If the name already exists in the SYSVFU.LIB file, the utility will give an error message.

The PAGE.HEIGHT is the maximum number of lines which can be printed on the selected form to be formatted. (for example, 66 lines on 11 inch pinfeed).

The END.OF.PAGE is the last line on the selected form on which printing is allowed. So, to stop six lines from the bottom on a 66 lines high form, enter 60.

The CHANNEL is a number from 2 to 11 which corresponds to the channel number given in a write statement in an application program.

The SKIP.TO.LINE indicates the print line corresponding to the channel to which the printer has to skip if a write statement with a channel option is executed.

For example, LOAD.VFU DEFINE LP.PAYROLL.FORM 66, 60, 2 10, 4 20

Format

The FORMAT format is used to load a format formerly defined in the SYSVFU.LIB file to a specific printer.

For PRINTER.DEVICE you have to enter a device mnemonic, for example, LPA. The FORMAT-.NAME is the name formerly defined in the SYSVFU.LIB file.

For example, LOAD.VFU FORMAT LPA WITH LP.PAYROLL.FORM

List

The LIST format is used to list one or all of the formats defined in the SYSVFU.LIB file.

The FORMAT.NAME is the name formerly defined in the SYSVFU.LIB file.

For example:
LOAD.VFU LIST ALL
LOAD.VFU LIST LP.PAYROLL.FORM

Delete

The DELETE format is used to delete a format defined in the SYSVFU.LIB file.

For example, LOAD.VFU DELETE LP.PAYROLL.FORM

Limitations

The utility does not check the following:

1. For each channel, no more than four line numbers can be specified.
2. Page height defined should not be greater than 94 lines.

The FORMAT function allows the operator to load a predefined VFU format to a specified printer. The printer must be ON LINE and NOT IN USE.

Note

The default values of page height and end of page are 66 and 60 respectively. Non-default values require that the format should be loaded prior to the execution of the program.

Output Messages

Message	Possible Cause	Suggested Action
INCORRECT INPUT TO LOAD.VFU : <input>	Typing error in initiating message.	Correct input and re-enter.
PRINTER LPx NOT FORMATTED WITH <format>	FORMAT function of LOAD.VFU was not successful.	Look at the second error message given to find the reason.
<format> NOT DEFINED	DEFINE function of LOAD.VFU was not successful.	Look at the second error message given to find the reason.
<format> NOT LISTED	LIST function of LOAD.VFU not successful.	Look at the second error message given to find the reason.
<format> NOT DELETED	DELETE function of LOAD.VFU not successful.	Look at the second message given to find the reason.
-SYSVFU.LIB IS NOT A RECOGNIZED SYSTEM FILE	The filetype of the SYSVFU.LIB file is wrong.	Remove the SYSVFU.LIB file and create a new one using the DEFINE option, or copy a good version SYSVFU.LIB from backup disk.
CANNOT ACCESS SYSVFU.LIB	LOAD.VFU cannot open the SYSVFU.LIB file.	Look at the second message given to find the reason.
-FILE NOT FOUND	A FORMAT, LIST or DELETE function was specified but there is no SYSVFU.LIB file on disk.	Make SYSVFU.LIB file available or create one using the DEFINE option.
-FILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	Self-explanatory.	Wait until the other program stops using the SYSVFU.LIB file.
-HARDWARE ERROR DETECTED WHILE READING SYSVFU.LIB	The utility has found an error on the SYSVFU.LIB file.	Check integrity of the disk.
-HARDWARE ERROR DETECTED WHILE WRITING TO SYSVFU.LIB	The utility has detected an error while adding a new format to the file.	Check integrity of the disk.
-<format> NOT DEFINED IN SYSVFU.LIB	Self-explanatory.	Check input for correct format name or use DEFINE option to create the requested format.
-<format> DEFINED IN SYSVFU.LIB	LOAD.VFU DEFINE was successful.	None.
-<format> DELETED FROM SYSVFU.LIB	LOAD.VFU DELETE was successful.	None.
SYSVFU.LIB REMOVED	LOAD.VFU DELETE was successful on the only format in the file.	None.

(continued)

Message	Possible Cause	Suggested Action
PRINTER LPx FORMATTED WITH <format>	LOAD.VFU FORMAT was successful.	None.
END LOAD.VFU	Normal end-of-job.	None.

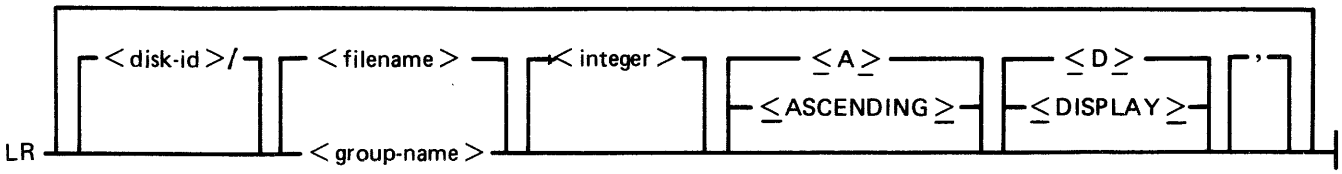
Note

For details of the VF intrinsic see section 3.

LR (List Directory)

This utility allows the operator to display, on the printer or the SPO, detailed information about particular files or groups of files on disk. For each file n the family, if <file-name> is specified, an output line is generated. The overflow pack-id of a dual-pack file will be listed with the size of each area which is on the overflow-pack or on an extra line, if none.

Format:



If a file has areas on an associated overflow disk, the disk name of the overflow disk is printed beside each relevant area address and size. Note that the addresses for the areas on an overflow disk are not necessarily correct.

If a particular file or group is not found on a specified disk, this is indicated on the listing.

If "<ASCENDING>" or "<A>" is selected, the utility will print the information requested in ascending order of filenames. If more filenames or groupnames are in the initiating message and the ascending option is specified once, all the files are printed in ascending order.

If <integer> is specified after the LR of an entire disk (that is, LR ARDISK2/), then LR will only print information about those files whose total number of sectors allocated is greater than <integer>; this will be followed by a listing, with totals, of all available and temporary areas on the disk. This feature is only permitted for physical disks, not pseudo packs.

If "<DISPLAY>" or "< D >" is specified, then the utility will list the information on the SPO. The output format is designed for a 24 line by 80 character screen.

NOTE

The <integer> option described above is not intended for use with the < DISPLAY > option.

An output line concerning a keyfile will normally be followed by a second line showing the name of the data file to which the keyfile points and the key offset and length.

The heading lines printed at the top of each page will provide a good deal of information about the disk itself.

When handling Pseudo-pack disks, the utility will attempt to open SYSMEM on all PPIT listed units for directory scanning and will search for a PPIT entry with a tag of 20 for system Pseudo disk-name.

Examples:

To print the entire directory of the system disk:

```
LR =
```

To print the entire directory of ARDISK2 in ascending order:

```
LR ARDISK2/= <A>
```

To print information about the file called "AR200" and a group of files beginning with the letter "C" only:

```
LR C=, AR200
```

To print information only about files on the system disk which have been allocated greater than 1000 sectors:

```
LR = 1000
```

Output Format

LR displays in its page heading the type and version number of the bootstrap residing on the disk being analyzed. The format is:

```
<bootstrap type> DISK
```

where <bootstrap type> is detailed in the following table:

Bootstrap Type	Disk Type	System Initialized On
CMS/WARMSTART	Removable disks only	B 90
B 900SYSTEM	Removable disks only	B 900
MTR	Removable disks only	B 90/B 900
CMS/WARMSTART&MTR	Fixed disks only	B 90
B 900SYSTEM&MTR	Fixed disks and packs	B 900
COLDSTART	BSMII Mini 1MB Mini	B 90

The bootstrap version number will be displayed if the bootstrap located in track zero, of the disk being analyzed, contains its own version number and correct checkstring. The bootstrap version number appears following the bootstrap type in the following format:

```
BOOTSTRAP VERSION dd.dd.dd
```

Fourteen columns of information will be output to the printer for each disk for which information is requested. The column headings, the format of the values these columns contain, and the significance of these values are as follows:

Heading	Value	Significance
FILE NAME	12 characters	File name.
ACTUAL SIZE	7 digits	Number of records currently contained in file.
MAXIMUM SIZE	7 digits	Maximum number of records which file may contain.
RECORD SIZE	5 digits	Number of bytes per record.
RECORDS BLOCK	5 digits	Number of records in each block.
CREATED	5 digits	File creation date (Julian YYDDD).

(continued)

Heading	Value	Significance
ACCESSED	5 digits	Last access date (Julian YYDDD).
UPDATED	1 character	Flags files which have been updated. See note 1.
GEN. NO.	3 digits	
FILE TYPE	7 characters, 2 hex characters	See note 2.
VERSION NO.	6 digits	See note 3.
NO. AREAS	2 digits	Number of areas currently allocated.
AREA LOCATIONS	8 digits, 6 hex characters, 7 characters	See note 4.
AREA SIZES	8 digits, 6 hex characters	See Note 4.

Note 1. If a file which is being listed has been updated, the letter "U" will be printed on the same line as the file name in the column called UPDATE. The utility identifies that the file has been updated by checking the update flag in the file. The following utilities reset this update flag: LD, CHECKA-DUMP if zipped by LD and DUMPADISK if the dumps are successful.

Note 2. The actual filetype of each file is is displayed with type identifiers.

Note 3. A VERSION NUMBER FIELD is included in the output line for a file. If the file is an S-CODE file, then:

1. If it is a new S-CODE file and a version number is present in its Program Parameter Block (that is, the source file used to produce the S-CODE file contained the dollar option "$ VERSION dd.dd.dd" where d is a decimal digit), the version number field will contain the version number in the Program Parameter Block, in the format dd.dd.dd.
2. If it is an old S-CODE file or no version number is present in its Program Parameter Block, then the version number field will contain ".....".

If the file is not an S-CODE file, then the version number field will contain spaces.

Note 4. For each file the area addresses and sizes of allocated areas will be printed in these columns. For areas on an overflow disk the overflow disk name will follow the area location.

Note 5. If an LR listing of a disk created by DUMPADISK is empty, use PRINT.DIR.

Output Messages

Message	Possible Causes	Suggested Action
<filename> NOT LISTED	The utility has not listed the specified file. The reason is given in the second message displayed.	Look at second error message given by utility to find the reason.
FAMILY <familyname> NOT LISTED	The utility has not listed the specified family. The reason is given in the second message displayed.	See second error message given for the reason.

Message	Possible Cause	Suggested Action
-LR CANNOT HANDLE THIS NUMBER OF DIRECTORY ENTRIES	LR cannot handle more than 2804 entries.	Specify family names to list parts of the disk.

Refer to "Common Utility Output Messages" for additional aid.

MODIFY

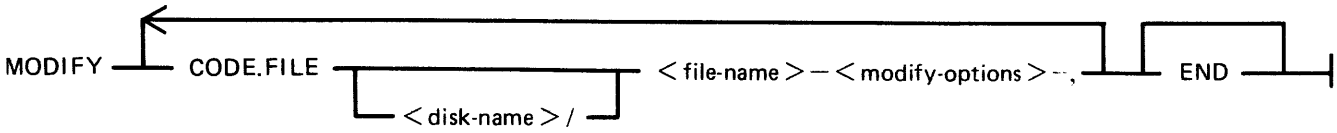
This utility allows the changing of a number of file attributes within the file parameter block (FPB) and program attributes within the program parameter block (PPB) of a code file. It should not be used unless the meaning of each attribute is thoroughly understood. Refer to the CMS MCP manual for more information on FPB and PPB formats.

The utility operates in an interactive manner using a console file or a datacomm terminal if no further information is provided when initiating the utility, thus:

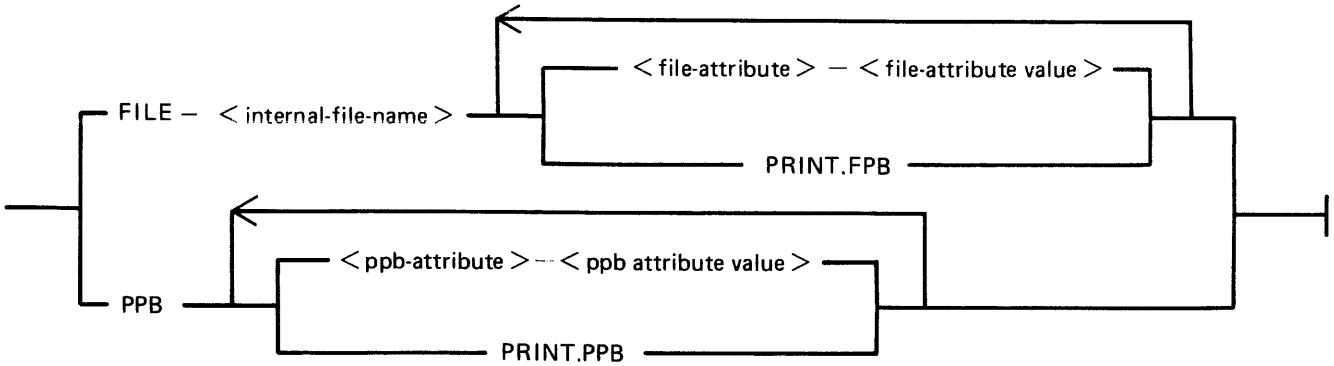
Using console: MODIFY
Using datacomm: <control char> RN MODIFY

For details of the interactive mode, see later. Specifications can be entered when starting the utility. The name of the code file to be modified is preceded by the keyword "CODE.FILE". The word "CODE.FILE" can be omitted from the first element of the initiating message. Following the code file name is either the keyword "FILE" to enable file attributes to be modified, or the keyword "PPB" to enable program attributes to be changed. The file whose attributes are to be changed is specified by the internal file name (i-f-n) as given by the program source code listing. The i-f-n is determined by the programmer. Additional keywords are "PRINT.FPB" and "PRINT.PPB" to print the complete FPB and PPB respectively. The complete specifications to the utility are terminated by the keyword "END".

Format



where <modify options>



The commas are optional, but may be used to improve readability. See later for the list of attributes and allowable values.

When a PPB is being printed, MODIFY checks the release level of the code file, printing the S-LANGUAGE for pre-3.3, and MCP level and HARDWARE type for post-3.2. It should be noted that the MCP level printed for certain released code files is not that of the host system MCP, but that of the system on which the code files were compiled.

If an unknown machine type is encountered, "UNKNOWN" is printed.

The PPB version field (bytes 21-23) will be printed if the priority extension bit (BIT 12) is set. The version will be shown as DD.DD.DD where D is a decimal digit.

The HARDWARE type is indicated as follows:

Byte Value	Type
1 – 10	B 80, B 90
11 – 20	B 776
21 – 30	B 800
31 – 40	B 900
41 – 50	B 1700
51 – 60	B 1800

The second byte of the field in the PPB is machine-dependent and is therefore not printed.

Examples

To modify the value of FID (file-id) and change the device kind of a file whose internal name is INFILE in a program code file COPY on disk SYS2:

```
MODIFY CODE.FILE SYS2/COPY, FILE INFILE FID CARDS DEVICE CR, END
```

To change the value of CONTROL.STACK to 50 in code file AR768 on disk AR1, and print the resultant PPB:

```
MODIFY AR1/AR768 CONTROL.STACK 50 PRINT.PPB END
```

Interactive Mode

Console Operation

If no initiating specifications are given, PKs 1 to 6 are lit for various functions.

PK1	PK2	PK3	PK4	PK5	PK6
help	modify PPB	modify FPB	specify code file	print FPB or PPB	END

Pressing PK1 gives a display of the meanings of the six PKs, as shown above, followed by the request

```
CODE.FILE?
```

Enter the code file name, followed by OCK1. The utility requests

```
SELECT FUNCTION
```

and lights appropriate PKs. While any relevant PK is lit, the corresponding function can be started.

If PK2 (modify PPB) or PK3 (modify FPB) is pressed, the utility requests

```
PPB ATTRIBUTE or
FPB ATTRIBUTE
```

Enter the name of the attribute, as given in table 4-6. The utility displays the current value, then requests

```
NEW VALUE
```

Enter the new value required. The utility then returns to the select function loop.

Datacomm Operation

If MODIFY is given a datacomm start (for example, $RN MODIFY) any initiating message is ignored. The terminal must be known to the system by a prefix "TD830" and must have TD 830 capabilities. MODIFY supports a GEMCOS interface and a shifted TMCS interface.

The program, once started, is menu-driven. The first screen shows four selectable options:

1. Terminate the program.
2. Enquiry/modification of a code file's program parameter block.
3. Enquiry/modification of one of a code file's file parameter block
4: Enquiry of the filenames in a code file's internal file name block

Option 1.

This option will terminate the program.

Options 2 and 3

If either of these options is chosen, the relevant field names and associated data are automatically presented on the screen. Any redefinable fields have their data presented between forms characters, so the terminal is in forms mode and data which can be changed is in an unprotected field.

On the bottom of the page there are six options offered to the user:

1:	"T"	Terminate
2:	"O"	Other page
3:	"P"	Print information
4:	"U"	Update
5:	NUMBER	Displays a help screen
6:	BLANK	Back to the main menu

1: "T"
Terminates the session and displays the termination screen on the users terminal.

2: "O"
Produces another screen with the rest of the fields which did not fit on the first page.

Page 1 and page 2 together make up a complete display of either a Program Parameter Block or a File Parameter Block.

3: "P"
Produces a print of the information presented on the screen. If no line printer is available and printer backup is possible, the printer backup filename will be displayed on the status line.

4: "U"
Will force the utility to update the parameters in the codefile if the updated data in the forms fields is valid.

5: NUMBER
Displays a help screen for the item whose number is entered. To return to the calling screen press XMT.

6: BLANK
When nothing is entered the program returns to the main menu without updating any fields.

Option 4

This option produces a screen with the filenames held in the internal file block of the program name entered on the menuscreen. At this point there are five options.

1: "T" Terminate
2: "O" Other page
3: "P" Print information
4: NUMBER Go to FPB analyses for the associated file
5: BLANK Back to main menu

1: "T"
Terminates the session and displays the termination screen on the users terminal.

2: "O"
Produces another screen with the rest of the file names which did not fit on the first page.

3: "P"
Produces a print of the information presented on the screen. If no line printer is available and printer backup is possible, the printer backup filename will be displayed on the status line.

4: NUMBER
Every filename is associated with a number on the forms screen. Typing in one of these numbers forces the program to go directly to the File Parameter Block analysis of the specified file.

5: BLANK
When nothing is entered the program returns to the main menu.

File Attributes – Interactive/Non-interactive Analysis

Table 4-6 gives the keywords for each file attribute that can be changed by the MODIFY utility, together with allowable values for each attribute. Table 4-7 gives the keywords of each PPB attribute that can be changed, and allowable values for each.

Note that each modification is performed in turn, so that the keywords PRINT.FPB and PRINT.PPB will reflect the FPB and PPB after any modifications specified previously in the message to MODIFY, but before any modifications are made that are specified after the print request.

Most attributes that require a number may be assigned a hex value by supplying @hh@ where hh is a string of characters in the set 0..9, A..F.

Attributes which require a character representation of a decimal number, for example, REEL, will not accept a hex number.

Output Messages

Message	Possible Cause	Suggested Action
ATTRIBUTE VALUE MISSING	Value is either missing or incorrect; other modifications carried out.	Check current values by PRINT.FPB or PRINT.PPB; then use utility for the attribute in error.

(continued)

Message	Possible Cause	Suggested Action
KEYWORD IN ERROR	Self-explanatory; other modifications carried out.	As above.
ATTRIBUTE-VAL INCONSISTENT	The attribute being assigned cannot take the value being given; other modifications carried out.	As above.
INCORRECT ATTRIBUTE	A value is being assigned to a value, rather than to an attribute; other modifications carried out.	As above.
DEVICE – MYUSE INCONSISTENT	Incompatible values of these file attributes. This is a warning that the program may give an error when executed.	use MODIFY to correct either or both of these fields.
FILE-SIZE TOO LARGE	Value for FILE-SIZE is incorrect. This is a warning.	Check with PRINT.FPB if necessary and correct the attribute.
TOO MANY BUFFERS	Value for NO.BUFFERS is incorrect. This is a warning.	As above.
REC. NOT INTEGRAL OF BUF.	The buffer size is not an exact multiple of the record size. This is a warning.	Use MODIFY to correct one or both of these attributes before running program.
CODE FILE NAME IN ERROR	Self-explanatory; all modifications are ignored.	Re-input.
FILE NAME NOT FOUND	The internal file name is not in code file; all modifications are ignored.	As above.
CURRENCY SYMBOL EXPECTED	Self-explanatory	Re-input.
NUMERIC ATTRIBUTE – VAL REQD	Non-numeric characters were input where a numeric value is needed.	Check with tables 4-6 and 4-7 and re-input.
FILE NOT SPECIFIED	Missing keyword "FILE"	Re-input.
PPB NOT SPECIFIED	Attempt to modify PPB while not in PPB mode.	Re-input.
NOT AN INDEXED FILE	Attempted to use an indexed-file attribute on a non-indexed file.	Check initial input to MODIFY
NOT A COBOL PROGRAM FILE	Wrong PPB attribute used (see table)	Re-input.
NOT AN MPL PROGRAM FILE	Wrong PPB attribute used (see table)	Re-input.

(continued)

Message	Possible Cause	Suggested Action
NOT A PPB ATTRIBUTE	Attempted to modify an attribute which is not of the PPB.	Check input and re-enter.
NOT AN FPB ATTRIBUTE	Attempted to modify an attribute which was not of the FPB.	Check input and re-enter.
CODE FILE – BAD FILE TYPE	Attempted to modify a code file which was not of type CODE.	None.
CODE FILE IN USE	The specified code file is in use.	Wait until the task using code file goes to End of Job.
CANNOT OPEN THAT CODE FILE	Attempted to modify a code file which was not available for some reason.	None.
NUMBER TOO BIG	Attempted to assign a value greater than 65535.	None.
MUST HAVE 0 < KEY LENGTH < 29	Attempted to assign a KEY LENGTH value out of range.	None.
DECIMAL NUMBER REQUIRED	Decimal number required.	Re-enter with decimal number.
FILETYPE NOT 0-15 OR @80@	New value of FILETYPE not in range.	Re-enter.
FILE NOT DATA OR INX TYPE	Old value of FILETYPE not in range.	Re-enter.

Table 4-6. File Attributes Accessible by MODIFY

File Attribute Name	Allowable Values
MFID	1-7 alphanumeric characters
FID	1-12 alphanumeric characters
REEL	3 decimal digits in range 000-999
DEVICE	One of the mnemonics given in table 4-8.
RECORD	1-5 decimal digits in range 0-65535
BUFFER	1-5 decimal digits in range 0-65535
FILESIZE	1-7 decimal digits in range 0-1048560
NO.BUFFERS	1-2 decimal digits in range 1-16
CYCLE	2 decimal digits in range 00-99
FORMS	ON, OFF
SET.UPDATE	ON, OFF
NO.LABEL	ON, OFF
CONDITIONAL	ON, OFF
SINGLEAREA	ON, OFF
GEN.CHECK	ON, OFF
NO.REWIND	ON, OFF
REVERSE.ESCAPE	ON, OFF

Table 4-6. File Attributes Accessible by MODIFY
(continued)

File Attribute Name	Allowable Values
CLOSEMODE	LOCK, PURGE, REMOVE, RELEASE, HALF.CLOSE
CRUNCH	ON, OFF
MERGE	ON, OFF
OTHERUSE	FREE, LOCK.ACCESS, LOCKED or SHARED
MYUSE	INPUT, OUTPUT, IO
EXTEND	ON, OFF
ACCESSMODE	SEQUENTIAL, STREAM, RANDOM
GEN.NO	1-5 decimal digits in range 0-65535
LAST.ACCESS	5 decimal digits in Julian date format, YYDDD
SAVE	1-3 decimal digits in range 0-999
FILE.DEFAULT	TYPE1 thru TYPE29 (see MPL Reference Manual)
NON.STANDARD	ON, OFF
D.MFID	1-7 alphanumeric characters (indexed files only)
D.FID	1-12 alphanumeric characters (indexed files only)
ROUGH.TABLE	1-5 decimal digits in range 0-65535 (indexed files only)
KEY.LENGTH	1-2 decimal digits in range 1-28 (indexed files only)
KEY.OFFSET	1-5 decimal digits in range 0-65535 (indexed files only)
BACKUPPERMITTED	DONTCARE, DONTBACKUP or MUSTBACKUP
CREATENAMEDBACKUP	ON, OFF
FILETYPE	If original value @80@ or 0-15, then new value must be in same range.
USERNAME	1-17 alphanumeric characters or 0
SECURITY.TYPE	PUBLIC/PRIVATE/SECURED
SECURITY.USE	INPUT/OUTPUT/IO
SECURITY.GUARD.MFID	1-7 alphanumeric characters
SECURITY.GUARD.FID	1-12 alphanumeric characters
HOSTNAME	1-17 alphanumeric characters or " " (blank)

Table 4-7. PPB Attributes Accessible by MODIFY

PPB Attribute Name	Allowable Values
INTERP.PACK	1-7 alphanumeric characters
INTERP.NAME	1-12 alphanumeric characters
CLASS	A, B, C
EOJ.SUPPRESS	ON, OFF
DATA.STACK	1-5 decimal digits in range 0-65535 (MPL/BIL programs only)
CONTROL.STACK	1-5 decimal digits in range 0-65535
CURRENCY.SYMBOL	one character (COBOL/RPG programs only)

Table 4-8. Mnemonics for Device Attribute for MODIFY

Mnemonics	Meaning
PR	Any printer.
KP	Keyboard printer.
KD	Keyboard display.
KB	Keyboard any output.
SP	Serial printer.

Table 4-8. Mnemonics for Device Attribute for MODIFY
(continued)

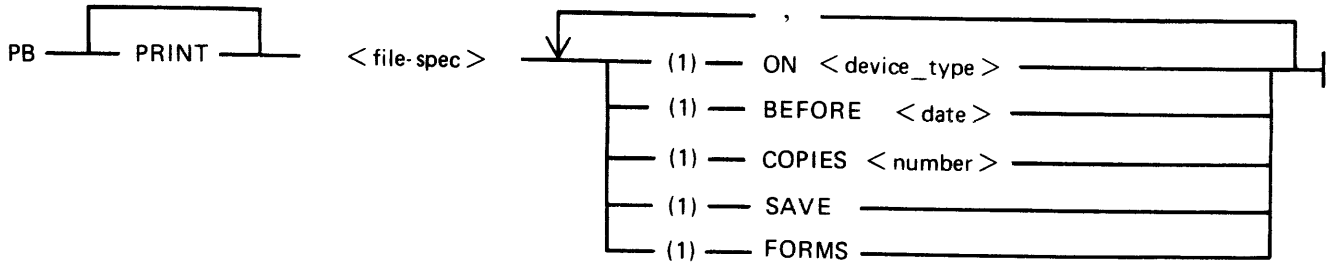
Mnemonics	Meaning
LP	Line printer.
CR	Any card reader.
CP	Any card punch.
CRP	Any card reader/punch.
CR80	80-column card reader.
CP80	80-column card punch.
CRP80	80-column card reader/punch.
CR96	96-column card reader.
CP96	96-column card punch.
CRP96	96-column card reader/punch.
MT	Magnetic tape reel or cassette.
MTIN	Magnetic Tape Reel or cassette without write permit.
MT.PE	P.E. magnetic tape reel or cassette.
MTIN.PE	P.E. magnetic tape reel or cassette. without write permit.
MT.NRZ	NRZ magnetic tape reel or cassette.
MTIN.NRZ	NRZ magnetic tape reel or cassette without write permit.
MT9	Magnetic tape reel.
MT9IN	Magnetic tape reel without write permit.
MT9.PE	P.E. magnetic tape reel.
MT9IN.PE	P.E. magnetic tape reel without write permit.
MT9.NRZ	NRZ magnetic tape reel.
MT9IN.NRZ	NRZ magnetic tape reel without write permit.
CS	Magnetic tape cassette.
CSIN	Magnetic tape cassette without write permit.
CS.PE	P.E. magnetic tape cassette.
CSIN.PE	P.E. magnetic tape cassette without write permit.
CS.NRZ	NRZ magnetic tape cassette.
CSIN.NRZ	NRZ magnetic tape cassette without write permit.
DC	Any disk.
DM	Any mini-disk.
DM	Any Burroughs Standard mini disk.
DMII	BSMII disk.
DK	Cartridge disk.
DF.201I	201I disk.
DF.211	211 disk.
DP	Disk pack.

PB (List Printer Backup Files)

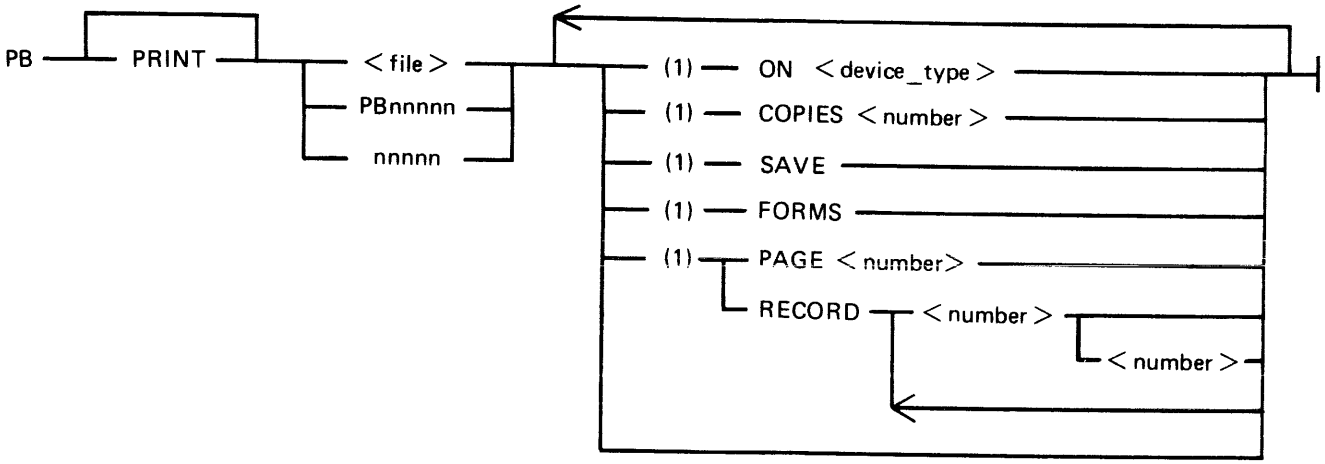
The CMS Printer Backup utility (PB) is an MPLII program which allows files created in accordance with the CMS Printer Backup facility to be printed by CMS users.

The utility has five possible formats:

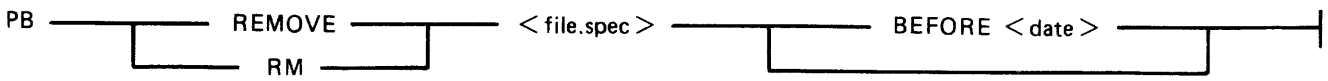
Format 1:



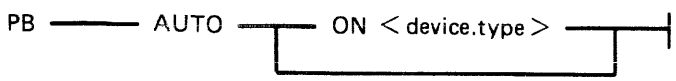
Format 2:



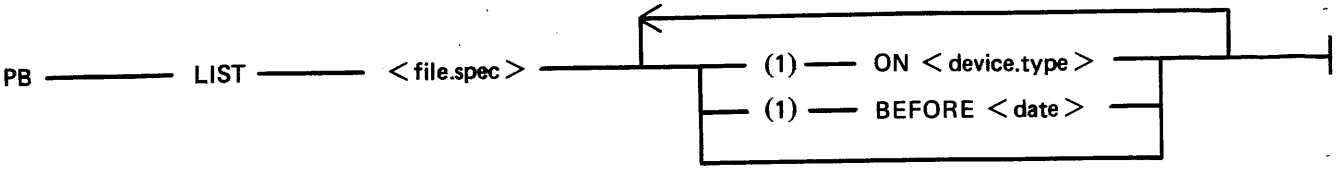
Format 3:



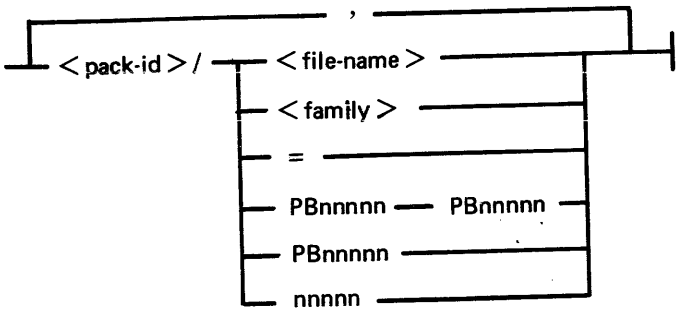
Format 4:



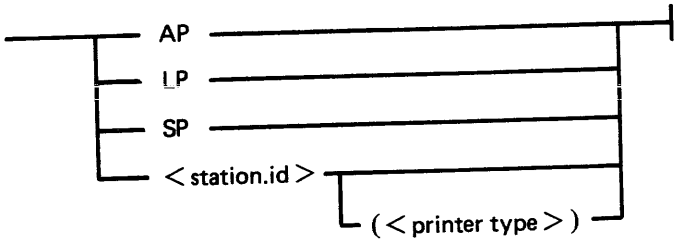
Format 5:



where < file.spec > is :



where < device.type > is :



AP 300 printers which do require special handling may be identified to PB in one of two ways:

1. If their <station.id> is of the form AP300XAxx where xx is any decimal number.
2: If <printer.type> follows the <station.id> where <printer.type> is (AP300).

<date> has the format of mm/dd/yy
where mm = month, dd = day and YY = year.

<number> is any decimal number.

PBnnnn stands for a file name in the range PB00001 through PB99999.

PB Initiating Message Parameters

<diskid>/	Identification of disk on which the backup file resides. Contains one through seven characters. If <diskid> is specified, the system status communicate is invoked, determining the name of the Printer Backup Designate Disk (PBDD) as specified by the SO SCL. If no PBDD is in use, default is to the system disk. The <diskid> must always be terminated with a slash.
PB<nnnnn>	Filename of printer backup file to be printed. Contains one to five decimal digits. If the user omits the letters "PB", they are appended to the front of the digits, forming the filename. In either case, the file must be of printer backup file-type to be printed by this utility. Leading zeros are not required in the PB<nnnnn> filename.

=	Specifies that all files identified by a "PBnnnnn" filename will be printed. Only a file which is of printer backup file-type can be printed by this utility.
<family>	Format is <prefix> =, this specifies all printer backup files beginning with <prefix>, for example, MYBACKUP=.
<filename>	Filename of the named printer backup file to be printed.
ON	Specifies the device on which the printing is to be done. "AP" means "Any Printer". It instructs the utility to use the first printer device of any type to become available. "LP" indicates that the line printer is to be used; "SP" indicates the serial printer. "Station-id" indicates that a datacomm printer is to be used. The PB utility recognizes two families of station-ids DCPRINTxx and AP300XAxx. Their use is described in DATACOMM PB. If no ON option is entered, default is to the line printer.
BEFORE <MM/DD/YY>	This is a qualifier to the <file spec> and restricts the execution of the utility to those files created on or before the specified date.
COPIES	Specifies a non-zero decimal number containing at most 65535. "COPIES cc" specifies the <number> of copies to be printed. The default value is one.
SAVE	Prevents the removal of the file from disk after printing is completed. If "SAVE" is not specified, removal occurs.
FORMS	Files requiring special forms and ordinary files may not be printed in the same run of PB. If FORMS option is specified, then files requiring special forms will be printed. In a "forms" run, only files requiring the same record and block size of printer will be printed. The first file in the <file spec> that PB finds will determine the record and block size for the run. In a "forms" run, the printer remains open all the time so that the user only has to AD the printer at the beginning of the run. After completion of the run, the printer is saved. In a "forms" run no headings, extra line feeds or top of forms are sent to the printer. The user is responsible for ensuring the paper is correctly aligned for the print.
PAGE	Specifies the page of the document at which printing is to begin. The PB utility begins from the start of the file, counting "Top-of-Forms" instructions: page numbers within the file text mean nothing. When it reaches the number of pages specified, (up to seven decimal digits may be used), it begins printing. The value of the page option may be from one to seven decimal digits. A value of zero is invalid.
RECORD	Selects specific parts of the printer backup file for printing by giving a starting record number and the number of records to be printed. More than one range of records may be specified, and overlapping of record number pairs may exist. NOTE: When there is no quantity in a record number pair, printing continues until end of file. This will only occur, however, in the last pair when an odd-numbered amount of record pairs is entered. The record option numbers, regardless of how many are entered, are always interpreted in pairs. An attempt to print from record 25 to endfile followed by printing from record 30 to endfile would, if given as "25 30", result instead in the printing of the thirty records beginning at

record 25. Start-number, quantity values may be from one to seven decimal digits. A value of zero is invalid. Record number pairs do not have to be entered in pairs of increasing value. The number of record number pairs is limited to 100.

REMOVE
RM

Causes the removal of the specified printer backup file(s).

This option replaces the CMS RB utility. If the file specification equals "=" all files starting with PB are removed. The utility offers the option to remove files before a specified date.

AUTO

Results in the printing of all available PB files. The utility then enters an "Idle" state until one or more PB files become available for printing.
To terminate the AUTO option, DS PB.

NOTE:
The AUTO option cannot be used on systems with no real-time clock.

LIST

Causes the printing of a list of all printer backup files of the form "PBnnnnn". The following information is printed for each file:

1. "Packid" (followed by that file's <packid>).
2. "External backup filename" (as it appears on disk)
3. "Internal backup filename" (as it appears in the header record)
4. "Print file name" (followed by the name used for the file by the program that generated it, for example "LISTPRT").
5. "Program Name" (followed by the name of the program that generated the file, for example, "LIST").
6. "Creation time HH:MM" (the time, in hours and minutes, at which the file was first opened; the range is 00:00 through 23:59).
7. "Creation date MM/DD/YY" (the date on which the file was first opened).
8. "Device kind" (followed by "LP" (line printer), "AP" (any printer) or "SP" (serial printer).
9. "Block size" (followed by the size of the block in bytes as specified in the file's header record).
10. "RECORD SIZE" (followed by the size of the record in bytes as specified).
11. "First record" (followed by the number of the record at which printing begins).
12. "FPB flags" (followed by verbal explanations of the flags, as follows:
    1) "Forms" or "No forms"
    2) "Labelled" or "Unlabelled"
    3) "Translation" or "No translation")

DATACOMM PB

PB supports any printer correctly connected through a data comm link and with a suitable entry in the NDL program, provided the printer satisfies the following conditions:

1. The printer is an ASCII printer.
2. The printer recognizes CR, LF and FF and requires no other special characters or combinations of characters for normal operation.
3. The printer can handle a line of up to 132 characters, that is, printers with line width of less than 132 will not be damaged by being sent a line of 132 characters.

PB treats remote printers differently from ordinary line printers in the following ways:

1. All tab commands will be replaced by two line feeds.
2. The banner heading "COMPUTER MANAGEMENT SYSTEM" is not printed and the print file information is shifted to the left of its current position on normal line printers. The effect of this is that the file information is central on the page of 80 character paper.

PB supports the AP journal printer. For this to occur, the AP must either be declared in the NDL as AP300XAxx where the last two characters are arbitrary (numbers only), or the (AP 300) option must be used in the initiating message.

PB operates in the following way:

1. PB will only go to end of job if a fatal data comm error is encountered.
2. If PB receives an error condition from the AP printer it produces the message:

   "ERROR CONDITION REPORTED BY <printer station id>".

   On systems without real time clocks, it may be found that an error condition occurs (for example, no paper), the operator loads more paper, the print resumes and then the message is produced. There is no sensible evasion of this problem.
3. If the AP printer is switched off, it is frequently not possible to detect and report this condition. In the case of a power off, data stored in the print buffers is lost (about 1 to 10 lines of output depending on buffer size). When the power is switched on, PB should continue the print.

Examples

```
PB PB02401
```

One (1) copy of file "PB02401", which resides on the disk identified by the system status communicate, will be printed. After printing, the file "PB02401" will be removed from the disk.

```
PB MYDISK/2401
```

One (1) copy of file "PB02401" residing on the disk named "MYDISK" will be printed. (Note that leading zeros are not required.) After printing, the file "PB02401" will be removed from "MYDISK".

```
PB MYDISK/2411 COPIES 22 SAVE
```

Twenty-two (22) copies of file "PB02411" residing on the disk named "MYDISK" will be printed. After printing, the file will remain on the disk.

```
PB 01238 RECORD 25 50 300 100 SAVE COPIES 5
```

Five copies of file "PB01238", residing on the disk identified by the system status communicate, will be printed. In each copy, the only records printed will be 50 records starting at record 25, followed by 100 records starting at record 300. After printing, the file will remain on disk.

```
PB MYDISK/PB412 ON LP COPIES 3
```

Three copies of file "PB00412" residing on the disk named "MYDISK" will be printed on a line printer. After printing, the file will be removed from disk.

PB MYDISK/=

All PB files residing on the disk "MYDISK" and belonging to the family "PBnnnnn" will be printed. The PB utility will then go to EOJ.

PB MYDISK/PB00719 RECORD 28 200 SAVE COPIES 10 ON SP

Ten copies of file "PB00719" residing on disk "MYDISK" will be printed on the serial printer. Each copy will contain only the 200 records which begin at the twenty-eighth record of the file. After printing, the file will remain on disk.

PB 02413 RECORD 170

One copy of file "PB02413", residing on the disk identified by the system status communicate, will be printed. This one copy will contain the records beginning at record 170 and continuing to the end of the file. After printing, the file will be removed from disk.

PB 2413 PAGE 15 SAVE ON LP COPIES 30

30 copies of file "PB02413", residing on the disk identified by the system status communicate, will be printed on the line printer. Each copy will contain all records found after the fifteenth "Top-of-forms" in the file. After printing, the file will remain on disk.

PB = ON SP

All printer backup files which belong to the "PBnnnnn" family and reside on the disk identified by the system status communicate will be printed on the serial printer. After printing, the files will be removed from disk.

PB AUTO

Prints all available "PBnnnnn" files residing on the disk identified by the system status communicate. Then enters an "Idle" state, printing each PB file that becomes available. Does not automatically go to EOJ.

PB LIST MYDISK/=

Produces a listing which identifies all "PBnnnnn" files residing on the disk "MYDISK". Lists the basic characteristics of each backup file.

PB 12 ON AP300XA45

The file PB00012 will be printed on the datacomm printer called AP300XA4 The PB utility will start a datacomm version of PB to communicate with the datacomm printer. The datacomm printer in this example is treated as an AP 300 journal printer.

PB 15 ON PRINTSTATDC

The file PB00015 will be printed on a datacomm printer called PRINTSTATDC. The datacomm printer in this example is NOT treated as an AP300 journal printer.

PB 6 ON PRINTSTATDC (AP300)

The file PB00006 will be printed on a datacomm printer called PRINTSTATDC. The datacomm printer in this example be will be treated as an AP 300 journal printer.

```
PB REMOVE 6, 12, 18
PB RM 6, 12, 18
```

Removes the file PB00006, PB00012, PB00018 from disk.

```
PB REMOVE =
PB RM =
```

Removes all printer backup files with the names in the range PB00001 through PB99999.

```
PB REMOVE AB=
PB RM AB=
```

Removes all printer backup files in the family AB

PB Messages

The banner is followed by an automatic page throw. Printing of the data contained in the backup file then begins, controlled by the forms-control data accompanying each record.

Files Requiring Special Forms

For files which have the "Special Forms" bit set, the banner is not printed.

The PB utility displays the following message to the user:

```
SPECIAL FORMS REQUESTED
```

The user must perform an "AD" to inform the utility whether to continue execution. The user enters:

```
AD <mix-number> LPx
```

NOTE

If a line printer is already opened, then the utility closes it with lock and opens the printer with the forms bit set. At this time, the MCP displays a message on the screen requesting that the user "AD" the device. When a special-forms job concludes, the printer is closed with lock.

Output Messages

Message	Possible Cause	Suggested Action
-FILE REQUIRES SPECIAL FORMS AND CAN ONLY BE PRINTED IF FORMS IS ENABLED	The specified file has to be printed on special forms.	Re-run PB with the FORMS option, for example PB 1 FORMS
<filename> REMOVED	Execution of PB RM was successful.	None.
-FILE NAME <filename> EXCEEDS 12 CHARACTERS	Self-explanatory.	Correct input and re-enter.
<filename> NOT REMOVED	Self-explanatory.	See the second message given for the reason.

(continued)

Message	Possible Cause	Suggested Action
-FILE DOES NOT REQUIRE SPECIAL FORMS AND CANNOT BE PRINTED WHEN FORMS IS ENABLED.	PB was started with the FORMS option but the specified file should not be printed on special forms.	Re-run the utility without the FORMS option.
-FILE REQUIRES DIFFERENT FORMS FROM PREVIOUS FILE(S) AND CANNOT BE PRINTED IN THE SAME RUN	PB was requested to print a group of files of which at least two files require different forms.	For one run of PB specify only files which use the same forms.
<filename> NOT LISTED	PB cannot satisfy the request	See the second message given for the reason.
FAMILY <familyname> NOT REMOVED	PB cannot remove the specified family.	See the second message given for the reason.
-PB CANNOT HANDLE MORE THAN 2804 FILES, LIMIT REACHED AT FILE <filename>	An attempt was made to print more than 2804 files in one run.	Run PB on smaller groups of files.
-FILENAME <filename> IS NOT A VALID FILE NAME	There are either more than 12, or invalid characters in the initiating message.	Correct input and re-enter.
FAMILY <familyname> NOT LISTED	PB cannot list the specified family.	See the second message given for the reason.
PARAMETER FOLLOWING KEYWORD <string> EXCEEDS <number>	The specified number is too high for the specified option.	Correct input and re-enter.
-FILE REQUIRES TRANSLATION WHICH IS NOT PROVIDED WHEN PRINTING AT A REMOTE PRINTER	Self-explanatory.	Print the file on a line-printer instead of a datacomm printer.
<filename> NOT PRINTED	PB cannot print the specified file.	See the second message given for the reason.
-HARDWARE ERROR DETECTED WHILE READING FILE <filename>	PB has detected a parity error on the specified file.	Check integrity of the disk.
IRRECOVERABLE DATACOMM ERROR	PB cannot run on the specified datacomm printer while it is not ready or in use by another program.	Check status of required printer using various MCS commands.
-<number> IS NOT A VALID NUMBER	Self-explanatory.	Correct input and re-enter.

(continued)

Message	Possible Cause	Suggested Action
-FILE HAS NO PRINTABLE RECORDS	The specified file is empty.	Check input for correct filename.
-CANNOT FIND RECORD <number> IN FILE	The specified record is beyond the end of the file.	None.
-RECORD <offset> <number> IS NOT WHOLLY WITHIN FILE	The starting record is in the file but the number of records to be printed plus the starting record ends outside the file.	Correct input and re-enter.
-DISK NAME <disk-id> EXCEEDS 7 CHARACTERS	The specified diskname is too long.	Correct input and re-enter.
-FAMILY NAME <family-name> NOT EXPECTED	The utility cannot list from family to family name.	Correct input and re-enter.
-<filename> IS NOT A VALID BACKUP FILE NAME	The utility cannot list from filename to filename.	Correct input and re-enter.
-RANGE OF BACKUP FILES FOR PRINTING MUST BE ON THE SAME DISK	The range of files given in the initiating message is not on the same disk.	Correct input and re-enter.
-FILE IS EMPTY	The specified filename does not contain any records for printing.	None.
-FIRST FILE IN RANGE <filename> IS GREATER THAN LAST FILE <filename>	Self-explanatory.	Correct input and re-enter.
-PROGRAM HAS BEEN DETACHED BY MCS	Self-explanatory.	None.
-HARDWARE ERROR DETECTED WHILE READING DISK <disk-id>	The utility has encountered a parity error on the specified disk.	Check integrity of the specified disk.
-<printer> IS NOT A VALID PRINTER TYPE	The printer type in the initiating message is not (AP 300).	Correct input and re-enter.
RANGE <disk-id>/<number> - <disk-id>/<number> NOT LISTED	The diskname may only appear once in the initiating message.	Correct input and re-enter.
RANGE <disk-id>/<number> - <disk-id>/<number> NOT REMOVED	The diskname may only appear once in the initiating message.	Correct input and re-enter.
-FORMS MAY NOT BE ENABLED FOR REMOTE PRINTERS	The FORMS option was specified while in the ON option. A datacomm printer is specified.	Correct input and re-enter.

(continued)

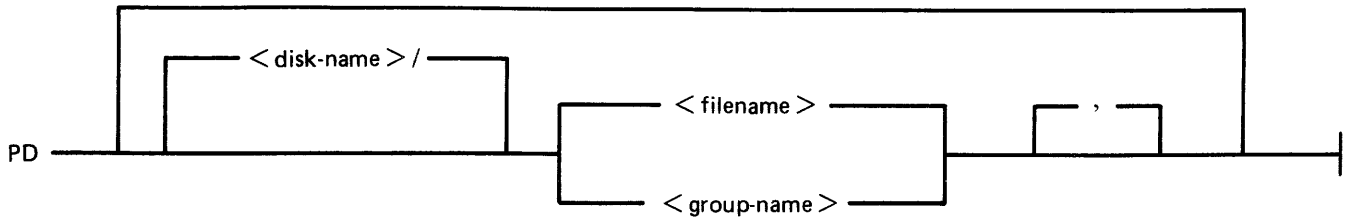
Message	Possible Cause	Suggested Action
-PRINT WIDTH IS GREATER THAN 132	Only files requiring printers of 132 characters per line will be printed.	None.
-CANNOT FIND PAGE <page number>	The specified page is not in the file.	None.
-FILE IS NOT A PRINTER BACKUP FILE	The filetype of the specified file is not suitable for PB.	None.
-AUTO OPTION REQUIRES A REAL TIME CLOCK ON THE SYSTEM	The AUTO option has been specified in the initiating message but there is no real time clock on your system.	Contact the Burroughs sales representative for information about a real time clock.
-<station name> IS NOT A VALID STATION NAME	There are non-CMS characters in the specified station name.	Correct input and re-enter.
-CREATED AFTER SPECIFIED DATE	The specified file is created after the specified date.	None.
RANGE <disk-id>/<filename> - <disk-id>/<filename> NOT PRINTED	The utility cannot list the specified range.	See the second message given for the reason.
-WRONG VERSION OF MCSWARM ON DISK	PB is not defined in the current MCSWARM file.	Copy a good version from backup disk or generate a new MSCWARM file using MCSGEN.
ERROR CONDITION REPORTED BY <station>	Self-explanatory.	Copy a good version from backup disk or re-run utility if necessary.

PD (Print Disk Directory)

(a function of SYS-SUPERUTL)

This utility allows the operator to verify the presence on disk of a particular file or a group of files.

Format:



Examples:

To find out if a particular file is on disk:
PD PR210
PD PR2/PR020

To find out if a group of files is on disk:
PD PR2/PR0=
PD PR3=

To find out if several different files or groups are on disk:
PD PR3= , PR2=
PD GL2GL0= , GL2/GL30= , GL250

To inquire about all files on disk:
PD =
PD PR2/=

Output Messages

Message	Possible Causes	Suggested Action
ON LINE <program-name>	File found on disk if single file requested.	None.
<group-name> ON <disk-name> CONTAINS:	Group of files found on disk.	None.
NOT ON LINE <program-name>	File not found on disk.	Check input (re-input if necessary). Check for correct disk.
REQUESTED FILES WERE NOT FOUND	Group not found on disk.	Check input (re-input if necessary). Check for correct disk.
END PD	Successful End of Job.	None.
<file-name> REQUIRES	Remainder of specified file	Supply appropriate disk.

(continued)

Message	Possible Causes	Suggested Action
OVERFLOW DISK <disk-name>	resides on another disk. PD cannot complete until appropriate disk is supplied.	
<group-name> on PSEUDO DISK <pseudo-disk-id> ON DISK <disk-id> CONTAINS	Group of files found on pseudo pack.	None.

Note: See "Common Utility Output Messages" for additional aid.

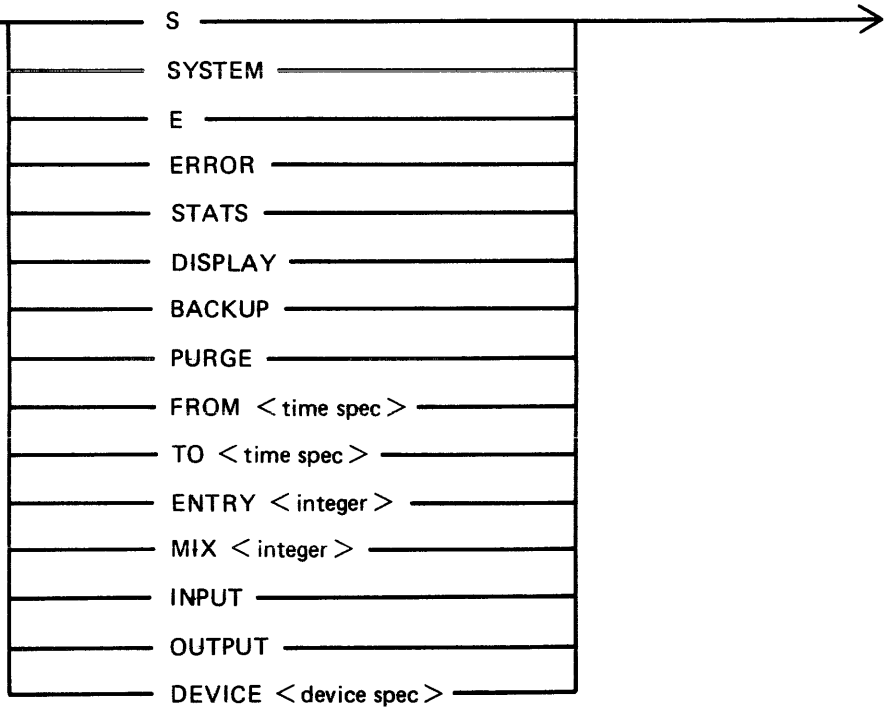
PL (Print Log Files)

PL lists the contents of log files present during any particular session. Attempts to list other filetypes will fail, since a check is made on the file entries themselves and these must be of a compatible format. Log files created on previous release levels are incompatible.

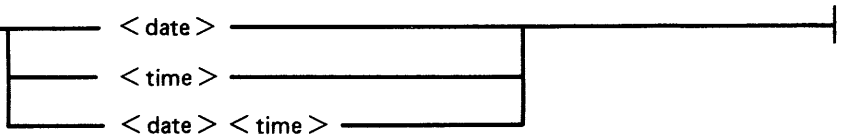
The utility incorporates optional facilities to analyze B 900 and B 1800 error entries, analyze statistics for 211 and BSMII disks and backup files.

Format

< option-list > is defined as :



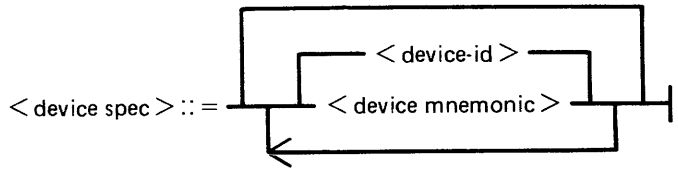
where < time spec > is defined as :



< date > :: = MM/DD/YY

< time > :: = HH : MM : SS

Format for DEVICE option:



<device-id> ::= form DFA, CTB etc.

<device mnemonic> ::= form DF, CT etc.

where:
<dvce-id> is of the form DFA CTB etc.
<dvce-mnemonic> is of the form DF CT etc.

NOTE

Duplicate entries of the same device will cause PL to go to End Of Job. This can be inadvertently caused by having both the device-id and the family mnemonic name specified. For example, having DFA and DF in the DEVICE option is a duplicate entry and will cause PL to terminate.

NOTE

The PURGE option can only be used on Secondary log files. That is SYS-LOG-HOLD, SYS-HLDjjjhh and SYS-MLGjjjhh. If used on a Primary log file (SYS-LOG-01 to SYS-LOG-04), the option will be ignored.

where <time spec> = <date>
or <date><time>
and <date> = MM/DD/YY
<time> = HH:MM:SS

The option "SYSTEM" (or "S") is specified to list only system messages from the log-file.

The option "ERROR" (or "E") is specified to list only error messages.

The option "FROM <time spec>" is used to list the logged message from the specified date and time. If time is not specified, then 00/00/00 is assumed.

The option "TO <time spec>" is used to list the messages up to that date and time. If time is not specified, then the last data and time in the log are assumed.

If the option "ENTRY" is used, the utility will print starting from the record number specified by the operator.

The "MIX number" option is used to print all messages related to specified mix (number(s)).

The "INPUT" and "OUTPUT" options allow the operator to print either input or output messages.

Any combination of SYSTEM, ERROR and STATS is permitted, but, if DISPLAY or BACKUP is requested, only ERROR and STATS are permitted.

The default options that are set are:

SYSTEM, ERROR and STATS messages;
INPUT and OUTPUT messages;
FROM 00/00/00 00:00:00;
TO <last date and time>;
ENTRY 1;
Output direct to printer.

All entries are displayed irrespective of their mix numbers. Any of these defaults can be reset at run time. If no real-time clock was available when the file was created, then no check will be made on the "time" portion of the operator input, and "N/A" will be printed under the "TIME" heading on the report. The default for DISPLAY/BACKUP is both ERROR and STATS.

STATS request will only output entries logged as statistics entries for BSMII and 211 disks, and only if those disks are used with the Standard Disk Interface (SDI) Common Controller.

If DISPLAY is requested, then error and statistics entries will be displayed on a screen with dimensions not less than 80 characters wide and 10 lines deep.

BACKUP will cause all output to go to a disk file of type "data". The format of this file will be exactly the same as if the output had gone directly to a printer. The file will have 120 byte records and three records per block. Page throws will be replaced by four space-filled records. The default size of the file will be 4096 records, but this can be altered using MODIFY. The internal file name and resultant output FID will be MLOGBACKUP; the destination disk will be the system disk unless MODIFYed; CLOSEMODE will be LOCK, but may also be MODIFYed.

Entries with multiple records will only have the record number and record contents displayed; all other columns will be blank, since the contents of these records will all be of the same type and created at the same time.

Only entries which conform to either the defaults, or operator input specifications will be displayed, all others will be ignored.

The range of values for ENTRY and MIX numbers are 1-65535 and 1-254 respectively. Checks at run-time are made on the values entered and messages issued if they are in error.

Examples

To print the contents of the log-file called SYS-LOG-HOLD:
PL SYS-LOG-HOLD

To print the error messages logged in the log-file called SYS-LOG-01:
PL SYS-LOG-01 ERROR

To print entries in SYS-LOG-HOLD file from record 100, related to mix number 12 from January 1, 1979 until latest date:
PL SYS-LOG-HOLD ENTRY 100 MIX 12 FROM 01/01/79

Output Messages

Message	Possible Causes	Suggested Action
<filename> REMOVED	Successful completion of PURGE option.	None.

(continued)

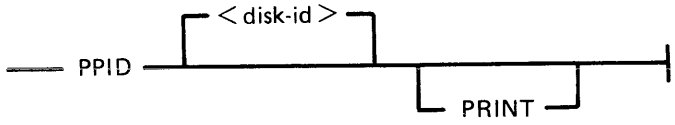
Message	Possible Causes	Suggested Action
-PARAMETER FOLLOWING KEYWORD <keyword> EXCEEDS <number>	Incorrect input to PL.	Correct input and re-enter.
-PARAMETER EXPECTED AFTER KEYWORD <keyword>	Incorrect input to PL.	Correct input and re-enter.
-OPTION <option> HAS ALREADY BEEN SPECIFIED	The specified option occurs more than once in the initiating message.	Correct input and re-enter.
-NO FILENAME SPECIFIED	There is no filename in the initiating message.	Correct input and re-enter.
UTILITY LIMIT REACHED WHEN RUNNING PL	The utility cannot handle the information in the specified log-file.	Check integrity of the specified file.
-HARDWARE ERROR DETECTED WHILE WRITING TO FILE <filename>	The utility has encountered an error while writing to the printer backup file, since BACKUP was specified in the initiating message.	Check integrity of the disk.
-HARDWARE ERROR DETECTED WHILE READING FILE <filename>	The utility has encountered an error while reading the specified file.	Check integrity of the disk.
NO LOG ENTRIES WERE FOUND TO CONTAIN REQUESTED INFORMATION	The specified logfile is empty.	None.
UNRECOGNISED DEVICE IN MAINTENANCE LOG ENTRY	Self-explanatory.	None.
-OUTPUT FILE <filename> TOO SMALL	The printer backup file is too small.	Use SO command to increase printer backup filesize.
HARDWARE ERROR DETECTED WHILE READING RECORD <number> OF <filename>	The utility has encountered a parity error on the specified file.	Check integrity of the specified file.
-SCREEN SIZE TOO SMALL	The screen on which PL is to run is smaller than 80 characters per line or less than 14 lines per screen.	Use FD command to change screen format or use another terminal with the right screen size.
-PLEASE ENTER END TO FINISH OR NEXT TO CONTINUE	Normal message to give the possibility to stop or to continue.	Enter AX <mix-no> END or AX <mix-no> NEXT.

(continued)

Message	Possible Causes	Suggested Action
<number> RECORDS WRITTEN TO BACKUP FILE <filename>	Self-explanatory.	None.
PRIMARY LOG FILE <filename> MAY NOT BE PURGED	An attempt was made to purge a primary log file.	None.
– LOG FILE WILL BE PURGED ONLY IF ALL ENTRIES ARE PRINTED	If PURGE option is specified and all entries are not printed, this message is displayed.	None.
ENTRY OPTION IGNORED FOR MAINTENANCE FILE <filename>	An attempt was made to list a maintenance log file with the ENTRY option.	Correct input and re-enter.
–DEVICE <device.type> HAS ALREADY BEEN SPECIFIED	The specified device occurs more than once in the initiating message.	Correct input and re-enter.
–SPECIFIED DEVICE <device.type> NOT RECOGNISED	The specified device is not known to the utility.	Correct input and re-enter.
–SPECIFIED DEVICE <device.type> WAS NOT FOUND IN HARDWARE CONFIGURATION TABLE	The specified device is not one of the devices of this configuration.	Correct input and re-enter.

PPID (Pseudo Pack Identifier Display – B 900 Only)

The Pseudo Pack Identifier Display utility may be used to list the Pseudo Pack Identifier Table (PPIT). This utility does not provide the *<file-name> option in the initiating message. The syntax for this utility is:



If PRINT is specified, the output will be listed on a printer, otherwise the output will be displayed on the Operator Display Terminal (ODT).

One of the following messages will be output for each used entry in the PPIT

disk-id – PHYS.UNITxx/READY	(Physical disk on line)
disk-id – PHYS.UNIT xx/LOCKED	(Physical disk on line)
disk-id – PHYS.UNIT xx/NOT READY	(Physical disk not on line)
disk-id – PSEUDO ON physical disk-id	(Restricted Pseudo disk)
disk-id – PSEUDO UNRESTRICTED	(Unrestricted Pseudo disk)
disk-id – PSEUDO UNRESTRICTED/SYSTEM	(System Pseudo disk)
xx is the logical unit number of	the fixed disk.

Physical unit entries are displayed in order of logical unit number, each being immediately followed by entries of pseudo packs restricted to that particular unit. Lastly, unrestricted pseudo pack entries are displayed.

Output Messages

Message	Possible Causes	Suggested Action
NO PSEUDO-PACK TABLE ON SYSTEM DISK	No pseudo pack identifier table was found on the system disk.	None.
– HARDWARE ERROR DETECTED WHILE READING PSEUDO-PACK TABLE ON DISK <disk-name>	An error was encountered while reading the PPIT.	None. Irrecoverable error.
– PHYSICAL DISK <disk-name> LOCKED	Another program is using the specified disk.	Wait until the other program stops using the specified disk, then re-run PPID.

Note: Refer to "Common Utility Output Messages" for additional aid.

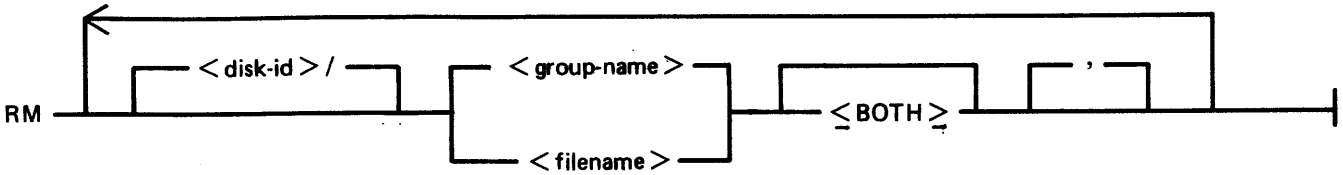
RM (Remove Files from Disk)

(a function of SYS-SUPERUTL)

This utility allows the removal of individual files and groups of files from disk. The disk areas associated with those files are returned to the available table.

If the utility detects that a keyfile is to be removed and the <BOTH> option has been specified, then it will remove both the keyfile and the associated data file if both are on disk. If <BOTH> is not specified, then only the keyfile will be removed.

Format:



Examples:

To remove a single file:

```
RM AR300
RM PR1/PR300
```

To remove a group of files:

```
RM AR=
RM INDISK2/IN3=
```

To remove several different groups and/or individual files:

```
RM IC230, IN076, INDISK1/IN2=
```

To remove a keyfile and associated data file:

```
RM PR200K <BOTH>
```

A request for the removal of a system file will cause the utility to output the following:

```
filename IS A SYSTEM FILE
AX "mix number"/RM ACPT
```

Then, to remove a system file:

```
AX mix-number OK
```

If the operator types any other sequence, the system file will not be removed.

Example:

```
RM NDL=
NDL.INTERP IS A SYSTEM FILE
```

12/RM ACPT
AX 12 OK

Output Messages

Message	Possible Causes	Suggested Action
<filename> REMOVED	File was removed.	None.
<FILENAME> NOT REMOVED – FILE NOT FOUND	Specified file was not removed because it is currently being used by the system.	Check input (re-input if necessary); check for correct disk.
<filename> NOT REMOVED – FILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	Specified file was not removed because it is currently being used by the system.	Wait until file is no longer in use, then re-input.
<filename> NOT REMOVED – SYSTEM FILE	Specified file was not removed because it is a "system" file (for example MCP, an interpreter).	If a file is to be removed, type "AX <mix-no>/RM filename OK". If file is not to be removed, type AX <mix-no>/RM NO.
INDEXED FILE PAIR <filename> <filename> REMOVED	Keyfile and associated data file were removed.	None.
INDEXED FILE PAIR <filename> <filename> NOT REMOVED	Keyfile and associated data file were not removed. This message is followed by a second message giving the reason.	Check input (re-input if necessary); or check for correct disk.
INDEXED PAIR <filename> <filename> NOT REMOVED	Specified keyfile and data file were not removed because at least one is currently being used by the system.	Wait until files are no longer in use, then re-input.
END RM	Normal EOJ message.	None.
<filename> IS A SYSTEM FILE	Specified file is a system file and is protected.	Enter "AX <mix-no> OK" to remove the file or AX <mix-no> to leave it on disk.
– FILE IS NOT A KEYFILE	<BOTH> was specified for a file which is not a keyfile.	None.
– HARDWARE ERROR DETECTED WHILE READING KEYFILE FILE <filename>	The utility has encountered a parity error while reading the file.	Check integrity of the disk.

SCR (System Confidence Routine)

The SCR utility is used to establish a level of confidence in the overall performance of the software/hardware interface of a CMS system. It is not intended to perform the function of an MTR program in rigorously testing isolated hardware components, nor is it intended as a proof of correctness for system software.

The SCR operates under MCP control, and relies on the MCP for basic error detection. In addition, SCR makes certain comparisons between actual results and expected results, and performs checks on output results.

Operator input required is minimal, and is restricted to initiation of the utility and replies to prompts displayed immediately thereafter. Information concerning errors detected and program performance is displayed at the SPO throughout execution of the SCR. The SCR may be DS'ed by the user at any time. The user specifies the number of records to be read.

The integrity of disk files is maintained: all disk files created by SCR are CLOSEd with RELEASE. Thus, when the utility goes to EOJ, those files do not remain on disk.

The SCR utility is written in MPL. One of its major subsections (CPU.IO) ZIPs to two independent programs, one of which is written in MPL and the other in COBOL.

The utility comprises four major subsections. The user can specify which of these subsections are to be performed for any given run of SCR. To assist him in making his selection, a HELP display is provided. The four subsections are:

1. DCR — Disk Confidence Routine. This tests CMS disk input/output operations.
2. CPU.IO — This subsection tests the interface between the MCP, IO, the MPL and COBOL Interpreters by ZIPping to an MPL program which outputs the result of calculations to disk, and by ZIPping to a COBOL program which prints the edited results of its calculations.
3. MT.IO — This subsection writes files of binary information to user-selected magnetic devices, reads these files back, and compares the output and input records.
4. OTHER.IO — This subsection outputs files of information to user-selected non-magnetic devices, reads the files back and checks them when possible. Currently, this subsection only applies to printer, and therefore relies on the user to perform a visual check of the information output.

Any or all of these subsections may be selected by the user for a given run of the utility. The order of performance is determined by the order of selection. The number of iterations for each of the subsections may be specified by the user; all iterations of a given subsection are performed before execution of another subsection is begun.

It is recommended that care is taken when specifying the size of file for the DCR WRITE-INDEX test. Since the records are written and read randomly, access time becomes proportionately greater as the filesize increases.

Operating Instructions

Enter: SCR

Prompt 1 is displayed:

Prompt 1:

"ENTER TEST NUMBERS OR ENTER HELP"
"ENTER SPACE FOR DEFAULT VALUES OR 0 TO TERMINATE SCR"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> HELP <PROMPT 1>	ENTER (ON 1 LINE) NUMBERS FOR TESTS WANTED 1 = DISK CONFIDENCE ROUTINE (DCR) 2 = CPU/IO INTERACTION TEST (CPU.IO) 3 = MAGNETIC DEVICE EXERCISER (MT.IO) 4 = NON-MAGNETIC PERIPHERAL EXERCISER (OTHER.IO) ALL = ALL TESTS WANTED ENTER SPACE FOR DEFAULT VALUES OR 0 TO TERMINATE SCR DEFAULT VALUES ARE: TESTS 1,2,4 DCR – ALL SUBTESTS, 1 DISK INDEX FILESIZE 100 RECORDS READING 1000 RANDOM SECTORS CPU.IO – 1 ITERATION OTHER.IO – 1 ITERATION <PROMPT 1>		Select tests required and input test numbers, or ALL.
AX <mix-no> <empty>	<PROMPT 2.1>	The default values have been chosen.	Enter disk-name.
AX <mix-no> 0	SCR TERMINATED BY USER	Utility has been terminated.	None.
AX <mix-no> <test list>	INVALID INPUT CHARACTER <PROMPT 1>	Test list input did not consist of valid test numbers or "ALL".	Re-input test list, or terminate the utility.
	SCR INITIATING MESSAGE ERROR DUPLICATE SCR SUBTEST SCR ABORTED	Self explanatory.	Re-start SCR utility, taking care to only enter one of each test number.
	DUPLICATE TEST NO.	A digit has been repeated in the test list entered.	Re-enter test list, or terminate the utility.
	<PROMPT 2>	Test 1 (DCR) or "ALL" has been selected.	Refer to possible replies to Prompt 2.

(continued)

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
	<PROMPT 3>	Test 2 (CPU.IO) or "ALL" has been selected.	Refer to possible replies to Prompt 3.
	<PROMPT 4>	Test 3 (MT.IO) or "ALL" has been selected.	Refer to possible replies to Prompt 4.
	<PROMPT 5>	Test 4 (OTHER.IO) or "ALL" has been selected.	Refer to possible replies to Prompt 5.

Spaces act as a delimiter but are ignored.

If the reply contains the string "ALL", any valid digits contained in the reply are ignored.

Prompt 2:

"ENTER NUMBER OF DCR ITERATIONS – DEFAULT = 1"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	<PROMPT 2.1>	DCR subsection will be performed once.	
AX <mix-no> <decimal-no>	<PROMPT 2.1>	DCR subsection will be performed <decimal-no> of times.	None.

PROMPT 2.1

"ENTER DISK.ID TO CONTINUE.
ENTER SPACE TO TERMINATE DCR"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	DCR TERMINATED BY USER	DCR has been terminated.	None.
AX <mix-no> <disk-name>	FETCH VALUE = <hex-no.> DISK <disk-name> NOT AVAILABLE DCR ABORTED	The MCP has been unable to open <disk-name>.	None.
	FETCH VALUE = <hex-no.> READ ERROR ON <disk-name> LABEL DCR ABORTED	MCP input error on reading the disk label.	None.
	ILLEGAL PARAMETER LIST 0000000 <PROMPT 2.1>	The default System Disk identifier has been used.	Use the System Disk, as the disk name of the actual disk.
	DISK <disk-name> IS A PSEUDO-DISK <PROMPT 2.1>	DCR does not use PSEUDO-DISK disk-names.	Re-enter a non pseudo-disk disk-name.
	<PROMPT 2.2>	<disk-name> has been opened successfully and disk-label has been read.	Proceed to the instructions for PROMPT 2.2.

PROMPT 2.2:

"SECOND DISK TO BE TESTED? <Y or N>"

If the answer to prompt 2.2 was "Y", denoting that a second disk is to be tested, prompt 2.1 is displayed. Refer to the following table for possible replies.

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <neither Y nor N>	REPLY NOT "Y" OR "N" <PROMPT 2.2>	Reply not "Y" or "N".	Enter Y or N.
AX <mix-no> N	SECOND DISK NOT WANTED <PROMPT 2.3>	No second disk was specified.	Continue.
AX <mix-no> Y	<PROMPT 2.1>	Second disk is to be tested.	Enter disk-name or space.

Prompt 2.2 (continued) – Possible replies if a second disk has been requested:

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	DCR TERMINATED BY USER	DCR has been terminated.	None.
AX <mix-no> <disk-name>	FETCH VALUE = <hex-no.> DISK <disk-name> NOT AVAILABLE DCR ABORTED	The MCP has been unable to open <disk-name>.	None
	FETCH VALUE = <hex-no.> READ ERROR ON <disk-name> LABEL DCR ABORTED	MCP input error on reading the disk label.	None.
	ILLEGAL PARAMETER LIST 0000000 <PROMPT 2.1>	The default System Disk identifier has been used.	Use the System Disk as the disk name of the actual disk.
DISK <disk-name> IS A PSEUDO-DISK <PROMPT 2.1>		DCR does not use PSEUDO-DISK disk-names.	Re-enter a non pseudo-disk disk-name.
	CAPACITIES ON 2ND DISK NOT SAME AS ON 1ST DCR ABORTED	Indicates inequalities in the sizes of the SYSMEM files or in the number of sectors per buffer between the two disks.	None.
	<PROMPT 2.3>	Disk capacities have been compared and are equal, disk-name has been successfully opened and the disk-label has been successfully read.	Proceed to the operating instructions for PROMPT 2.3.

PROMPT 2.3:

"ENTER TEST NUMBERS OR ENTER HELP
ENTER SPACE TO TERMINATE DCR"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	DCR TERMINATED BY USER	DCR has been terminated.	
AX <mix-no> HELP	ENTER ON 1 LINE NUMBERS FOR TESTS WANTED 1 = READ.ALL 2 = OPEN.FILES 3 = READ.CYLINDERS 4 = WRITE.INDEX 5 = READ.RANDOM ALL = ALL TESTS WANTED SPACE = TERMINATE DCR <PROMPT 2.3>	Help function has been requested.	Select tests required and input, else terminate the utility.
AX <mix-no><non-numeric entry>	NON-NUMERIC INPUT CHARACTER <PROMPT 2.3>	No valid test numbers input.	Re-input correctly.
AX <mix-no><test number list>	<PROMPT 2.3>	Test numbers not in the range 1-5 and not "ALL".	Re-input correctly.
	DUPLICATE TEST NO.	A test number has been repeated.	Re-enter correctly.
	SCR INITIATING MESSAGE ERROR DUPLICATE DCR SUBTEST SCR ABORTED	Self-explanatory.	Re-start SCR utility, taking care to only enter one of each test number.
	<PROMPT 2.4>	Test 1 or "ALL" tests have been selected.	Proceed to input for PROMPT 2.4.
	<PROMPT 2.5>	Test 4 or "ALL" tests have been selected.	Proceed to input for PROMPT 2.5.

(continued)

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
	<PROMPT 2.6>	Test 5 or "ALL" tests have been selected.	Proceed to input for PROMPT 2.6.

If tests 1, 4, or 5 (or ALL) have not been requested, the utility will begin to execute test 2 and/or test 3 if selected.

PROMPT 2.4:

"MULTI-SECTOR BUFFERING WANTED? <Y or N>"

This prompt is displayed if test 1 or ALL is requested.

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <either Y nor N>	REPLY NOT "Y" OR "N" <PROMPT 2.4>	Reply was not Y (for yes) or N (for no).	Enter Y or N.
AX <mix-no> Y		When performing test 1 (READ.ALL), a buffered read will be invoked, with 32 sectors being read in each read operation.	None.
AX <mix-no> N	MULTI-SECTOR BUFFERING NOT WANTED	When performing READ.ALL (test 1), single sector reading will be invoked, thus maximizing head movement.	None.

PROMPT 2.5:

"ENTER NUMBER OF RECORDS IN INDEX.FILE OR SPACE TO CANCEL TEST"

This is displayed if test 4 or ALL has been requested. The number specified is used as the size of the index file written and read in test 4.

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	Test 4 (WRITE.INDEX) will not be performed.		
AX <mix-no> <decimal no>	REQUESTED IX FILE SIZE EXCEEDS DISK CAPACITY <PROMPT 2.5>	n is greater than the number of sectors on the disk.	Enter a number smaller than or equal to the number of sectors on the specified disk.

(continued)

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no><non-numeric entry>	NON-NUMERIC INPUT CHARACTER <PROMPT 2.5>	The reply was not a decimal number.	Re-input the desired (decimal) number of records.

PROMPT 2.6:

"ENTER NUMBER OF RANDOM SECTORS TO BE READ FROM EACH DISK OR SPACE TO CANCEL TEST"

This prompt is displayed if test 5 or ALL has been specified.

The number entered must be less than 16 777 216 and will be used as the number of sectors to be read in test 5 (READ.RANDOM).

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>		READ.RANDOM (Test 5) will not be performed.	
AX <mix-no> <decimal no>	INPUT NUMBER > 16777215 <PROMPT 2.6>	Self explanatory.	Re-input number smaller than 16777216.
AX <mix-no><non-numeric entry>	NON-NUMERIC INPUT CHARACTER <PROMPT 2.6>	Self explanatory.	Re-input desired (decimal) no.

TEST 1 (READ.ALL) may output any of the following error messages:

Message	Possible Cause	Suggested Action
READING ALL n SECTORS ON <disk-name> MULTI-SECTOR BUFFERED (or) BUFFERED BY SECTOR	Number of records in SYSMEM, for each disk being tested, is obtained and that number displayed in this message (n). The number of records in SYSMEM may be fewer than the number of physical sectors on a disk.	None.
MULTI-SECTOR BUFFERING NOT AVAILABLE	If OPEN fails on a 32 sector buffered SYSMEM, this message is displayed and input (for both disks, if two are used) will be via the single-sector buffers.	None.

(continued)

Message	Possible Cause	Suggested Action
ERROR ON <disk-name> MULTI-SECTOR READ	If multi-sector buffering is being used and the MCP reports an error when a given track is input to the buffer, this message is displayed. DCR re-reads the track via the single sector buffer in an attempt to isolate the bad sector.	None.
BUT NO ERRORS WHEN SECTORS READ	The MCP has reported no errors when records of the track have been re-read.	None.
READ ERROR ON <disk-name>,<sector address>	The MCP has reported a read error, other than when a multi-sector buffer is first filled and DCR displays this message.	None.
NO ERROR ON MULTI-SECTOR READ	Multi-sector buffering is being used and a read error was not reported when the buffer was first filled.	None.
LISTED AS BAD IN AVAIL.TABLE (or) NOT LISTED AS BAD IN AVAIL.TABLE	The available table has been checked, and appropriate message output. The count of sectors for which read errors have been reported is maintained.	None.
n SECTORS READ n BAD SECTORS READ ON <disk-name>	After every 10% of sectors on each disk being used have been read, DCR displays this message.	None.
READING ALL SECTORS ABORTED	More than 300 read errors have been reported on one disk – READ.ALL will be terminated if this message is output.	None.
READING ALL SECTORS COMPLETED. <n> SECTORS READ ON <disk-name>	All sectors have been read and, for each disk, a summary of read errors reported. DCR proceeds to next specified test.	None.

Test 2 (OPEN.FILES) may display any of the following messages:

Message	Possible Cause	Suggested Action
OPENING UP TO 20 FILES ON <disk-name>	This is displayed for each disk.	None.
ERROR READING FDNL ON <disk-name> OPEN. FILES TRUNCATED	The file directory name-list (FNDL) is searched to obtain a maximum of 20 file identifiers. This error is reported while reading the FDNL. DCR will attempt to open only those files whose identifiers were obtained before the error was reported.	None.
FILE <file-name> ON <disk-name> OPENED	DCR attempts to open conditionally each of the files. Names were obtained from the FDNL's. This message indicates that the open is successful and closes the file and removes it without performing any other I/O operations.	None.
<file-id> ON <disk-name> DID NOT OPEN	Attempt to open a file is unsuccessful.	None.
NO FILES ON <disk-name>	The address of an FDNL is incorrect or the FDNL is corrupt in a way that does not produce MCP read error reports.	None.
<n> OPENS TRIED ON <disk-name>	Attempts have been made to open all possible (up to 20) files on a given disk.	None.
<n> SUCCESSFUL, <n> UNSUCCESSFUL	Summary of the results of all possible opens (up to 20).	None.

Test 3 (READ.CYLINDERS) may display any of the following messages:

Message	Possible Causes	Suggested Action
READING <n> CYLINDERS ON <disk-name>	For each disk DCR obtains the number of cylinders on the disk and displays this message.	None.
READ ERROR ON <disk-name>, SECTOR <sector address>	If the MCP reports an error on a read, DCR displays this message.	None.

After every 10% of the sectors (cylinders) read on each disk, DCR displays:
"<n> SECTORS READ"
and from each disk:

"<n> BAD SECTORS READ ON <disk-name>"
n being a count maintained of MCP error reports during this test.

When all cylinders have been read, DCR displays:
"READING CYLINDERS COMPLETED"
and for each disk a summary of the read errors reported by the MCP:
"<n> BAD SECTORS READ ON <disk-name>"

It then proceeds to the next test specified by the user.

Test 4 (WRITE.INDEX) may display any of the following messages:

Message	Possible Cause	Suggested Action
WRITING INDEX FILE ON <disk-name>	This is displayed for each disk being tested.	None
FETCH VALUE= <hex.no> OPEN FAILED ON <disk-name> WRITE.RANDOM ABORTED	DCR attempts to open an arbitrarily named ("YELNOC") index file on each disk. If the open fails, this message is displayed and the test is abandoned.	None
FETCH VALUE= <hex.no>	This is displayed by DCR if the MCP reports a conditional failure on any write operation.	None.
DATA SPACE NOT AVAILABLE ON <disk-name>	There is not enough user disk for the utility to operate.	Remove some back up files to make room and re-execute the utility.

If the MCP reports an error on any write operation during WRITE.INDEX, the following message is displayed:

"WRITE ERROR ON <disk-name>, KEY = <key-value>

The utility compares the sector number against the list of bad sectors in the disk's available table, displaying one of the following messages accordingly:

"LISTED AS BAD IN AVAIL.TABLE"
or
"NOT LISTED AS BAD IN AVAIL.TABLE"

Message	Possible Causes	Suggested Action
WRITING INDEX FILE TRUNCATED	A conditional failure or an error on any write operation. No more records are written (to either file if two disks are being tested).	AX <mix-no><empty>
READ ERROR ON <disk-name>, KEY = <key-value>	DCR reads back index files in the order in which they were written. If the MCP reports	None.

(continued)

Message	Possible Causes	Suggested Action
	an error on any read operation, this message is displayed by DCR, which maintains a count of bad sectors for each disk.	
READING INDEX ABORTED	More than 300 read errors are reported on a file, proceeds to next subtest requested.	None.
DATA MISMATCH ON <disk-name> RECORD <hex.number>	When each record is read in, the data is compared against the value determined by the record's ordinal number. If the values are not identical, DCR displays this message.	None.
WRITING INDEX FILE FINISHED	All records have been written.	None.
READING INDEX COMPLETED	All records have been read.	None.
<n> BAD SECTORS READ ON <disk-name>	Summary of read error reports.	None.
<n> DATA MISMATCHES ON <disk-name>	Summary of data mismatches.	None.

Test 5 (READ.RANDOM) may display any of the following messages:

Message	Possible Cause	Suggested Action
READING <n> RANDOM SECTORS ON <disk-name>	This is a response in reply to number of random sectors requested in PROMPT 2.6.	None.
READ ERROR ON <disk-name>, SECTOR <sector address>	If the MCP reports a read error, DCR outputs this message and maintains a count of read-errors.	None.
READING RANDOM SECTORS ABORTED	If more than 300 errors are reported from a disk, DCR displays this message and abandons the test.	None.
<n> SECTORS READ	After every 10% of the specified number of sectors have been read from each disk being tested, this message is displayed.	None.
<n> BAD SECTORS READ	After every 10% of the specified number of sectors have been read from each	None.

(continued)

Message	Possible Cause	Suggested Action
	disk being tested, this message – a summary of the error error count n – is output.	
READING RANDOM SECTORS COMPLETED	When the total number of requested sectors have been read from each disk, this message is displayed.	None.
<n> BAD SECTORS READ ON <disk-name>	Total number of bad sectors read.	None.

Prompt 3:

"ENTER NUMBER OF ITERATIONS FOR CPU.IO – DEFAULT = 1."

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>		The CPU.IO test will be performed once.	None.
AX <mix-no> <decimal no>		CPU.IO will be performed the number of times specified, up to a maximum of 99 times. Only the first two digits are read.	None.
	INVALID INPUT CHARACTER <PROMPT 3>	A non-decimal character was input.	Re-input number of iterations.

Prompt 4:

"ENTER NUMBER OF MT DEVICES – DEFAULT = 1."

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	<PROMPT 4.1>	SCR will write to and read from one magnetic tape reel.	Refer to possible replies to Prompt 4.1.
AX <mix-no> <decimal no>	<PROMPT 4.1>	SCR will write to and read from the number of magnetic tape reels specified – up to a maximum of five reels. Only the first character input is read.	Refer to possible replies to Prompt 4.1
	INVALID INPUT CHARACTER <PROMPT 4>	An invalid non-decimal has been input.	Re-input no. of reels required.

PROMPT 4.1:

"ENTER NUMBER OF CS DEVICES – DEFAULT = 0"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	<PROMPT 4.2>	SCR has used the default value of zero, and will not write to or read from cassette.	Refer to possible replies to Prompt 4.2.
AX <mix-no> <decimal no>	<PROMPT 4.2>	SCR will attempt to write to and read from the specified number of cassettes, up to a maximum of five cassettes.	Refer to possible replies to Prompt 4.2
	INVALID INPUT CHARACTER	Self explanatory	Re-enter (decimal) no.
	ERROR: MORE THAN 5 DEVICES REQUESTED. <PROMPT 4>	The total number of magnetic tape reels and cassettes requested, specifically or by default, exceeds five.	Re-input the parameters for MT and CS devices, taking care that the total does not exceed five.
	ERROR = TOTAL MAGNETIC DEVICES = 0	Zero devices (magnetic tape reels and cassettes) were	Re-enter the parameters for MT and CS devices, taking care that the

(continued)

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
	<PROMPT 4>	explicitly requested.	total is greater than zero.

PROMPT 4.2:

"ENTER NUMBER OF RECORDS IN FILE – DEFAULT = 256"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	<PROMPT 4.3>	Each file written to and read from magnetic tape reels and/or cassettes will contain 256 records.	Refer to possible replies to Prompt 4.3.
AX <mix-no> <decimal no>	<PROMPT 4.3>	The size of all magnetic media files will be the (3-digit) decimal number specified.	Refer to the possible replies to Prompt 4.3.
	ERROR : FILESIZE =0 <PROMPT 4.2>	Self explanatory.	Re-enter number of records greater than zero.
	INVALID INPUT CHARACTER <PROMPT 4>	Self explanatory	Re-enter number.

PROMPT 4.3:

"ENTER NUMBER OF ITERATIONS – DEFAULT = 1"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	READY SCRATCH TAPES ON <m> MAGNETIC TAPE DRIVES, <n> CASSETTE DRIVES	The MT.IO subsection will be performed once only.	Make ready the specified number of tapes on their drives.
AX <mix-no> <decimal no>	READY SCRATCH TAPES ON <m> MAGNETIC TAPE DRIVES, <n> CASSETTE DRIVES	The MT.IO subsection will be performed the number of times specified.	Make ready the specified number of tapes on their drives.
	INVALID INPUT CHARACTER <PROMPT 4>	Self explanatory.	Re-enter number.

Prompt 5:

"ENTER NUMBER OF OTHER.IO ITERATIONS – DEFAULT – 1"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> <empty>	<PROMPT 5.1>	The OTHER.IO subsection will be performed once only.	Refer to possible replies to Prompt 5.1.
AX <mix-no> <decimal no>	INVALID INPUT CHARACTER <PROMPT 5>	The input string contains a non-decimal character.	Re-input the required no. of iterations correctly.
	<PROMPT 5.1>	The OTHER.IO subsection will be performed the number of times specified by the first two digits of the input string (that is, up to 99 times).	Refer to possible replies to Prompt 5.1.

PROMPT 5.1:

"IS PRINTER TO BE TESTED? <Y OR N>"

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> Y	<PROMPT 5.2>	SCR will write files (the number of files being specified by the reply to Prompt 5) of a pre-determined content to the first printer available at execution time.	Refer to possible replies to Prompt 5.2.
AX <mix-no> N	<PROMPT 5.2>	The printer test will not be performed.	Refer to possible replies to Prompt 5.2.
AX <mix-no> <not "Y" or "N">	REPLY NOT "Y" OR "N" <PROMPT 5.1>	Self explanatory.	

PROMPT 5.2:

"IF CARD PUNCH AND READER TO BE TESTED, ENTER Y."

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> Y	CARD PERIPHERALS NOT IMPLEMENTED	No card tests will be performed.	None.
AX <mix-no> N		No card tests will be performed.	None.
AX <mix-no.> <not "Y" or "N">	REPLY NOT "Y" OR "N" <PROMPT 5.2>	Self explanatory	Re-enter either "Y" or "N".

Execution Details

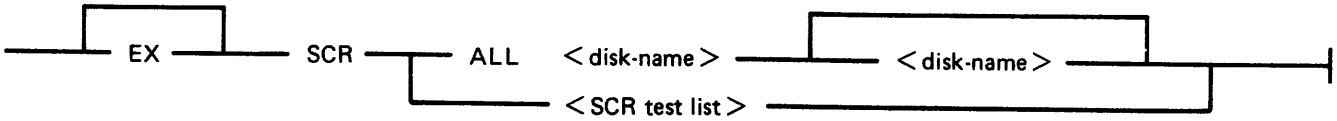
"SCR" is followed by an initiating message. This message can take two forms:

1. SCR parameters (Example 2), or
2. names of star-files containing SCR parameters (Example 3).

An SCR parameter is the name of an SCR subtest wanted by the user, followed by parameters to that subset. The available SCR subtests are:

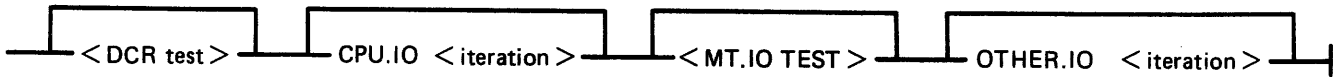
DCR
CPU.IO
MT.IO
OTHER.IO

The first parameter to each SCR subtest (that is, immediately following its name) must be a decimal integer specifying the number of times the subtest is to be performed. This number may not be "0". In the case of CPU.IO and of OTHER.IO, the number of iterations is the only parameter to the subtest.



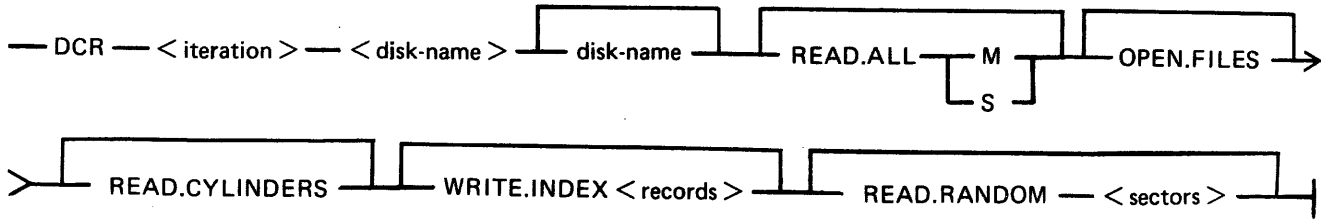
iteration = the number of iterations of the subsection.

Where SCR test list is:



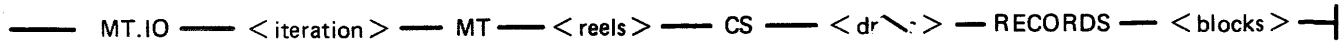
Where DCR test is:

or



records = The number of (data) records in the index file.
sectors = The number of sectors to be read randomly.

Where MT.IO test is:



reels = The number of magnetic tape reels to be tested.
drives = The number of cassette drives to be tested.
blocks = The number of records to be written.

For MT.IO, the number of iterations must be followed by "MT" followed by the number of magnetic reels to be tested followed by "CS" followed by the number of cassettes to be tested. The total of reels and cassettes must be greater than 0 and no greater than 5.

For DCR, the number of iterations must be followed by the name(s) of the disk(s) to be tested. There must be at least one and no more than two disk names, but the same System Disk identifier "0000000" are invalid.

NOTE

Omission of a disk name may result in the syntactic element immediately following it being mistaken for a disk name; consequent error messages may therefore seem misleading.

Each parameter following the disk name(s) is the name of a DCR subtest wanted by the user followed by parameters (if any) to that DCR subtest. The available DCR subtests are:

READ.ALL
OPEN.FILES
READ.CYLINDERS
WRITE.INDEX
READ.RANDOM

Repetition of DCR subtest names is invalid.

DCR subtests OPEN.FILES and READ.CYLINDERS have no parameters.

The DCR subtest name "READ.ALL" must be followed by a parameter indicating the type of buffering wanted:

"M" for multi-sector buffering,
"S" for single-sector buffering.

The DCR subtest name "WRITE.INDEX" must be followed by a decimal number specifying the number of (data) records in the index file. The number 0 is invalid.

The DCR subtest name "READ.RANDOM" must be followed by a decimal number specifying the number of sectors to be read. The number 0 is invalid.

If the user wishes all DCR subtests to be performed once only, with default parameters, he may follow "DCR" with "ALL" followed by the disk name(s), omitting the DCR subtest names and parameters (Examples 4 and 6).

If the user wishes all SCR subtests (including all DCR subtests) to be performed once only, with default parameters, he may follow "SCR" with "ALL" followed by the name(s) of the disk(s) to be tested in DCR, omitting the SCR subtest names and parameters (Example 5).

Examples

1. SCR

2. EX SCR DCR 1 DISK1 DISK2 READ.ALL M OPEN.FILES READ.CYLINDERS WRITE.INDEX 100 READ.RANDOM 1000, CPU.IO 1, MT.IO 1 MT 1 CS 0 RECORDS 256, OTHER.IO 1

5. SCR ALL DISK1 DISK2

6. SCR DCR ALL DISK1 DISK2

Note: Examples 2, 3, 4 and 5 produce the same effect.

Default values are:

```
number of iterations (for all SCR and DCR subtests = 1)
MT.IO : MT = 1
        CS = 0
        RECORDS = 256
DCR : WRITE.INDEX : <index.file size> = 100
        READ.RANDOM : <records> = 1000
        READ.ALL : <buffering> = M
```

NOTE

Example 6 has the same effect as was obtained under pre-3.04 release levels by executing DCR and making the appropriate replies to the various prompts.

DCR (Disk Confidence Routine)

The DCR subsection is used to establish a level of confidence in the ability of the CMS system hardware/software interface to service disk I/O. In doing so, it exercises the system as fully as possible and provides as much information as possible and practical about faults detected. With respect to hardware, DCR is designed to accommodate any CMS disk and attempts to maximize disk head movement, demands on the common electronics module, and demands on the directory search hardware.

The DCR subsection includes five separate tests. These are:

READ.ALL
OPEN.FILES
READ.CYLINDERS
WRITE.INDEX
READ.RANDOM

All, or a subset of these, may be selected by the user in a given run of the subsection. Integrity of files already existing on disks being tested is maintained: except for one test, all I/O operations are read only, and all files created by the subsection are closed and removed at or before EOJ.

For any run of DCR one or two disks may be used. The use of two disks is intended for dual disk drives and thus maximizes demands on the common electronics module. Both disks (when two are used) must be of the same capacity.

READ.ALL reads all sectors on a disk, maximizing head movement by reading alternately from low-numbered and high-numbered sectors. The user may place greatest stress on the head by specifying that a system buffer with the capacity of one record be used.

OPEN.FILES stresses the directory search hardware by opening and closing, without intervening I/O operations, up to 20 files on a disk.

READ.CYLINDERS reads one sector from every cylinder on a disk, addressing sectors in such a way as to maximize head movement while ensuring that sectors are read from all tracks (all heads are activated).

WRITE.INDEX stresses the directory search hardware by writing an index file, of a length specified by the user, to disk and then reading the file back again, checking the data records for accuracy.

NOTE

This test requires a high proportion of execution time (approximately two minutes for a single file of 25 records; approximately 30 minutes for a single file of 500 records).

READ.RANDOM stresses the head movement but under conditions more closely approximating those of a real-life environment than those of test 1, in that records to be read are randomly selected. Also, unlike test 1, the duration of the test does not depend on disk size as the user specifies the number of records to be read.

Tests are always performed in the order in which they are listed above.

CPU.IO

For each iteration of the CPU.IO subsection, the programs SCR.MPL and SCR.COBOL are each executed once, each via a ZIP (DISPLAY). If a ZIP fails, the fetch value is examined. If the failure was because of a full mix, the ZIP is repeated until either it is successful or another fetch value is returned. For all other fetch values after a ZIP failure, the appropriate message from those listed above for DCR is displayed, the program name "DCR" being replaced by "SCR.MPL" or "SCR.COBOL". Messages 12 and 13 do not apply to SCR.MPL or SCR.COBOL, as the ZIP is without PAUSE.

As SCR.MPL and SCR.COBOL are executed via ZIPs without PAUSE, multiple copies of each (produced by successive iterations of CPU.IO) may run concurrently. However, when control has been transferred to a copy of either SCR.MPL or SCR.COBOL, its status is not checked by SCR.

The ZIP to execute SCR.MPL includes a 2-character initiating message which is the ASCII value of the current CPU.IO iteration number.

SCR.MPL

The 2-character initiating message is used as the last two characters in the 8-character FID, "SCR.M.cc", of a random access disk file on the System Disk. This file is opened conditionally. If the open is unsuccessful, the following messages are displayed:

"FETCH VALUE = <hex no>"
"SYSTEM DISK NOT AVAILABLE"
"SCR.MPL COPY <2-digit no.> TERMINATED"

(where the <2-digit no.> is the 2-character initiating message)

and goes to end-of-job.

If the file is opened successfully, the following comparative expression is computed iteratively:

$$(((I+1)*5)*((I+1)*5))/((I+1)*5) = (I+1)*5$$

where I is incremented by 1 from 0 to 50 and is equal to the ordinal number of the computation iteration.

At each computation, if the expression is evaluated as false, the following message is displayed:

"P1 COMPUTATION ERROR, I = <decimal no>"

If the expression is evaluated as True, a fixed value for I is computed and output as the first two bytes of a 180-byte record. If the WRITE operation results in a conditional fail and the last two bytes of the fetch value are hexadecimal "0000", the WRITE operation is repeated. If the last two bytes of the fetch value on a conditional fail are not hexadecimal "0000", the following messages are displayed:

"SCR.MPL COPY <decimal no>"
(where <decimal no> is the 2-character SCR.MPL initiating message) and
"FETCH VALUE = <hex no.>"

If the WRITE operation causes an error, the MCP error message is the only message displayed.

Immediately after a record is written, it is read back into memory and the first two bytes compared with the current value of I.

If the READ operation causes a conditional fail and the last two bytes of the fetch value are hexadecimal "0000", the READ operation is repeated. If the last two bytes of the fetch value are not hexadecimal "0000", the following messages are displayed, as for the WRITE operation:

"SCR.MPL COPY <decimal no>"
"FETCH VALUE = <hex no>"

If the READ operation causes an error, the MCP error message is the only message displayed. If the comparison between the input value and I fails, the following message is displayed:

"INPUT/OUTPUT MISMATCH, I = <decimal no>"

After 51 compute-write-read-compare iterations have been performed, SCR.MPL displays the following messages:

"SCR.MPL COPY <decimal no> TERMINATED"
"WRITE CONDITIONAL FAILS = <decimal no>,
WRITE ERRORS = <decimal no>."
"READ CONDITIONAL FAILS = <decimal no>,
READ ERRORS = <decimal no>

SCR.MPL then closes the disk file with RELEASE, and goes to end-of-job.

Using the initial values:

A = 10
B = 1
C = .1
D = .01
E = .2

SCR.COBOL calculates:

X = (A+B) * (C−D)/(E*E)

using the COMPUTE statement, and calculates the same expression in an alternative manner, using the individual arithmetic statements ADD, SUBTRACT, MULTIPLY and DIVIDE, assigning the results to variable X1. The value X2 = −1 * X is also calculated.

The values X, X1, and X2 are moved to fields of a line image, using the editing symbols : % Z B − * CR DB, and the line is printed by the first available printer on a page headed "PROGRAM CMTHS". Sub-headings are printed for each of the fields.

The values X, X1, and X2 are calculated and printed five times, the values of A, B, C, D and E being multiplied by 2 after each calculation.

Each execution of SCR.COBOL should produce one printed page, as shown here:

COMP. VALUE	CALC. VALUE	EDIT 1	EDIT 2	EDIT 3	EDIT 4	EDIT 5	EDIT 6	EDIT 7
002475	002475	$ 24.75	$ 24.75	.75	**24.75	$002475	**24.75CR	$002475DB
009900	009900	$ 99.00	$ 99.00	.00	**99.00	$009900	**99.00CR	$009900DB
039600	039600	$ 396.00	$396.00	.00	*396.00	$039600	*396.00CR	$039600DB
158400	158400	$1584.00	$584.00	.00	1584.00	$158400	1584.00CR	$158400DB
633600	633600	$6336.00	$366.00	.00	6336.00	$633600	6336.00CR	$633600DB

After the last line of the page has been printed, SCR.COBOL goes to end-of-job.

MT.IO

For each iteration of the MT.IO subsection, a file is opened for output on each of the magnetic devices requested by the user.

To each device on which a file was successfully opened, a file of the size specified by the user (or the default size of 256 records) is written. Each record is 120 bytes long and contains 60 fixed point values, where each fixed point value is the current record number. The current record is written to each of the open files before the record is updated. If the WRITE operation causes a conditional fail and the last two bytes of the fetch value are hexadecimal "0000", the WRITE is repeated. If the WRITE operation fails, causes a conditional fail and the last two bytes of the fetch value are not hexadecimal "0000", or if the WRITE operation causes an error, the following messages are displayed:

"SCR MT.IO; ITERATION <decimal no>:"
"WRITE FAIL ON <device> FILE <file no> RECORD <record no>"
"FETCH VALUE = <hex no>"

and the file is half-closed.

After output of all files has been completed, all open files are half-closed and re-opened for input. If an open fails, the following messages are displayed:

"SCR MT.IO; ITERATION <decimal no>:"
"FILE <file no> DID NOT OPEN FOR INPUT ON <device>"
"FETCH VALUE = <hex no>."

All open files are read, the current record being read from all open files before the record number is updated. If the READ operation causes a conditional fail and the last two bytes of the fetch value are hexadecimal "0000", the READ operation is repeated. If the READ operation causes a conditional fail and the last two bytes of the fetch value are not hexadecimal "0000", or if the READ operation causes an error, the following messages are displayed:

"SCR MT.IO; ITERATION <decimal no>:" "READ FAIL ON <device> FILE <file no> RECORD <record no>" "FETCH VALUE = <hex no>"

and the file is closed and purged.

As each record of each file is input, the 60 fixed point values it contains are compared with the current record number. In the event of a mismatch, the following messages are displayed:

"SCR MT.IO; ITERATION <decimal no>:"
"WRITE/READ MISMATCH ON <device> FILE <file no>, RECORD <record no>"

After all files have been input, all open files are closed and purged.

The output-input operations described above are repeated for the number of MT.IO iterations specified by the user.

OTHER.IO

The OTHER.IO subsection of SCR is currently limited to the first printer available at execution, using MPL Line Controls.

Each iteration of the subsection outputs six pages of print, each print page containing 20 pairs of print lines. Pairs of lines are separated by a space line. Each of the first pair of lines contains 120 X's, with spaces replacing X's in the leftmost character positions of line pairs, the number of spaces being incremented by 1 for each pair of lines, as illustrated in figure 4-6.

At the end of each iteration, the PRINTER file is closed with RELEASE and re-opened at the start of the next iteration. The number of iterations that are performed is determined by user input.

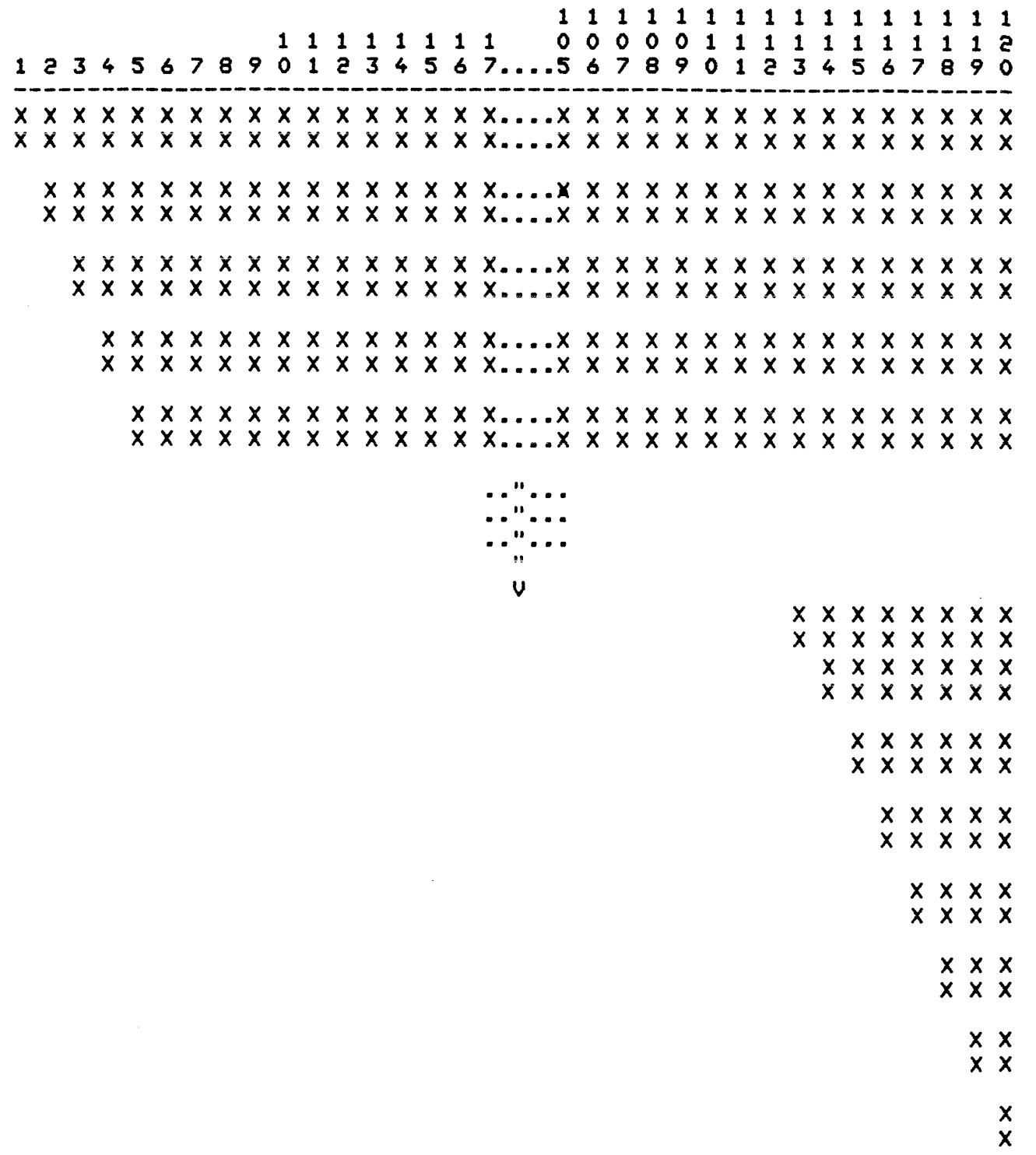
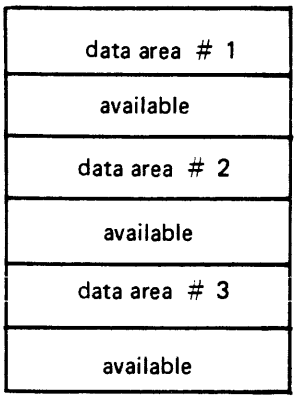


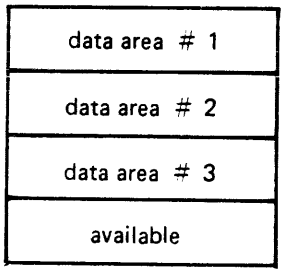
Figure 4-6. OTHER.IO Sample Printer Output

SQ (Squash Disk)

When a disk unit is used extensively with a high degree of file activity involving creation and removal of files, then it is possible for the available space on the disk to become so fragmented that it is increasingly difficult to find enough space in one single area to satisfy requests for disk space. This results in a degradation of system throughput with an increasing incidence of "NO DISK SPACE" failures and extra time is needed to search through available areas. This situation is known as "checkerboarding" of the disk. In the extreme case each area of disk in use is separated by an available area, as shown in the diagram below.



The SQ utility is designed to eliminate checkerboarding of disk, either for the whole disk or part of the disk. This process is called "squashing" the disk and is accomplished by moving each data area in turn to the first available area at a lower address. If an entire disk is squashed, then all available areas are merged into one area at the high-address end of the disk, as in the next diagram:



The options available within the SQ utility are:

Squash of complete disk.

> All data areas are moved to successively lower addresses until only one available area is left (as in diagram above).

Partial Squash

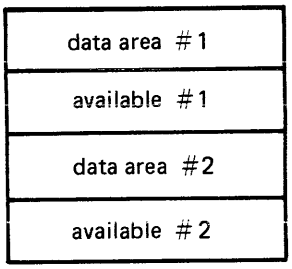
> Only data areas within a default section of the disk are moved to lower addresses within the section.

Fast Squash

> The aim of a fast squash is to create an available area of disk of a requested size. Only those data areas are moved which will allow an available area of sufficient size to be created.

Economic Squash

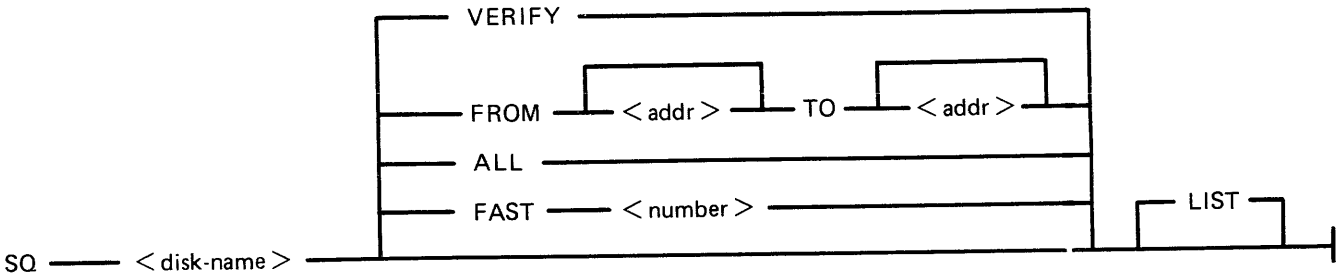
In this case, data areas are only moved if the gain in terms of available space justifies the time spent in movement of the data area. As an example, consider the following case:



where data area #1 is 100 units, data area #2 is 200 units and both available areas are 1 unit each. If available areas are merged, the available area gained would be 2 units. However, to acquire these 2 units, the 200 units of data area #2 would have to be moved. Therefore, an "economic squash" would not move data area #2. In general terms, an economic squash will ignore small available areas that are interspersed in large data areas. However, in some cases an economic squash will have the same effects as a full squash.

With all options of SQ, a further option is available to print a map of the entire disk in disk-address order both before and after squashing action.

Input is as follows:



Note: the number is in the range 1 to 65535; the start-address and end-address are 6-digit hexadecimal disk addresses, for example, 000AB3, 01A375.

When the SQ LIST option is requested, SQ opens conditionally a printer file so that the output will go to a printer irrespective of the system backup option. This prevents a printer backup file being opened on the disk being squashed.

Examples:

To perform an economic squash of disk PR2:

```
SQ PR2
```

To check the integrity of disk PR2:

```
SQ PR2 VERIFY
```

To perform a full squash of disk PR2 and list the disk map:

SQ PR2 ALL LIST

To perform a partial squash to obtain 1000 sectors:

SQ PR2 FAST 1000

To perform a partial squash on sectors 0 through 512 of disk PR2:

SQ PR2 FROM BEGIN TO 000200

To perform a partial squash on sectors 512 through 4096 of disk PR2:

SQ PR2 FROM 000200 TO 001000

To perform a partial squash on sectors 4096 to the last addressable sector of disk PR2:

SQ PR2 FROM 001000 TO END

Before performing any function which involves physically moving data areas, the integrity of the disk is checked. Integrity involves analyzing disk assignment to verify that the entire area of the disk is described in the file directories and available table, checking the directories themselves and attempting to resolve anomalies (for example, missing areas or overlapping areas). Only after the integrity is verified are areas of disk physically moved.

Certain areas of disk will not be moved in any circumstance. These are areas of disk currently marked as in use and any system log files.

Output Messages

Message	Possible Causes	Suggested Action
VERIFICATION OF <diskname> COMPLETED	SQ successful.	None.
-SQUASH COMMENCING	Squash phase begins.	None.
SQUASH OF DISK <diskname> COMPLETED	SQ successful	None.
-SIZE OF LARGEST AVAILABLE AREA IS <number> SECTOR(S)	Self-explanatory.	None.
-TOTAL AVAILABLE SPACE IS <number> SECTORS IN <number> AREA(S)	Self-explanatory.	None.
SQ HAS INFORMATION TO PRINT, ENTER Y IF IT HAS TO BE IGNORED OR N TO PRINT IT	The LIST option has been specified in the initiating message but there is no printer available.	Enter AX <mix-no> Y to continue without list or AX <mix-no> N after printer is made available.
PREVIOUS EXECUTION OF SQ ON DISK <diskname> WAS ABNORMALLY TERMINATED WHILE MOVING FILE <filename>	System crashed or SQ was DS'ed during previous execution of SQ.	Check integrity of specified file.

(continued)

Message	Possible Causes	Suggested Action
-THIS FILE MAY BE CORRUPT	Self-explanatory.	Check integrity of specified file and replace if suspect.
DISK <diskname> HAS MISSING OR OVERLAPPING AREAS	There are areas on disk not referenced by the available table or the disk directory or areas referenced by both.	None. Utility has corrected this situation.
-A LISTING OF DISK SPACE USE WILL BE PRINTED	Self-explanatory.	None.
HARDWARE ERROR DETECTED ON DISK <diskname> WHILE MOVING FILE AREA TO SECTOR RANGE <address> - <address>	The utility has encountered a parity error on the specified area on disk.	The utility left the area available. After SQ has finished, use CHECK. DISK to find the bad sector. Use XD to delete the sector.
HARDWARE ERROR DETECTED ON DISK <diskname> WHILE MOVING FILE AREA FROM SECTOR RANGE <address> - <address>	The utility has encountered a parity error on the specified area on disk. File is not moved.	Check disk integrity after SQ has finished.
HARDWARE ERROR DETECTED ON AN AREA OF FILE <filename> ON DISK <diskname>	The utility has encountered a parity error on the specified file.	Check disk integrity after SQ has finished.
-THIS AREA WILL NOT BE MOVED	Self-explanatory.	None.
DISK <diskname> NOT SQUASHED	The utility cannot run.	Look for the reason in the second message.
-NO SUITABLE AREAS IN SPECIFIED SECTOR RANGE	The utility cannot obtain the requested space in the specified area.	Expand the range for SQ.
-AVAILABLE AREA OF REQUESTED SIZE ALREADY EXISTS	Self-explanatory.	None.
-AVAILABLE AREA OF REQUESTED SIZE CANNOT BE CREATED	There is no way to create the requested area.	None.
-SPECIFIED SECTOR RANGE IS BEYOND END OF DISK <diskname>	Self-explanatory.	Correct input and re-enter.
-<diskname> IS NOT A PHYSICAL DISK	An attempt was made to run SQ on a pseudo disk.	Correct input and re-enter.

(continued)

Message	Possible Causes	Suggested Action
-DIRECTORY OF DISK <diskname> IS NO LONGER VALID	The problems the specified disk are so great that SQ cannot clear them.	Try to copy the files from the specified disk to another disk or to tape. Then re-initialize the disk.
-SPACE ON DISK <diskname> IS TOO FRAGMENTED TO SQUASH	There is insufficient space on disk for SQ to run.	Use COPY and RM or UNLOAD to save files and create space for SQ to run.
-HARDWARE ERROR DETECTED DIRECTORY OF DISK <diskname>	The utility has encountered a parity error on the directory of the specified disk.	Specified disk cannot be used any more and must be initialized.
-CANNOT SEPARATE OVERLAPPING BAD AREA AND AREA OF FILE <filename> ON DISK <diskname>	Self-explanatory.	Try to dump or copy files from disk. Disk must be re-initialized before re-use.
-TOO MANY FILE AREAS ON DISK <diskname> PLEASE TRY SQUASHING AN AREA OF THE DISK	Self-explanatory.	Use SQ FROM/IO option to SQ an area of the disk.
-CANNOT SEPARATE OVERLAPPING AREAS OF FILES <filename> AND <filename> ON DISK <diskname>	Self-explanatory.	Try to dump or copy files from disk. Disk must be re-initialized before re-use.
- AREAS MISSING FROM <diskname>, PLEASE RUN SQ VERIFY	Self-explanatory.	Run SQ <diskname> VERIFY.
-TOO MANY OPEN FILES OR BAD AREAS ON DISK <diskname>	Self-explanatory.	Close open files by bringing any running program to end-of-job. In case of bad sectors re-initialize the disk after having saved the required files.
-PLEASE BACKUP FILES AND RE-INITIALISE DISK	Self-explanatory.	Backup files using COPY or DUMP.
-SPECIFIED DISKNAME IS INVALID	Self-explanatory.	Correct input and re-enter.
-PARAMETER FOLLOWING KEYWORD FAST IS GREATER THAN 65535	Self-explanatory.	Correct input and re-enter.
-PARAMETER FOLLOWING KEYWORD "TO" OR "FROM" IS INVALID	Self-explanatory.	Correct input and re-enter.

General Guidelines

If the information contained on a disk is important always ensure that backup exists before attempting to squash it.

Always run "SQ VERIFY" before running an actual squash. This will give an indication of the state of the disk.

Do not allow disks to become too fragmented before squashing them. A full squash can be a lengthy process and can be avoided by running "SQ VERIFY" on a regular basis and running partial squash when the disk starts checker-boarding.

"SQ VERIFY" is a means of checking the integrity of any disk and, if run on a regular basis, may help pinpoint sooner any degradation in hardware performance or system software bugs. For disks that are in constant use "SQ VERIFY" should be run immediately after the first clear start of the day. This can help prevent catastrophic losses of information.

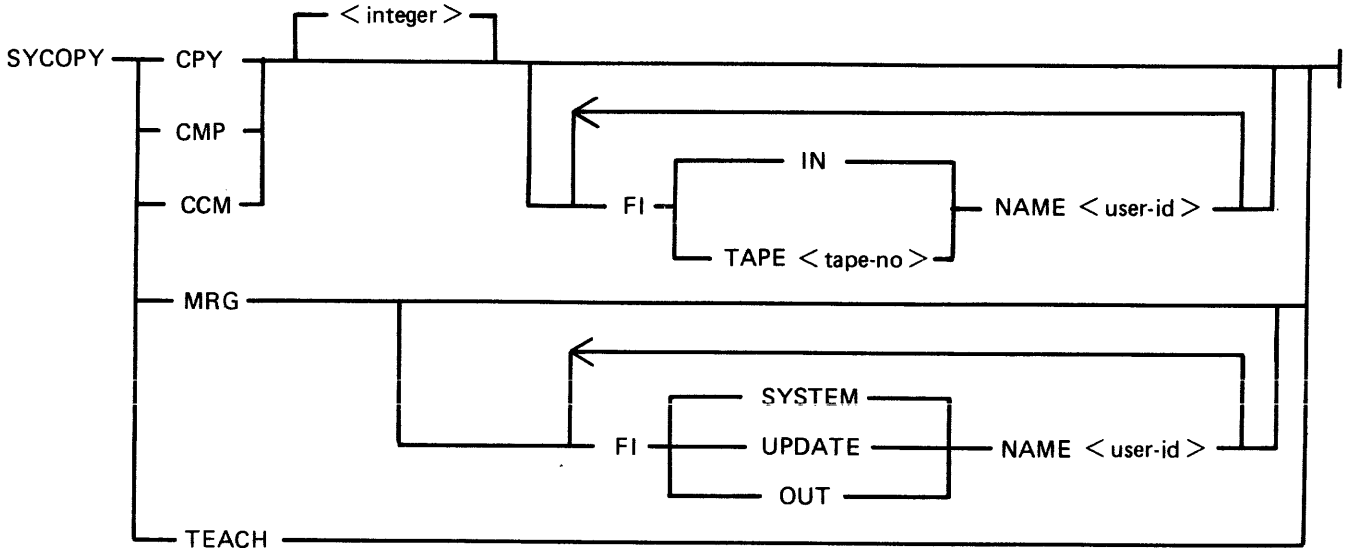
NOTE

It is strongly recommended that SQ, once loaded, is not DSed.

SYCOPY (Copy Library Tapes)

The utility SYCOPY provides the user with a means of duplicating, comparing or merging library tapes (or cassettes) with multiple copy capabilities (except for the merging option).

The initiating message has the following syntax:



where <integer> = 1-7 (defaults to 1)

The default-names for input tape is "IN" and for output tapes "TAPE.1" through "TAPE.7", except in MRG function where the input tape names are "SYSTEM" and "UPDATE" and the output tape name is "OUT". All output tapes are locked after job termination.

CPY

CPY makes <integer> copies of a library tape (created with the LD utility) with no compare. At end, the output tapes are locked, and the input tape is released.

Examples:

To copy one input tape labelled "IN" to two output tapes, labelled "TAPE.1" and "TAPE.2":
SYCOPY CPY 2

To copy one input tape labelled "FRED" to one output tape, labelled "FRED.OUT":
SYCOPY CPY FI IN NAME FRED FI TAPE.1 NAME FRED.OUT

CMP

CMP compares <integer> tapes with one input tape. On completion, all tapes are released.

For the FILE000, the comparison works as follows:

– first records: only bytes 1-20 (byte 1 being the first) and bytes 28-38 are compared (check if library tape and number of files).
– other records: only bytes 1-12 (filename) are compared.

The record sizes of input and output tapes must be the same. This restriction does not apply to buffer sizes, which may be different for input and output tapes.

Differences between the filesizes of the input tape and the output tape with which it is being compared will be detected; in this case, a comparison error for End-of-File or Not End-of-File will be displayed (see Output Messages).

Examples:

To compare one tape named "IN" with three tapes named "TAPE.1", "TAPE.2" and "TAPE.3":
```
SYCOPY CMP 3
```

To compare one tape named "FRED" with one tape named "TAPE.1":
```
SYCOPY CMP1FI IN NAME FRED
```

CCM

CCM performs CPY then CMP successively. On completion, the output tapes are locked and the input tape is released.

Examples:

To copy one input tape named "IN" to three output tapes named "TAPE.1", "TAPE.2" and "TAPE.3" and then to compare "IN" with "TAPE.1", "TAPE.2" and "TAPE.3":
```
SYCOPY CCM 3
```

MRG

MRG merges two input tapes (called SYSTEM and UPDATE by default) to one output tape (OUT by default), then compares the merged tape with the original tapes.

Duplicate files will be removed, with the UPDATE file taking precedence over the SYSTEM file.

The files will appear on the output tape in the following order:
– Duplicate Key Files (from second input tape)
– Non Duplicate Files (from first input tape)
– Remaining Files (from second input tape)

The record sizes of the input tapes must be the same, but buffer sizes may be different. The buffer size of the first input tape will be used for the output tape.

If more than one copy of the newly-merged tape is required, use CPY.

When a comparison error occurs, the faulty tape is either purged (for CCM or MRG) or released (for CMP) and an error message displayed. Refer to Output Messages for possible error messages.

Examples:

To merge two input tapes named "SYSTEM" and "UPDATE" and output one tape named "OUT":
```
SYCOPY MRG
```

To merge two input tapes labelled "A" and "UPDATE" to one output tape named "B":
SYCOPY MRG FI SYSTEM NAME A FI OUT NAME B

TEACH

TEACH displays the syntax of the initiating message.

Example:
SYCOPY TEACH

Output Messages

Message	Possible Cause	Suggested Action
<number> TAPE(S) COPIED	Successful copy function.	None.
<number> TAPE(S) COMPARED – NO ERROR	Successful compare function.	None.
<number> TAPE(S) COPIED AND COMPARED – NO ERRORS	Successful copy and compare function.	None.
2 TAPE(S) MERGED – NO ERROR	Successful merge function.	None.
<number> TAPE(S) COMPARED	One or more, but not all, of the output tapes has been	None.
<number> TAPE(S) DELETED	deleted.	
<number> TAPE(S) COPIED AND COMPARED	One or more but not all of the output tapes has been	None.
<number> TAPE(S) DELETED	deleted.	
– <tapename> IS NOT A RECOGNISED LOAD/ DUMP TAPE	The specified tape is not created by a function of the LD utility.	None.
UTILITY LIMIT REACHED WHEN RUNNING SYCOPY	The number of files on the input tape for MGR exceeds 65535.	None. Utility information.
–RECORD SIZE MISMATCH	Compare had failed. Possible hardware problem.	Re-run SYCOPY or other medium.
INVALID CASE – SYCOPY ABORTED	Error in SYCOPY.	Copy backup and re-run.
DISCREPANCY IN DUMP LEVELS	Compare had failed.	None.
END OF FILE REACHED	Self-explanatory.	None.
NOT END OF FILE	Self-explanatory.	None.

TAPELR (List Library Tape Directory)

This utility allows the operator to print detailed information about the library tape files. Output will appear either on the line printer or the console printer.

Tapes about which information is required are identified by "library-tape-name". More than one tape name may be requested during a single run of TAPELR.

Format

TAPELR library-tape-name

Examples

To print detailed information about the files on a tape called PRTAPE:
    TAPELR PRTAPE

To print detailed information about the files on tapes called PRTAPE and ICTAPE:
    TAPELR PRTAPE ICTAPE

Output Format

Ten columns of information will appear for each library tape indicated. The column headings, the format of the "values" these columns contain and the significance of these "values" is as follows:

Heading	Value	Significance
FILE.NUMBER	6 digits	Logical file number on tape.
FILE NAME	12 characters	File name.
ACTUAL SIZE	7 digits	Number of records in this file.
MAXIMUM SIZE	7 digits	Maximum number of records this file may contain.
RECORD SIZE	5 digits	Number of characters in each record.
RECS/BLOCK	5 digits	Number of records in each block.
CREATED	5 digits	Date file was created (Julian YYDDD)
ACCESSED	5 digits	Date file was last accessed by a program (Julian YYDDD).
GENERATION NO.	3 digit	
FILE TYPE	8 characters	See note below.

Note: FILE TYPE will be one of the following:
    DATA – normal data file
    CODE – object program file
    KEY – key file
    SYSTEM – system file (for example, MCP, interpreters)
    SRCELANG – source language file
    SRCELIBR – source library file

Output Messages

Message	Possible Cause	Suggested Action
– HARDWARE ERROR DETECTED ON PRINTER BACKUP DISK <disk-name>	The utility has encountered an error on the specified disk.	Check integrity of the specified disk.
– PRINTER BACKUP FILESIZE NEEDS TO BE INCREASED	The printer backup filesize is too small.	Use SO PRBK ON <number> to increase filesize.
<tapename> NOT A RECOGNIZED DUMP TAPE	The specified tape is not created by a function of the LD utility.	None.

Note: Refer to "Common Utility Output Messages" for additional messages.

TAPEPD (Print Name of a Library Tape)

This utility allows the operator to print the names of files found on a library-tape. More than one tape name may be requested during a single run of TAPEPD.

Format:

TAPEPD library-tape-name

Examples:

To print the names of files found on a tape called PRTAPE:

TAPEPD PRTAPE

To print the names of the files found on tapes called PRTAPE, ICTAPE and GLTAPE:

TAPEPD PRTAPE ICTAPE GLTAPE

Output format:

For each tape requested, the following information in displayed:

MT library-tape-name DUMPED ON day of week DD month YY year contains:

This message precedes the names of files found on each tape. The list itself contains 3 files per line.

Output Messages

Message	Possible Causes	Suggested Action
<library-tape-name> NOT A RECOGNIZED DUMP TAPE	This tape was not created by either the DUMP or UNLOAD function of LD utility. It is ignored by the TAPEPD utility.	None.
END TAPEPD	End of Job message.	None.

Note: Refer to "Common Utility Output Messages" for additional messages.

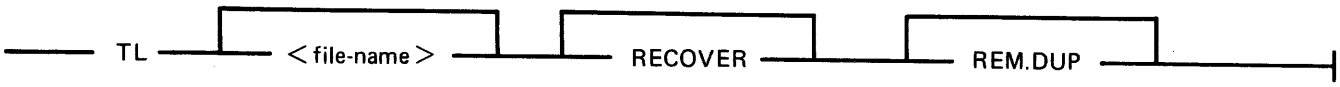
TL (Transfer Logs)

At the beginning of any session, a number of 'logging-files' are created. The purpose of these files is to maintain a record of all the input/output transactions which occur during this period of time.

For a fuller explanation of logging and its functions, see "Logging in the 3.04 Release" earlier in this section.

In order to produce an easily accessed file containing the data in all the primary files, they are consolidated into a single file. This is done automatically at warmstart, but the user can invoke consolidation of log files at any time by running the utility TL.

Format:



The utility determines the number of files to be consolidated. It then transfers each "READY-TO-TRANSFER" file, closing the consolidated file after each log-file has been transferred, until it reaches the file which was in an "ACTIVE" state at execution of TL.

If the option "RECOVER" has been specified, the utility enters the "ACTIVE" log file when it reaches it, and transfers all the entries up to the latest one, providing the file is not in use. If it is in use, the utility displays:

"ILLEGAL USE OF RECOVER PARAMETER", "ACTIVE FILE NOT CONSOLIDATED"

and only consolidates "READY-TO-TRANSFER" files. Otherwise, it stops consolidating when it reaches the "ACTIVE" file.

All log-files transferred are left in a "TRANSFERRED" state.

TL transfers maintenance entries to the secondary log file SYS-MLGjjjnn (jjj is the day number of the julian data, nn is a cyclic numeric progression in the range 00 to 99).

TL will also transfer system messages to the secondary log file SYS-HLDjjjnn, only if logging method is non-cyclic. If the logging method is cyclic and TL has been zipped by SYS-SUPERUTL (this only occurs at warmstart), the system messages are transferred to a secondary log file called SYS-LOG-HOLD.

Each time the MCP fills a log file, TL is zipped and transfers the contents of all the READY-TO-TRANSFER files, guided by information held in SYSCONFIG. If system messages are NOT to be logged, or they are to be logged in a cyclic manner, then TL only transfers maintenance entries.

When TL is zipped by SYS-SUPERUTL, both options are included in the initiating message. When TL is zipped by the MCP, NO options are specified, therefore if a duplicate file condition arises, the operator must take action to resolve the situation.

When entries are transferred to the maintenance log, the various input/output counts from the input files are merged with those contained in the output file. Therefore, when the output file becomes full, input/output counts in this file are valid for all the maintenance entries contained in it.

When the date changes on the system, the next time TL is zipped, the previous secondary log files will be closed with CRUNCH.

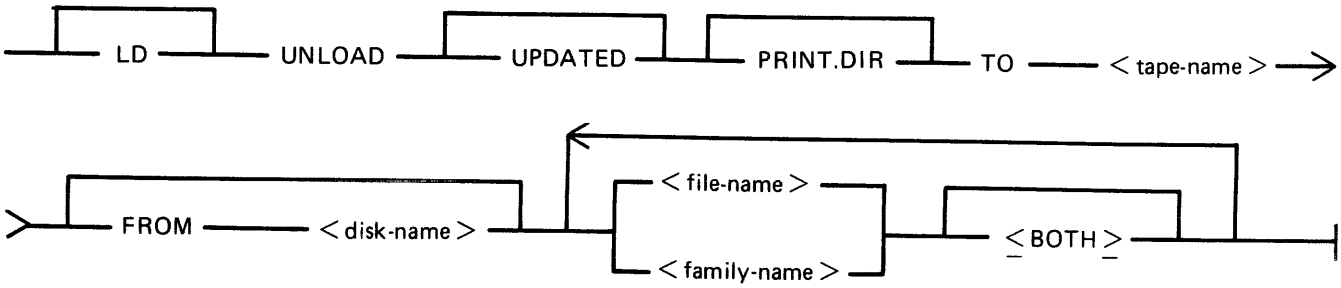
Output Messages

Message	Possible Cause	Suggested Action
UNABLE TO ACCESS SYSCONFIG	Another program is using SYSCONFIG	Wait until the other program is finished then start TL again.
-TL IS UNABLE TO RUN WITH THIS VERSION OF SYSCONFIG	Self-explanatory.	Copy a suitable version of SYSCONFIG to your system disk.
LOG INFORMATION TRANSFERRED	TL successful.	None.
NO LOG INFORMATION TRANSFERRED	TL failed. The reason is given in the second message.	See the second message for the reason.
-NO ACTIVE LOG FILE WAS FOUND	Self-explanatory.	None.
-NO READY-TO-TRANSFER LOG FILES WERE FOUND	Self-explanatory.	None.
SECONDARY LOG FILES WILL BE CREATED ON THE SYSTEM DISK	Self-explanatory.	None.
SECONDARY LOG FILE <filename> IS SMALLER THAN EXPECTED AND WILL NOT BE USED	Self-explanatory.	None. Utility continues.
SECONDARY LOG FILE <filename> CREATED	Self-explanatory.	None.
<filename> REMOVED	The REM.DUP option was used.	None.
-NO SPACE ON DISK <diskname> FOR TRANSFERRING LOG INFORMATION	Self-explanatory.	Create space by RM or UNLOAD and re-run the utility.
-FILE DIRECTORY ON DISK <diskname> IS FULL	Self-explanatory.	Create space in the directory by removing files from disk.
-FILE <filename> IS NOT A VALID LOG FILE	The specified file has not the filetype of a log file.	Check input for correct file name.

UNLOAD (LD – TAPE LIBRARY UTILITY)

This function is part of the utility LD. UNLOAD provides the capability of copying files or groups of files from the disk specified by disk-name or, if no disk-name is specified, from the system disk. After each file is copied to tape, it is removed from the disk.

Format:



NOTE

A purged tape must be on line and ready for use by this utility. Purged tapes can be produced using either the PG or SN intrinsics on an on line and ready tape.

UPDATED

If UPDATED is specified, only updated or newly created files will be dumped, that is, files with the "Modified" bit in the DFH set. (An LR listing will identify which files have the bit set.)

PRINT.DIR

If PRINT.DIR is specified, an accurate tape directory will be printed at the end of the dump. The list of files will be the same as in the directory at the head of the tape. For each file dumped, a list of its attributes will be printed. If any file was not dumped, the reason why will appear beside the name. When PRINT.DIR is specified, all messages of the form:

"<file-name> DUMPED"
"<file-name> REMOVED"

are suppressed.

Displays will be limited to exception conditions, for example, if a file cannot be dumped or loaded for some reason.

TO <tape-name>

This is the name which will be given to the tape.

FROM <disk-name>

If the option FROM disk name is specified, the files will be dumped and removed from the disk named <disk-name>. If this option is not used, then the system disk is assumed.

<file-name> <family-name>

This is the file or family of files to be dumped to tape. Groups of files may be dumped by separating the file names or family names by a comma "," or a space " ".

<BOTH>

If <BOTH> is specified immediately following a request to dump a key file then, provided that the pertinent data file is on the same disk, the data file following the key file on the tape will also be dumped.

Since the UNLOAD function is part of the utility LD, "LD" is actually what appears in a mix message. To discontinue the UNLOAD function, "DS <mix-number>/LD" must be used.

Examples:

Example 1. To copy and remove all the files in the family named TEST from the system disk to a tape named TPTEST, the following initiating message is used:

```
LD UNLOAD TO TPTEST TEST=
```

Example 2. To copy and remove all the key and data file pairs in the family MILK, which have been created or updated since the last time they were saved onto tape or disk, from the user disk DKMILK to the tape named TPMILK, the following initiating message is used:

```
LD UNLOAD UPDATED PRINT.DIR TO TPMILK FROM DKMILK
MILK= <BOTH>
```

The addition of the keyword PRINT.DIR produces an accurate directory at the end of the dump.

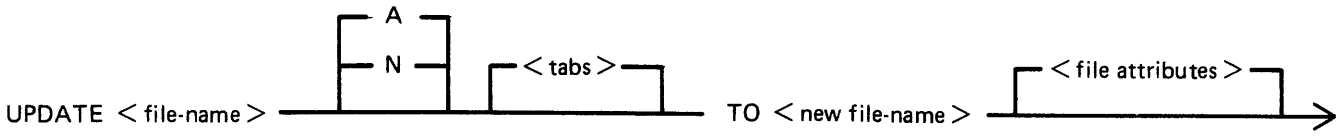
Output Messages

For output messages, refer to LD utility.

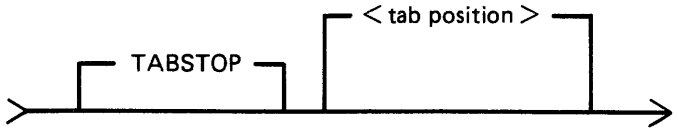
UPDATE (Disk File Update – B 90 Only)

This utility allows the operator to construct new disk files from existing files. "CREATE" and "AMEND" use many similar features.

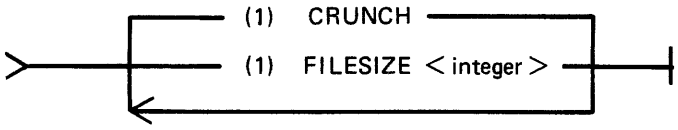
Format:



where < tabs > is



and < file attributes > is



< tab position > and < no. of records > are integers.

The existing file must be a source or data file. Attributes such as Record Size will be taken from this file and used for the "new" file.

Input may be specified as "A" (alphanumeric) or "N" (hexadecimal) (see CREATE utility for details).

By specifying TABSTOP in the initiating message, UPDATE will set up tab positions coinciding with the end of the console line as well as any other tabs specified.

The "number" option may be used to set "tab" positions for character input (see CREATE utility for details).

The maximum number of records likely to be written to the new file may be specified using the FILE-SIZE option. If no total number of records is specified, the number will be taken from the old file.

The CRUNCH option allows the operator to specify that the new file should occupy the minimum area of disk, but never be extended.

The utility operates in three modes: "Record Modify" (PK2), "Record Select" (PK3) or "Record Insert" (PK4).

PK1	PK2	PK3	PK4	PK5	PK6
write last and get next	modify	select	insert	delete	EOJ

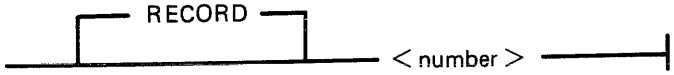
OCK3 will display the current tab position anywhere in the record (except at the start of a record where the record number is displayed).

OCK4 will output a display of the above options. If OCK4 is selected while in MODIFY, SELECT or INSERT mode, an asterisk (*) will be displayed preceding that mode.

PK1 is used to write the last record processed to the new file and then select and print the next logical record from the old file. The printout or screen will show the record number in the old file of the selected record, together with the next record number to be written to the new file.

PK5 is used to delete the last record printed by selecting and printing the next logical record from the old file without writing the last record to the new file. The printout will show the record numbers in the old file of the selected record, together with the next record number to be written to the new file.

If PK3 is used, the required record is identified by logical record numbers using this format:



The "number" cannot be less than the last record obtained from the old file, or greater than the number of records in the file. During the process of locating the required record, all records from and including the last record processed, plus one up to the one immediately prior to the selected record, will be copied from the existing file to the new file. When found, the selected record will be printed, with its record number in the old file followed by the record number that the next record written to the new file will take. "Record Modify" (PK2) or "Record Insert" (PK4) may then be selected. Note that a record inserted by Record Insert mode will be positioned after the selected record in the new file. Selecting Record "0" allows records to be inserted before Record 1 of the old file.

PK2 is used to make alterations to existing records. This PK operates as PK2 in the CREATE utility (see CREATE for details).

PK4 allows the operator to insert additional records in the new file after the last selected record of the old file. Input may be made in accordance with the specified tab stops. UPDATE will accept input only up to the next tab position. Fill characters will be displayed up to the next tab position if that tab is not reached. The utility prints the record number in the old file of the last record taken from the old file, and the record number in the new file, of the next record to be output, prior to accepting keyboard input. When all insertions have been made at a particular point in the file, an available PK may be pressed to select the next mode or terminate the utility.

NOTE

To insert a record at the beginning of a new file, Record "0" should be selected in Record Select Mode, prior to Selecting Record Insert Mode.

Default tab positions have been selected to allow a maximum number of characters to be entered on one line.

Tabs may be selected manually in addition to the default tab positions.

Default tab positions for UPDATE are as follows:

(source or alphanumeric) : 101 201 301 401
(data hexadecimal) : 56 101 151 201 251 301 351 401 451

Examples:

UPDATE FILEA TABSTOP TO FILED	101 201 301 401	(source file)
UPDATE FILEB TABSTOP 151 200 TO FILEF	101 151 200 201 301 401	(alphanumeric input)
UPDATE FILEC N TABSTOP 46 420 TO FILEG	46 51 101 151 201 251 301 351 401 420 451	(data hexadecimal)

If the input file-name (OLD.FILE) is the same as the output file-name (NEW.FILE), the old version of the file will be removed and the generation number of the new version of the file will be set to one more than that of the old version.

Examples:

To update a source file called "APFILE" of record size 40 bytes into a file called "APFILE2":

```
UPDATE APFILE 5 10 15 20 TO APFILE2
```

The utility will illuminate PK1 and PK6. By pressing PK1, next sequential record will be selected and printed.

As the utility is already in the Record Select Mode, by typing a record number, the specified record number and its contents are printed.

```
4    4    ABCDEFGHIJKLMNOPQRST
```

Note that the first "4" is the sequence number in the old "APFILE" and the second "4" is the sequence number in the "APFILE2" file.

At this point, the following PKs are available for selection:

PK1 – select next sequential record and print
PK2 – modify the selected record
PK4 – insert new record after selected record (that is, "4")
PK5 – delete the last selected record by selecting next record
PK6 – terminate the utility

To replace characters within a selected record, press PK2 and type the replacement:

```
D:ZZZZ:  OCK1
```

resulting in:

```
4    4    ABCDZZZZIJKLMNOPQRST
```

To insert characters within a selected record, type:

```
Z:XXXXXX:  OCK2
```

resulting in:

4 4 ABCDZXXXXXXZZZOPQRST

To insert a record after record 7 of the existing file, press PK3 (Record Select Mode) and type a record number:

7 OCK1

Note: At this point, the record selection number given cannot be less than the last selected record, for example, records from 1 through 3 cannot be selected. In this example, records 5 to 6 are also copied to the new file.

Press PK4 (Record Insert Mode). The utility will print the last selected record number on the left and the next record number after that, and allows the operator to key in the record to be inserted.

7 8 AAAAAA

The record inserted will be numbered 8 in the file "APFILE" and will contain "AAAAAA".

Output Messages:

Refer to the section on the "CREATE" utility for output messages.

WL (What Log File)

This utility allows the operator to determine the number of log files present and their status.

Format:

WL

The WL utility determines which log files are present (from information contained in SYSCONFIG), and their states. WL displays this information in the following way:

File Name	File State
SYS-LOG-01	<status>
SYS-LOG-02	<status>
SYS-LOG-03	<status>
SYS-LOG-04	<status>
BACKUP FILE(S) ON	<disk name>
SYS-MLGjjjmm ACT.	<integer> RECS.
MAX.	<integer> RECS.
SYS-HLDjjjhh ACT.	<integer> RECS.
MAX.	<integer> RECS.

The file SYS-HLDjjjhh will only be present if Non-cyclic SPO logging is active.

<integer> is the number of records, or may be replaced by one of the status values if the relevant file falls into one of the categories.

<status> may have one of the following values:

<status>	Description
INVALID	This indicates that the relevant log file is not in a recognized state, possibly due to some sort of corruption.
ACTIVE	This log file is being used by the MCP.
READY TO TRANSFER	The MCP has filled this primary log file and TL is required to transfer the Maintenance entries to SYS-MLGjjjmm, and System message entries, if Non-Cyclic SPO logging is active, to SYS-HLDjjjhh.
TRANSFERRED	TL has transferred the Maintenance entries to SYS-MLGjjjmm, and System message entries, if Non-Cyclic SPO logging is active, to SYS-HLDjjjhh.
NEXT ACTIVE	The MCP has cleared this primary log file ready for use.
NOT USED	This log has not been used yet.
NOT FOUND	This log file should be present, due to the information contained in SYSCONFIG, but cannot be found on the disk.
IN USE	This log file is in use, by another program.

Output Messages

Message	Possible Cause	Suggested Action
– UNABLE TO ACCESS SYSCONFIG	Another program is using SYSCONFIG	Wait until the other program is finished then start WL again.
– WL IS UNABLE TO RUN WITH THIS VERSION OF SYSCONFIG	Self-explanatory.	Copy a suitable version of SYSCONFIG to your systm disk.
HARDWARE ERROR DETECTED WHILE READING SYSCONFIG	The utility has encountered a parity error on the SYSCONFIG file.	Copy a suitable version of the SYSCONFIG file to the system disk and re-start the system.

NOTE: Refer to "Common Utility Output Messages" for additional aid.

XD (Delete Bad Disk Sectors)

This utility allows the disk directory to be marked such that selected portions of the disk will not be used. The utility will normally be used after recurrent errors with the message:

DK...ERROR

where the dots indicate further information. Refer to section 7, MCP Output Messages, for the following numbered messages:

2 PARITY ERROR
3 TIMEOUT ERROR
4 ADDRESS ERROR
45 PARITY ERROR (fatal to program)
46 TIMEOUT ERROR (fatal to program)
47 ADDRESS ERROR (fatal to program)

The further information will indicate the disk address at which the failure occurred.

The utility is initiated as followed:

Format:

XD disk-name address length

The disk-name is the disk-id of the disk from which sectors are to be deleted. The area to be deleted is given in hexadecimal by the starting address and length.

Example:

To delete 64 sectors starting from hex 395F from disk PR2B:

XD PR2B 395F 40

NOTE

The specified sectors must not be in use as part of a file. The area must be made available by first removing any file if necessary.

Warnings

Once sectors are deleted via XD from a disk, they can be restored to use only by a disk initialization. Do not therefore XD a larger area than required.

As XD alters the disk directory, do not run any other programs with it.

Do not execute XD from the same disk as the one from which sectors are to be deleted: for example, it is recommended that XD is always executed from the system disk and always deletes sectors from a user disk.

Output Messages

Refer to "Common Utility Output Messages" for additional messages.

Message	Possible Causes	Suggested Action
@length@ SECTORS FROM @address@ DELETED	Successful termination of XD.	None.
DISK disk-name FOR XD NOT AVAILABLE	Specified disk is not available.	Check input: make disk ready.
PLEASE RUN SQ ON DISK disk-name	Available table is full and available entries are lost. SYSMEM will be opened on the specified disk with otheruse LOCKED.	Run SQ.
AVAILABLE TABLE FULL – ENTRY @address@ @length@ LOST	No entries left in available space table for XD to complete properly.	The disk may still be used, but a KA will indicate some sectors which cannot be accessed; these may only be retrieved by initializing the disk.
RANGE FROM SECTOR @address@ NOT DELETED, NOT AVAILABLE	Only the part of the area specified for deletion is available. The utility will remove the available area only.	None.
SPECIFIED RANGE IS BEYOND END OF DISK	The sector address specified is greater than the address of the last physical sector on the disk.	None.
DISK <disk-name> IS NOT A PHYSICAL DISK	The specified disk is a pseudo-disk. The utility will go to end of job.	None.

Note: Refer to "Common Utility Output Messages" for additional messages.

SECTION 5
THE SORT/MERGE

INTRODUCTION

This section describes the capabilities of the SORT facility. There are two modules: the sort itself, known as the "sort intrinsic" (file-name SORTINTRINS), and an interface to this intrinsic which allows the user to specify particular sorts and merges. The latter module is sometimes called "the sort", but is more properly called the "sort language processor" (file name SORT). The sort intrinsic is implementation-dependent, as it uses specific hardware features where possible (although output messages are standardized), while the sort language processor is a CMS common item.

This section first describes the user interface to the sort, and then covers the various facilities in some detail.

The interface to the sort from COBOL programs is described in the COBOL language reference manual.

GENERAL FEATURES

The following capabilities are provided:

The records within a file may be sorted on a series of specified keys, each key ascending or descending, using a regular sort or an in-place sort.

A tagfile (suitable for use as an ADDROUT file in RPG or for limited access in COBOL) may be created from a file using a series of keys, each ascending or descending.

A key file (suitable for full indexed access) may be created using a specified unsigned key (ascending only), with an optional check for duplicate keys.

A number of files may be merged using a series of keys, each ascending or descending.

The regular and index sort intrinsics are able to sort files up to the maximum allowable size in CMS, which is 1,048,560 records.

The inplace intrinsic has a B 90 implementation limit of 400,000 records.

The merge intrinsic will produce an output file of up to 1,048,560 records.

The memory required for a sort is calculated dynamically, with a default size of 15 K bytes.

The workfile buffer default size for index sorts and regular sorts is 720 bytes.

INVOKING THE SORT

The sort can be executed in three ways:

1. Entering the name SORT with the details for the sort in the initiating message.
   for example, SORT FILE IN INFILE (DISK) OUT OUTFILE (DISK) KEY (1 12)

2. Entering the name SORT with the details for the sort described in a starfile.
   for example, SORT *SORTTHISFILE

3. Entering the name SORT preceded by the datacomm control character from a datacomm terminal with TD 830 capabilities, in which case an interactive version of SORT will be started, for example, $SORT.

If the sort specification is given in the initiating message, the message may not be longer than 255 characters. If the sort specification is zipped from a user program, the message may not be longer than 716 characters. If it is not possible to specify a complex sort within these limitations, a starfile should be used.

If a required starfile is not present, the non-interactive SORT displays

FILE <filename> UNAVAILABLE ON (SYSTEM) DISK
FIX AND REPLY "OK" ELSE <NULL>

and waits on ACCEPT.

There are two alternative responses:

1. Make the <filename> available and enter AX <mix-no> OK
   or
2. Cause SORT to go to end-of-job by entering AX <mix-no>

If specification statements are given in the initiating message, control characters such as carriage return and line feed are treated as space characters.

If the name of a starfile is entered without a preceding diskname and the starfile name is not longer than seven characters, the file will be searched for first on cards, then on cassette and then on the system disk. If the name of the starfile exceeds seven characters, the file will be searched for on the system disk.

A starfile on a cassette must be created using the COPY utility. Files created by the LD utility are not suitable.

A starfile on disk must be of filetype source or data. Use FILEUTL or CMSCANDE to create them.

A starfile cannot be accessed by SORT if the file is in use by another program.

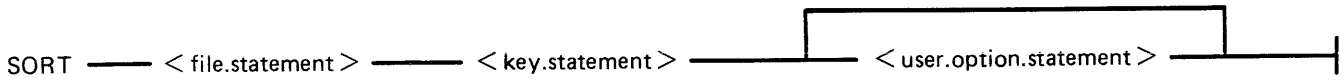
If SORT is to be started on a datacomm terminal, the terminal must be a TD 830 capable device. SORT can run with either a GEMCOS interface or a shifted TMCS interface. See the INTERACTIVE SORT later in this section for details.

THE SORT LANGUAGE

The specification for a sort consists of three statements:

1. The FILE Statement
2. The KEY Statement
3. The USER-OPTION Statement

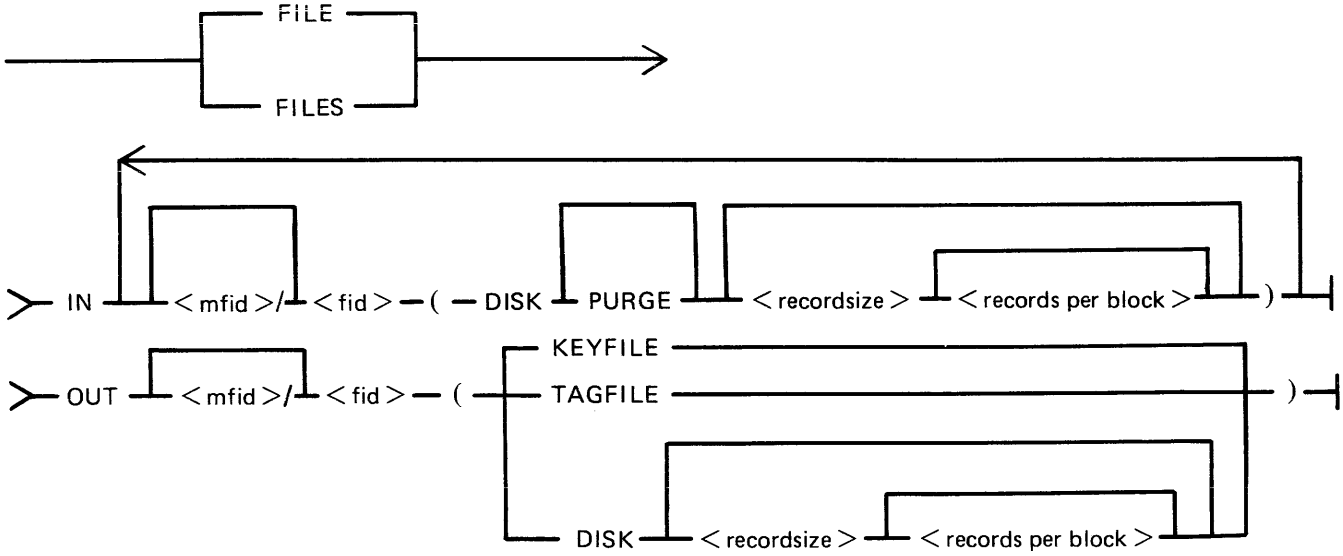
The main railroad diagram is:



There must be one file statement, one key statement, and, optionally, one or more user-option statements, in any particular sort invocation. All keywords are reserved: that is, they can only be used in the place specified below and cannot be used for other purposes such as filename.

The File Statement

This consists of two parts: the first describes the input file(s) and the second describes the output file. Multiple input files are used only for the merge, which is specified as a user-option (see later). A sort must have only one input file; a merge may have up to 16 input files.



The parentheses "(" and ")" may be replaced by the characters "<" and ">" respectively.

Rules for the file statement are as follows:

The medium for the specified input file(s) and output file is indicated by the keyword DISK. The absence of a disk-name (mfid) indicates the system disk. DISK refers to any kind of disk-device. The input file for a tagfile or keyfile creation must be on disk.

The PURGE option indicates that the input file(s) are to be purged after use.

The record size and records-per-block values are numeric values. When the input medium is DISK, the record size and records-per-block may be omitted. For a merge specification, input disk file descriptions with record size specifications may be interspersed with descriptions without such specifications.

If the records-per-block is omitted and record size is given, a blocking factor of 1 is assumed.

In all cases (except an index sort), input and output files must have the same record sizes.

The values of record size and records-per-block may be omitted for output files. For a sort, the values assumed are those of the input file. For a merge, the values assumed are those of the first specified input file.

For a keyfile creation sort, the output specification enclosed in parentheses must be the single word KEYFILE. The output will be on disk and record and block sizes are not user definable.

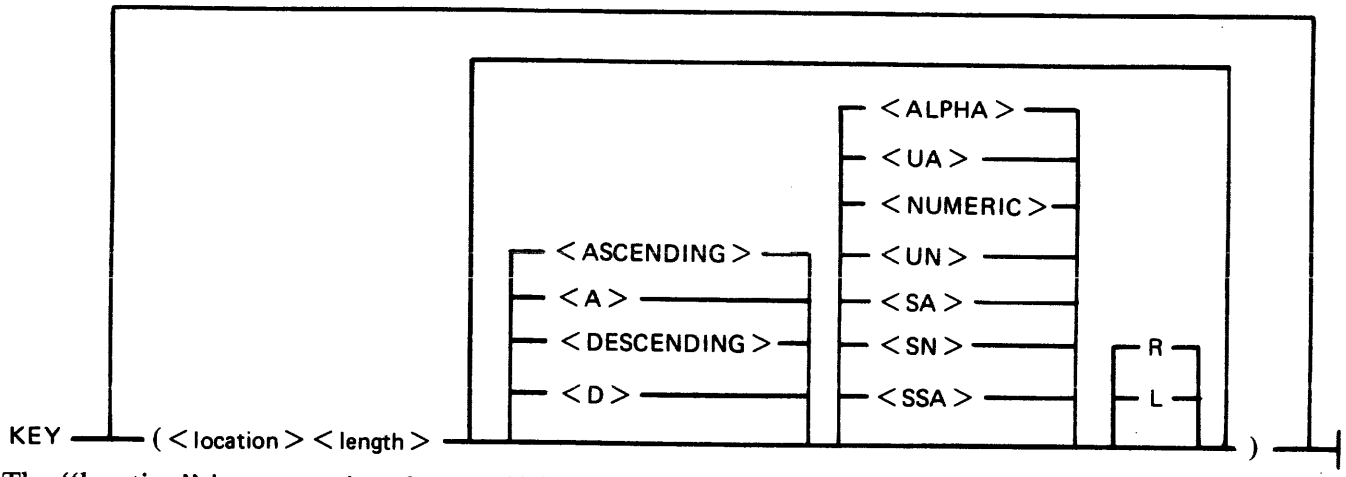
For a tagfile creation sort, the output specification enclosed in parentheses must be the single word TAGFILE. The output will be on disk and record and block sizes are not user definable.

The Key Statement

This statement defines the record key(s) that are used for the sort or merge.

A number of keys may be specified, each key description being enclosed in parentheses. The first key will be the major key and additional keys will be minor keys of decreasing significance.

Railroad diagram:



The "location" is a numeric value specifying the position of the key relative to the start of the record, in 4-bit units. The first 4-bit unit has a location of 1. The key location is given by the position of the left-hand 4-bit unit in the key (which, depending on the key format), may be a character or a sign. The key should start on a byte boundary unless a record sort with a numeric key is performed.

The "length" is a numeric value specifying the key length, in 4-bit units. This must include the sign, for signed keys.

The keywords ASCENDING and DESCENDING determine the order of collation. These keywords may be abbreviated to A and D respectively. If omitted, the default is ASCENDING. A keyfile sort can only be in ASCENDING order.

The format of the key is specified by one of the following keywords:

ALPHA or UA	–	unsigned 8-bit alphanumeric
NUMERIC or UN	–	unsigned 4-bit numeric
SA	–	signed 8-bit alphanumeric

SN – signed 4-bit numeric
SSA – 8-bit alphanumeric with separate sign

The default is ALPHA.

For a signed key, the position of the sign is specified by one of the following keywords:

R – right-hand (least significant) end of key
L – left-hand (most significant) end of key

The default is L.

For a description of key types and sign zone interpretation, see later under "KEYS".

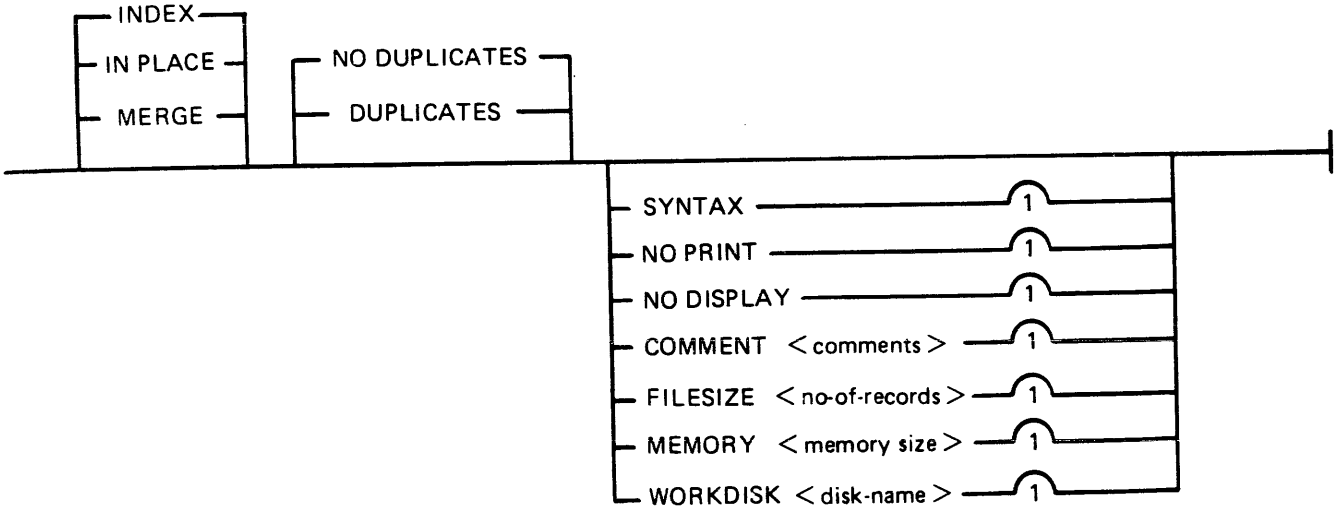
The User-Option Statement

These statements have three functions:

1. To specify which function is required.
2. To tailor a sort or merge to the particular machine configuration (for example, memory, printer availability).
3. To add comments.

The user-option statements are optional; if more than one are used, they may appear in any order relative to each other or to the file and key statements.

Railroad diagram:



The type of sort is given by one of the keywords INDEX, INPLACE, or MERGE. If one of these does not appear, a regular full record sort is assumed. The keyword INDEX specifies the creation of a keyfile or tagfile, depending on the output file details (see FILE statement). The keyword INPLACE specifies a full record sort using a minimal amount of disk work space. The keyword MERGE specifies a merge of several input files.

The keyword SYNTAX specifies that a check on the correctness of the sort statements is to be made without the sort actually being performed.

The keyword NOPRINT, if used in a starfile, stops the listing of the sort statements on the printer. If used, this keyword should be the first entry. If the statements are input via the SPO, they are not printed and so this keyword is not required in this case. The NOPRINT option also affects the printing of error and warning messages (see later).

The keyword NODISPLAY controls the display of messages on the SPO during the sort. This option can be used both in initiating messages and from file-oriented statements. It suppresses startup and termination messages. It does not affect the display of error and warning messages. Error-and warning-free sorts and merges will show no SPO activity if this option is used.

The keyword NODUPLICATES specifies that duplicate keys are not allowed in a keyfile creation. The keyword DUPLICATES specifies that duplicate keys are allowed in a keyfile creation. Both options are valid only when creating a keyfile. If neither is specified, the default is NODUPLICATES.

The keyword COMMENT introduces comment text. The end of the comment text is either the end of the input of the end of a record if the input comes from a starfile. Comments may appear between user-option statements and between file descriptions and key descriptions.

The keyword FILESIZE provides the following capabilities:

Specification of sort disk work space where the input file is not on disk (this use is not required if the input is from disk).
Specification of maximum size of the output file if on disk.
Allowance for future expansion of the output disk file where the sort/merge will not by default create a large enough file.

This keyword should be followed by a number giving the specified maximum number of records. For non-disk output files, the value is used for optimization purposes. If not used, default values are assumed where necessary. This option is not applicable to the inplace sort or to keyfile or tagfile creations.

The keyword MEMORY specifies the amount of non-overlayable work area to be used by the sort. This option is not applicable to the merge or to the inplace sort. If this is not enough for a successful sort, then this option is overridden. The memory size is in bytes; for example, MEMORY 1024.

The keyword WORKDISK enables the regular sort to utilize disk space in an efficient manner. It is not applicable to the merge or inplace sort. When the work-disk is specified, the sort locates up to half the work space on that disk, with the rest on the system disk. If this option is not used, but the input or output file is resident on a user disk, the work space is shared between that disk and the system disk. In all other cases, the work space is located entirely on the system disk. The named disk may be any type of disk applicable to the system in use.

Examples:

To sort the system disk file INP.FILE using the key starting at character 5 of length 3 characters, creating a system disk file OUT.FILE:

```
SORT FILE IN INP.FILE (DISK) OUT OUT.FILE (DISK) KEY (9 6)
```

To create a keyfile OUTKEY.FILE on disk PR2 from a data file INP.FILE1 on disk PR2, using a 5-byte key starting at the first byte:

```
SORT FILE IN PR2/INP.FILE1 (DISK)
    OUT PR2/OUTKEY.FILE (KEYFILE)
```

```
KEY (1 10)
INDEX
COMMENT DUPLICATES NOT ALLOWED
```

To merge the three system disk files FILE1, FILE2 and FILE3 into an output file MERGE.OUT:

```
SORT
FILE IN FILE1 (DISK) FILE2 (DISK) FILE3 (DISK)
OUT MERGE.OUT (DISK)
KEY (5 10)
MERGE
```

INTERACTIVE SORT

The SORT utility may be started from a datacomm terminal by entering <ncc> RN SORT, where <ncc> is the network control character. The datacomm terminal must be a TD 830 capable device.

Once started, the utility displays the first screen called "ROOT". On this screen the utility shows all the available facilities;

1: REGULAR SORT	Creates a new ordered output file.
2: INPLACE SORT	Reorders the input file (no new file created).
3: KEYFILE SORT	Creates a CMS keyfile (index contains key).
4: TAGFILE SORT	Creates a CMS tagfile (no keys in index).
5: MERGE	Merges up to 16 ordered input files.
6: RUN STAR FILE	runs a previously saved sort.
7: HELP	Provides assistance with SORT screen and gives extra information about the current screen.
8: EXIT	Terminates the sort.

The user is requested to enter a selection. Depending on the selection made, the utility will display another screen

If 1 (REGULAR SORT) or 2 (INPLACE SORT) is selected, the utility will display a screen on which the key and file specification may be entered. From this screen it is possible to request a second screen to enter more key specifications. After this screen, there may be a third screen to enter any user options.

If 3 (KEYFILE SORT) is selected, the utility will display a screen on which the required key and file specifications for the keyfile to be created may be entered. After this screen, there may be a second screen to enter any user options.

If 4 (TAGFILE SORT) is selected, the utility will display a screen on which the required key and file specifications for the tagfile to be created may be entered. After this screen there may be a second screen to enter any user options.

If 5 (MERGE) is selected, the utility will display a screen on which the first three file names and the key specification for the merge may be entered. If more than three files or more than one key specification are required the utility will display additional screens to enter the other names and key specifications, or to enter any user options.

If 6 (RUN STAR FILE) is selected, the utility will display a screen on which the name of the starfile to be created may be entered. On transmitting this screen with a sort command in the command selection, the utility starts searching the requested star file and executes the sort.

If 7 (HELP) is selected, the utility will display a screen with additional information about the SORT utility.

NOTE

All other screens also have HELP screens to supply additional information about the current screen.

If 8 (EXIT) is selected, the utility goes to End-Of-Job.

On all screens except the first, there are command selections requested. The possible commands with their explanations are:

1. HELP gives additional information about the options on the current screen.

2. Not available.

3. CONTINUE validates the input on the current screen and, if correct, it displays the next screen.

4. GO BACK. If the current screen is a non-help screen then the utility will return to the last non-help screen. It saves the input on the current screen but does not validate it.

5. SORT validates the input on the current screen and if correct and all necessary input has been provided, the utility zips the intrinsic.

6. SAVE displays a screen on which the name of a starfile to be created may be entered. The input on the current screen is not validated.

7. SORT & SAVE validates the input on the current screen and if correct and all necessary input has been provided, a screen is displayed on which the name of a starfile may be entered. Then the utility zips the intrinsic.

8. EXIT returns the utility to the ROOT screen.

NOTE

If a screen is disturbed, it may be recalled by entering "RECALL" in the top left corner of the screen and then transmitting.

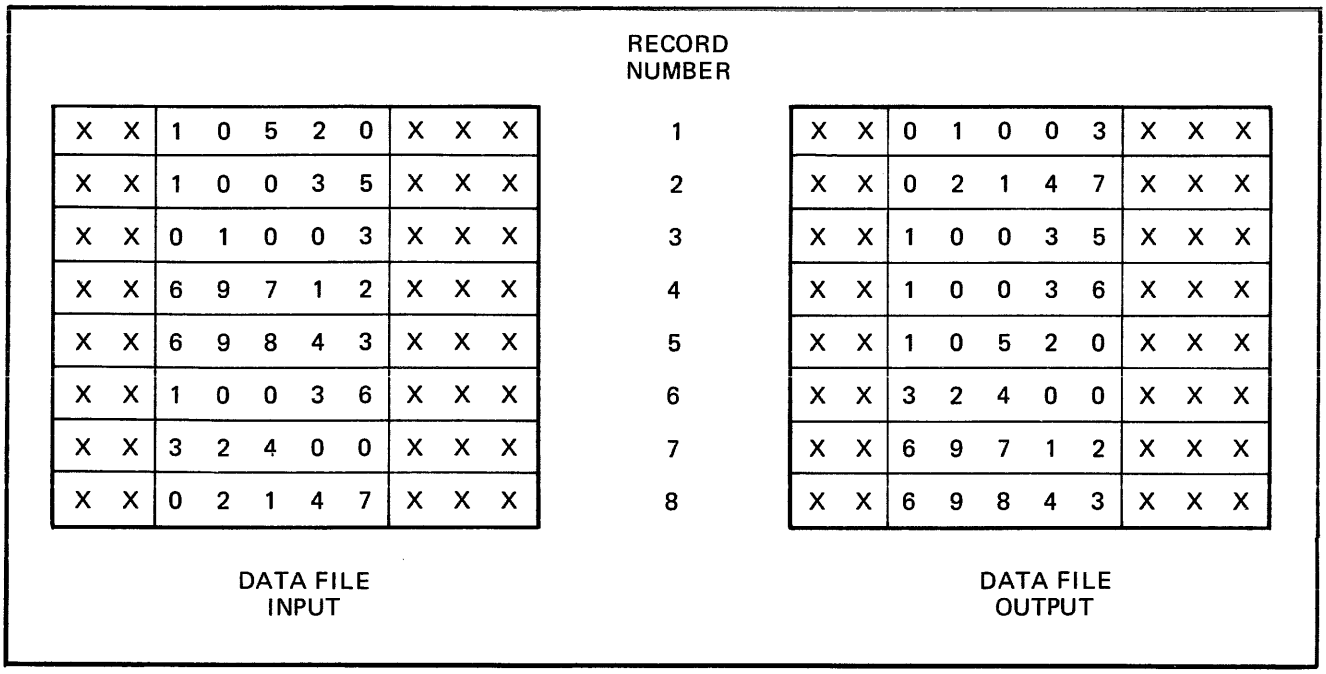
FUNCTIONAL DESCRIPTION

The five functions of the sort are described here:

1. Regular record sort.
2. Inplace record sort.
3. Keyfile creation.
4. Tagfile creation.
5. File merge.

Regular Record Sort

All the records contained within the specified input file are ordered using one or more keys. Deleted records (see later) are not included in the output file. See later for details of the keys. Refer to figure 5-1 for an example of a regular record sort, where the key is starting in byte 3 and is five characters long, and the sort is in ascending order. The X's refer to any other characters.



RECORD NUMBER	DATA FILE INPUT	DATA FILE OUTPUT
1	X X 1 0 5 2 0 X X X	X X 0 1 0 0 3 X X X
2	X X 1 0 0 3 5 X X X	X X 0 2 1 4 7 X X X
3	X X 0 1 0 0 3 X X X	X X 1 0 0 3 5 X X X
4	X X 6 9 7 1 2 X X X	X X 1 0 0 3 6 X X X
5	X X 6 9 8 4 3 X X X	X X 1 0 5 2 0 X X X
6	X X 1 0 0 3 6 X X X	X X 3 2 4 0 0 X X X
7	X X 3 2 4 0 0 X X X	X X 6 9 7 1 2 X X X
8	X X 0 2 1 4 7 X X X	X X 6 9 8 4 3 X X X

Figure 5-1. Regular Record Sort

The input file must be wholly contained on one hardware type, although it may be a dual-disk file. No other programs may write to this file during the execution of the sort.

The sort uses non-overlayable memory during execution. The amount is calculated according to the input file and key sizes. The amount may be specified as a user option, in which case the specified amount is used unless it is less than enough for a successful sort. In the latter case, the specified value will be overridden.

The sort uses disk work space, of up to 2.2 times the size of the specified input file. For the location of the work disk space, refer to the WORKDISK user option (see earlier). This work space is returned to the system at end-of-job.

Inplace Record Sort

This is the same as the regular record sort, except that the records are sorted within the input file. No new output file is created. The time taken is substantially greater than a regular sort, for the same input specifications. If deleted records are present in the file before the sort, they are removed: hence the number of records in the file may decrease after it has been ordered.

The inplace sort uses non-overlayable memory during execution. The size of this area cannot be specified at initiation.

The input file must be on disk. No other programs may access this file during execution of the inplace sort. The output file must be the same as the file specified for input.

If a particular system does not implement an in-place sort, a regular sort will be performed instead.

The inplace sort uses disk work space, of 0.2 to 0.3 times the size of the input file. When the input file is resident on a user disk, up to one-half of the work space is located on that disk, otherwise all work space is located on the system disk. This work space is returned to the system at end-of-job.

Keyfile Creation

A new file (the "keyfile") will be created containing one record for each record of the input file (the "data file"). The keyfile is sorted in order of the specified keys, and each keyfile record contains the key and a pointer to the corresponding record in the data file. Any deleted records in the data file are not referenced in the keyfile. Note that the records in the data file are not re-ordered and deleted records in the data file are not removed. Refer to figure 5-2 for an example of a keyfile creation, where the key is starting in byte 3 and is five characters long, and the sort is in ascending order. The X's refer to any other character.

Duplicate keys are not allowed unless specified (see the user-option statements DUPLICATES and NO-DUPLICATES). If they occur, then the record number is displayed on the SPO for each such occurrence, and the sort will continue but the output keyfile will be purged at end-of-job.

The keyfile creation uses disk work space, of up to 2.2 times the size of temporary file created by the sort in this case. This file is large enough to contain one record with the key value and record number for each record in the input file. For the location of the work disk space, refer to the WORK-DISK user option (see earlier). This work space is returned to the system at end-of-job.

Certain key values are not allowed during a keyfile creation. The key must not consist of all binary zeros, or must not contain any byte whose value is hex FF. If such a key is encountered, the record number is displayed on the SPO, and the sort will continue but the output keyfile will not refer to this record in the data file.

Tagfile Creation

A tagfile creation is similar to a keyfile creation, except that the output file contains only the record pointers, and not any key values. The tagfile records, however, are ordered in key value order, as specified by the sort. Any deleted records are not referenced in the tagfile. Refer to figure 5-3 for an example of a tagfile creation, corresponding to the keyfile creation in figure 5-2.

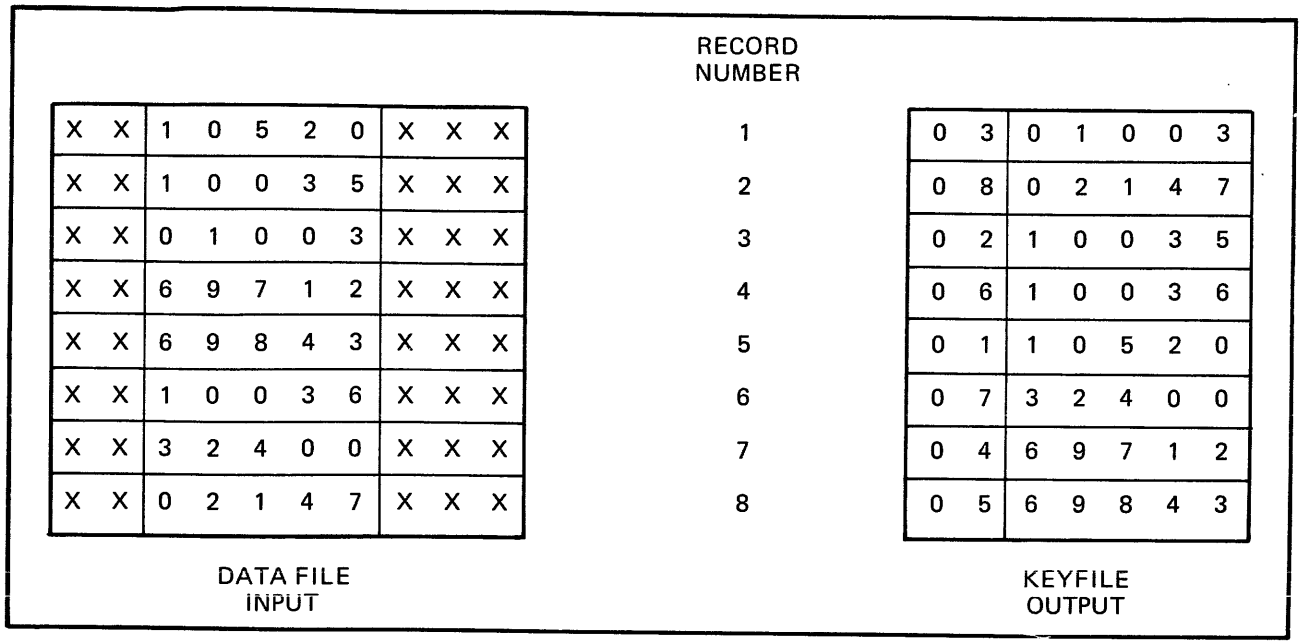


Figure 5-2. Keyfile Creation

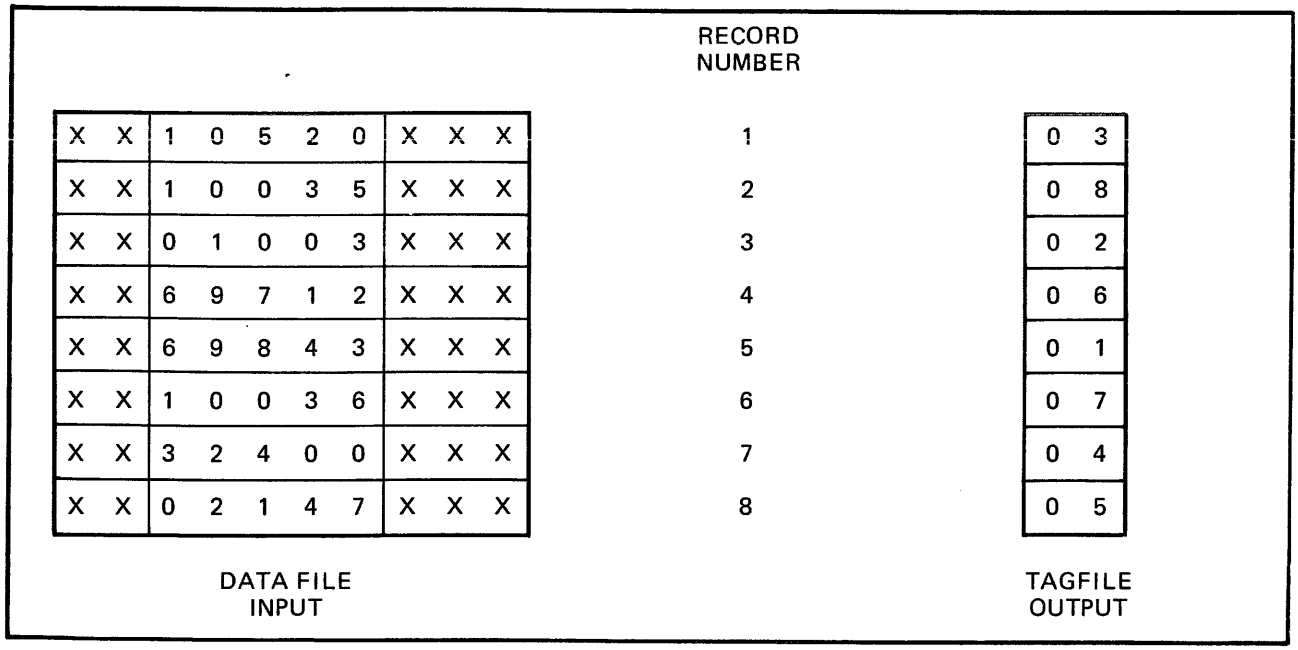


Figure 5-3. Tagfile Creation

A tagfile is a null keyfile. It is suitable for use as an ADDROUT file in RPG, and for limited indexed access in COBOL (the tagfile is read sequentially).

Disk space requirements are the same as for keyfile creation.

Merge

The merge merges up to 16 input files, using one or more specified keys, producing one output file. Deleted records in the input files are not included in the output file. If there are duplicate keys values, the order in which they are placed in the output file is given by the order in which the input files are specified.

Each input file must be wholly contained on one hardware type, although it may be a dual-disk file. No other programs may write to these files during the execution of the merge.

Each input file must have the same record size and the same position and length for each key. Each file must be already correctly ordered on the specified keys. If this is not the case, the merge will terminate prematurely after displaying a message on the SPO.

Refer to figure 5-4 for an example of a merge of two files, with a key starting at byte 3 which is five characters long. The X's and Y's refer to any character.

The merge uses non-overlayable memory during execution. The size of this area cannot be specified at initiation: it will be approximately equal to the sum of the block sizes of the input files and the output file.

The merge does not use any disk work space.

Details of Sort Keys

A "key" is the field within each record that is used for sorting or merging. If several distinct fields within a record are specified, then each field is a separate key. The relative order of importance of the keys is determined by the order in which they are specified. Figure 5-5 illustrates this with a two-key sort, using the KEY statement.

```
KEY (5 6 ALPHA) (15 2 DESCENDING ALPHA)
```

The X's indicate any character. In this example, the three-byte field is the major key, sorted in ascending order: the one-byte key is a minor key sorted in descending order within the order of the major key.

For a keyfile creation, only one key may be used. This key must be a maximum of 28 bytes long, must be a whole number of bytes in length, and must start on a byte boundary.

For all sorts except keyfile and tagfile creation, there can be up to 10 keys. The sum of the length of all keys (including signs) must be a maximum of 29 bytes.

The available key types are discussed here, under the keyword specified in the KEY statement (see earlier):

ALPHA (or UA)

Unsigned 8-bit alphanumeric field, containing ordinary ASCII characters. Note that this may consist of the 8-bit ASCII digits "0" to "9" but still be termed alphanumeric. This key type is the default.

NUMERIC (or UN)

Unsigned 4-bit numeric field, where each 4-bit unit is a binary coded decimal digit, 0000 to 1001 (0 to 9).

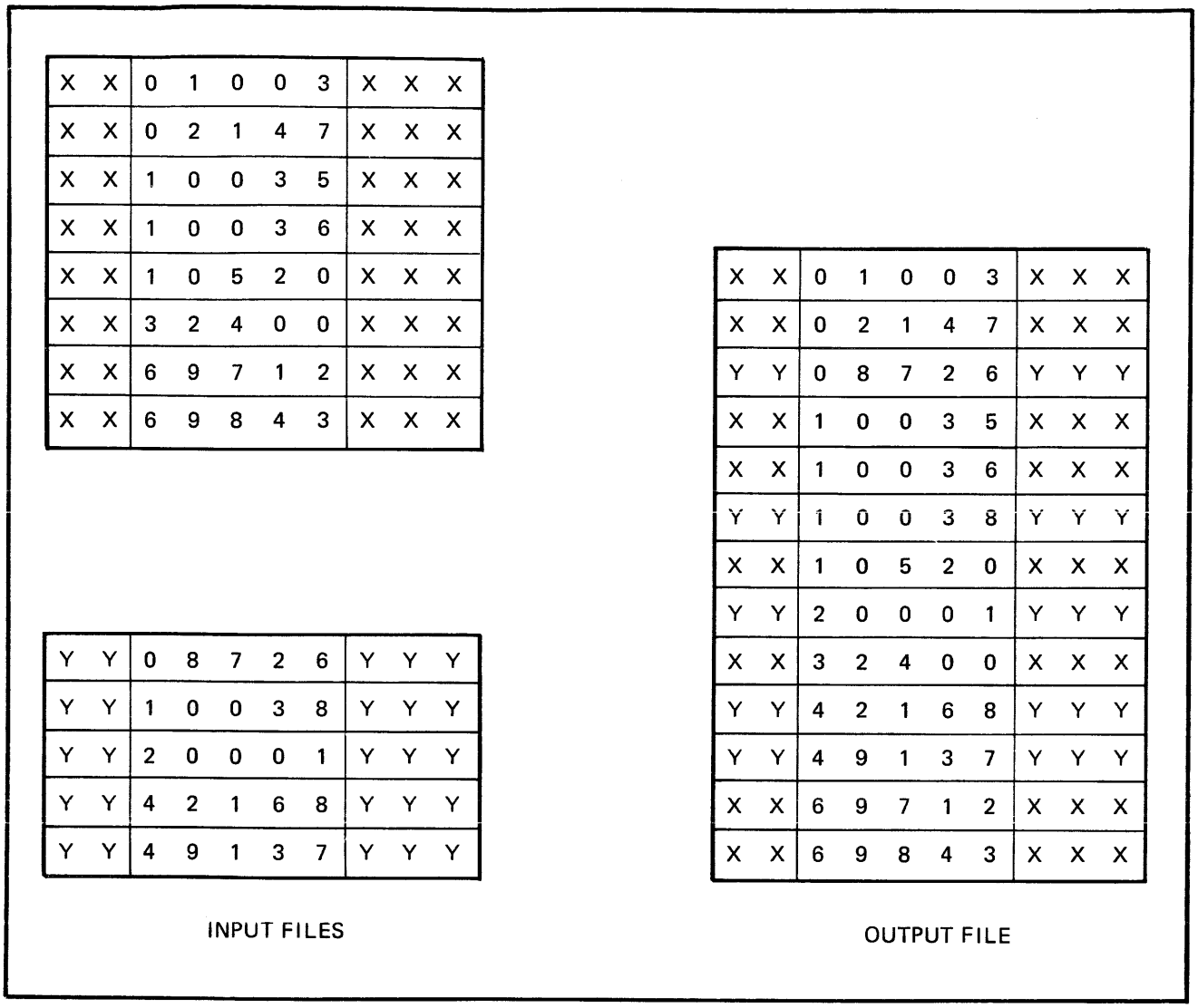


Figure 5-4. File Merge

SA

Signed 8-bit alphanumeric field. Each byte is an ordinary ASCII character (including the digits 0-9), except that either the first or the last character indicates the sign. Whether the sign is the first or last character is specified by the keyword L (left) or R (right). The default is L (first character; leading sign). The convention for coding the sign character is given in table 5-1. These characters are termed "overpunched signs" by analogy with historical punched card systems.

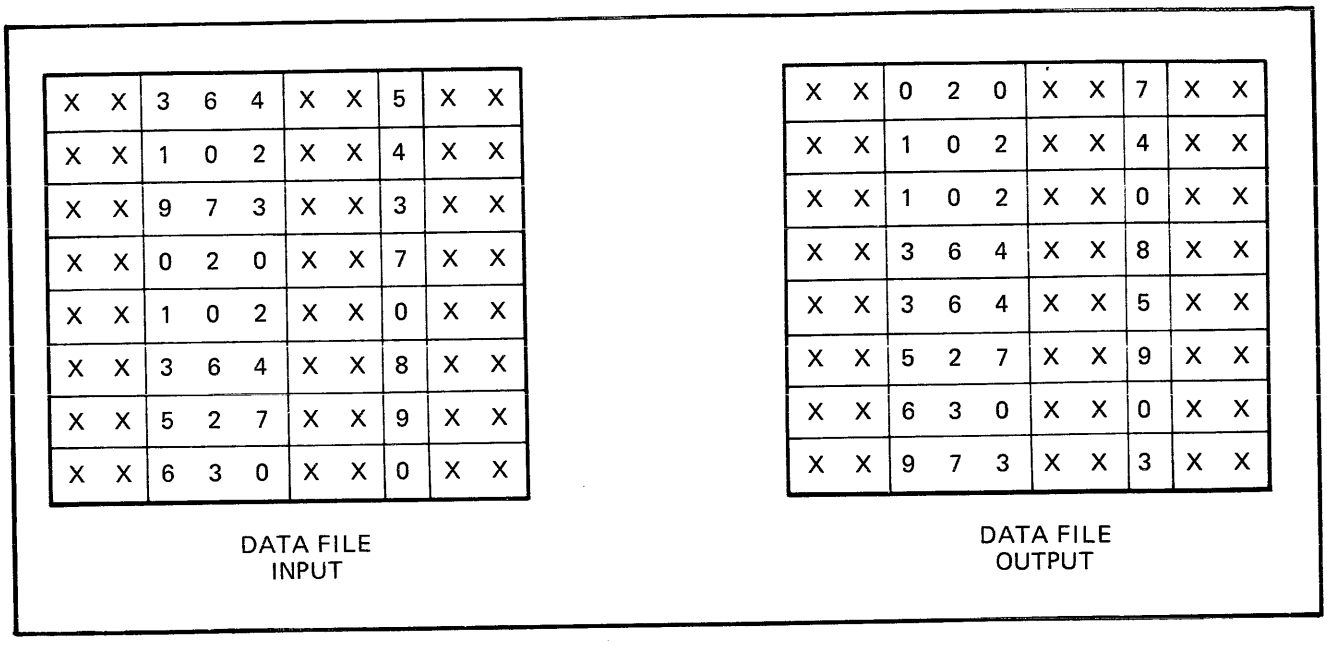


Figure 5-5. Multiple Key Sort

Table 5-1. Sign Convention for Signed 8-Bit Alphanumeric Fields

Key Value	Hex Code	ASCII Character
−0	2D	−
−0	7D	
+0	30	0
+0	7B	
−1	4A	J
−2	4B	K
−3	4C	L
−4	4D	M
−5	4E	N
−6	4F	O
−7	50	P
−8	51	Q
−9	52	R
+1	31	1
+2	32	2
+3	33	3
+4	34	4
+5	35	5
+6	36	6
+7	37	7
+8	38	8
+9	39	9

Note: Any other hex code in the sign character is interpreted as positive, with the key value given by the binary value of the right-hand 4 bits of the character.

SN

Signed 4-bit numeric field. Each 4-bit unit is a binary-coded decimal digit, 0000 to 1001 (0 to 9), except that either the first or the last 4-bit unit indicates the sign. Whether the sign is the first or last 4-bit unit is specified by the keyword L (left) or R (right). The default is L (first 4-bit unit); leading sign. The convention for coding the sign is given in table 5-2.

Table 5-2. Sign Convention for Signed 4-Bit Numeric Fields

Key Value	Binary Code (BCD Character)
negative	0101 (5)
positive	0011 (3)

Note: Any value other than 0101 (5) is interpreted as positive.

SSA

8-bit alphanumeric field with separate sign. Each byte is an ordinary ASCII character (including the digits 0 to 9), with the sign given by an ASCII character in either the first or last character. Whether the sign is given by the first or last character is specified by the keyword L (left) or R (right). The default is L (first character); leading sign. The convention for coding the sign character is given in table 5-3.

Table 5-3. Sign Convention for Separate Sign Character with 8-bit Alphanumeric Fields

Key Value	ASCII Character (hex value)
negative	"–" (2D)
positive	"+" (2B)

Note: Any character other than "–" is interpreted as positive.

The position of a sign within a signed key (left or right) must be the same throughout all occurrences of the key. Signed keys are ordered so that negative values come before zero and positive values.

8-bit keys may start on 4-bit unit boundaries, unless the separate sign type (SSA) is used, or the key is to be used in keyfile or tagfile creation.

Deleted Records

A deleted record is denoted by every byte in the record (including the key) containing the value hex FF. The action taken by the various sort options is discussed earlier. Deleted records may be physically removed by the FS utility.

Output Messages

Output messages cover warnings and errors. Messages are generated by both the sort intrinsic and the sort language processor. The intrinsic messages are numbered by event numbers in the same way as MCP output messages. The sort language processor messages are numbered in a similar way.

Messages can be divided by number as follows:

0-99

Sort language processor messages, displayed on the printer. Such messages appearing in the list below that are followed by a series of dots (...) should be read with the phrase NEAR COL XXX (with XXX replaced by an appropriate column number) in place of the dots.

0-34

Warnings, where corrective action is attempted.

35-39

Warnings, where no corrective action is attempted.

40-59

Errors in syntax (that is, the format of the sort statements is incorrect).

60-99

Errors in semantics (that is, an inconsistency has been detected in the statements, such as a key position greater than the record size).

170-200

Sort intrinsic messages, displayed on the SPO.

Certain messages may be suppressed by the NOPRINT and NODISPLAY keywords in the sort statements.

The NOPRINT option suppresses listing of the sort statements on the printer by the sort language processor. If this option is set, a maximum of five errors and four warning messages are directed to the SPO. The NOPRINT option has no effect on sort-intrinsic-generated messages.

The NODISPLAY option suppresses display on the SPO of start-up and termination messages by the sort intrinsic. Messages in the following list that are marked with an asterisk (*) are those that are suppressed when this option is set. Note that it is not possible to suppress individual messages; every applicable message is suppressed if the option is set. The NODISPLAY option has no effect on sort language processor messages.

Number	Message
0	EXPECTED SLASH NOT FOUND, "/" INSERTED...
1	EXTRA "FILE IN"...
2	MERGE INTRINSIC IGNORES <WORK-DISK OPTION>
3	OVERLENGTH PART OF <LABEL NAME> IGNORED...
4	INPLACE INTRINSIC IGNORES <WORK-DISK OPTION>
5	EXPECTED BRACKET NOT FOUND, "<" INSERTED...
6	<DUPLICATE OPTION> VALID IN INDEX-KEYFILE SORT ONLY
7	EXPECTED BRACKET NOT FOUND, ">" INSERTED...
8	ILLEGAL TO DELETE INPUT FILE, <PURGE OPT> IGNORED
9	OUTPUT BUFFER SIZE TOO BIG, <BLOCK FACTOR> REDUCED...
10	<USER OPTION> ALREADY INVOKED, LATEST USE...
11	MERGE <SORT TYPE OPTION> NOT SPECIFIED

Number	Message
12	OVERLENGTH PART OF <DISK NAME> IGNORED...
13	MISSING "FILE IN"...
14	INDEX <SORT TYPE OPTION> NOT SPECIFIED
15	EXTRA "KEY"...
16	<FILE SIZE OPT> VALID FOR MERGE/REGULAR SORT ONLY
17	MISSING "KEY"...
18	INPLACE INTRINSIC IGNORES <MEMORY OPTION>
19	<M-FILE/DP ID> IGNORED ON NON-MAGNETIC MEDIA FILE...
20	NUMBER TOO BIG, MAXIMUM VALUE ALLOWABLE ASSUMED...
21	not used
22	<SIGN POSITION> GIVEN FOR UNSIGNED KEY...
23	FIRST UNIT NUMBERED 0 RATHER THAN 1...
24	<FILE SIZE OPT> IGNORED SINCE OUT OF RANGE...
25	MERGE INTRINSIC IGNORES <MEMORY OPTION>
26	<BLOCK FACTOR> OF 0 NOT ALLOWED, 1 ASSUMED...
27	IN- AND OUT-FILE RECORD SIZES MADE EQUAL
28	<BLOCK FACTOR> TOO LARGE, MAXIMUM ASSUMED...
29	INPLACE SORT MUST HAVE IDENTICAL IN- AND OUT-FILES
35	IDENTICAL IN/OUT - FILES WILL PRODUCE DUPLICATE FILE
36	NOT NECESSARY TO PURGE CARD FILE...
37	ALPHANUMERIC KEY DOES NOT START ON BYTE BOUNDARY...
40	<KEY STATEMENT> ALREADY PROCESSED, NOW...
41	<DIGIT STRING> EXPECTED...
42	<CHARACTER STRING> EXPECTED...
43	<SEPARATOR STRING> EXPECTED...
44	<RCRD-BLCK PAIR> MUST BE GIVEN FOR NON-DISK IN-FILE...
50	NO <FILE STATEMENT> SPECIFIED
51	ILLEGAL WORD...
52	<LETTER STRING> EXPECTED...
53	MISSING <LABEL NAME>...
54	UNSUPPORTED <IN/OUT MEDIA>...
55	UNSUPPORTED <SORT TYPE OPTION>...
56	PART OF <FILE STATEMENT> MISSING, NOW...
57	NO <KEY STATEMENT> SPECIFIED
58	<FILE STATEMENT> ALREADY PROCESSED, NOW...
59	FINAL STATEMENT INCOMPLETE...
60	TOO MANY KEY SPECIFICATIONS...
61	TOO MANY FILE SPECIFICATIONS...
62	INPUT FILES RECORD SIZES NOT IDENTICAL...
63	<RECORD SIZE> OUT OF RANGE...
64	EXTRA DIGITS IN OVERLENGTH STRING IGNORED...
65	KEY LENGTH OUT OF RANGE...
66	MIN LENGTH OF SN KEY IS TWO 4-BIT UNITS...
67	BUFFER SIZE TOO LARGE...
68	DUPLICATE <IN-FILE PARAMS>, LATEST INSTANCE...
69	BUFFER SIZE TOO BIG FOR <IN/OUT MEDIA>...
70	ONLY ONE IN-FILE LEGAL FROM MULTIPLE TAPE...
71	MERGE INTRINSIC NEEDS AT LEAST 2 INPUT FILES
72	INDEX PARAM MUST BE "OUT...<KEYFILE/TAGFILE>"
73	KEY OVER-RUNS RECORD BOUNDARY
74	ILLEGAL TO OVERWRITE INPUT FILE WITH TAG/KEY FILE
75	ALPHANUMERIC KEY LENGTH NOT EVEN NUMBER OF 4-BITS...

(continued)

Number	Message
76	<MEDIA> MUST BE DISK FOR IN-PLACE SORT
77	IN- AND OUT-FILE RECORD SIZES MUST BE IDENTICAL
78	INDEX-KEYFILE KEY LENGTH NOT EVEN NUMBER OF 4-BITS
79	ONLY ONE KEY LEGAL IN INDEX-KEYFILE SORT
80	INDEX-KEYFILE SORT KEY TOO LONG
81	INDEX-KEYFILE SORT KEY MUST BE "... A UA/UN>"
82	ONLY INDEX SORT CAN SPECIFY "KEYFILE/TAGFILE"
83	INDEX-KEYFILE SORT KEY MUST START ON BYTE BOUNDARY
84	MIN LENGTH OF SSA KEY IS FOUR 4-BIT UNITS...
85	SSA KEY MUST START ON BYTE BOUNDARY...
86	CURRENT SUM OF KEY LENGTHS OUT OF RANGE...
170*	DUPLICATE RECORD <record number>
171	ILLEGAL INDEX KEY IN RECORD <record number>
172	RECORDS LOST / GAINED BY SORT-MERGE
173	<number> DUPLICATE RECORDS
174	<number> RECORDS CONTAINING INVALID INDEX KEYS
175*	<number> DELETED RECORDS
176*	<number> RECORDS MERGED
177*	<number> FILES MERGED
178	SORT-MERGE OUTPUT FILE NOT CREATED
179	SORT-MERGE ABNORMAL EOJ
180	SORT-MERGE SOFTWARE ERROR
181*	<number> RECORDS REFERENCED BY KEYFILE/TAGFILE
182	NO INITIATING MESSAGE
183*	<number> RECORDS SORTED
184	FILE ERROR <<number>> NEAR RECORD <record number> ON <file name>
185	UNORDERED MERGE INPUT FILE <file name> NEAR RECORD <record number>
186	TOO MANY RECORDS FOR SORT-MERGE
187	DUPLICATE RECORDS-KEYFILE NOT BUILT
188	INITIATING MESSAGE INVALID
189*	SORT-MERGE VER x.y.z INITIATED FROM <mix number>/<program name>
193	INPUT RECORD SIZES UNEQUAL - BAD FILE <filename>
194	IN/OUT RECORD SIZES BAD - OUTPUT SIZE CHANGED
195	BAD RECORD/BLOCK SIZE FOR OUTPUT DEVICE
196	KEY OVER-RUNS RECORD END
197	CANNOT SPLIT INDEX FILE
198	<number> PARITY BLOCKS
199	INDEX INPUT FILE NOT TYPE DATA

Message 184 represents differing file errors depending on the value of <number>. Defined meanings are as follows:

1 – EOF on output file
2 – parity on input file
3 – EOF on sort workfile
4 – parity on output file
5 – sort workfile error
6 – input file error
7 – output file error
8 – parity on sort workfile
9 – parity on input file (block ignored)

Section 6 Missing From Original Document

SECTION 7
CMS SYSTEM SOFTWARE OUTPUT MESSAGES

INTRODUCTION

This section describes all the messages generated by the MCP, COBOL interpreter, MPL interpreter, NDL interpreter and SORTINTRINS. The messages are grouped by error type, for example all the datacomm error messages are grouped together. In each group the messages are presented in the same order as they appear in the SYSLANGUAGE file.

This section is divided into two subsections:

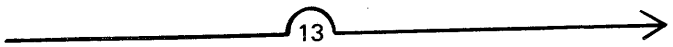
A – OUTPUT MESSAGES FOR B 90/B 900/CP 9500 SYSTEMS
B – OUTPUT MESSAGES FOR B 1800/B 1900 SYSTEMS

NOTE

Not all messages will be displayed with the corresponding event number.

SYNTAX RULES USED IN THIS SECTION

A syntax diagram is constructed of words formed of upper-and lower-case letters, arrows, special characters and digits. The basic rule is that any path traced along the forward directions of the arrows will produce a syntactically valid statement. All words formed of upper-case letters (except ASCII) and all special characters (for example commas, colons, hyphens, slashes) in the diagram will appear in the message as shown (acute parentheses are special characters when they are underlined). All words formed of lower-case letters are syntactic variables representing system supplied information. Any "bridge" over a number, such as



may be traversed the maximum number of times specified by the digit (thirteen times in this example).

Definition of Syntactic Variables Used

mix	mix number.
packid	Disk name or pack identifier. From 1 to 7 characters.
mfid	Multiple file identification. This may be diskid or the name of a tape or cassette. From 1 to 7 characters.
fileid	File identification. The name of the file. From 1 to 12 characters. If the file is a card or printer file, the name is truncated to 7 characters.
progname	Program name. From 1 to 12 characters.
peripheral	A three character mnemonic of which the first two characters describe the hardware device type and the final character describes the unit within that type. Units are specified by characters: A, B, C ..., for example DKC – disk cartridge, unit C.
device specifier	Either a three character mnemonic specifying a particular device type and unit, as for peripheral, or a two character device type specifying the device family. For example DK – all disk cartridges.
faulty input	The whole of the input message up to and including the first element detected to be erroneous. This forms part of many output messages.

device status	The current status of the device. This will take the form of the response to an OL intrinsic described in section 3.
serial no	The six digit serial number of a disk, or the five character serial number of a tape.
reel	The tape reel number in a multi reel tape.
number	An integer number.
hex-number	A hexadecimal number.
priority	The priority of a program (A, B or C).
event no	The event number of the message.
filename	The mfid/fileid specified in the program's FPB. For indexed files, the fileid referred to is the keyfile of the key/data pair, unless otherwise stated.

NOTE

For certain files the presence of the filename is only required at close time. In this case, the filename displayed may not correspond to the filename which might be expected.

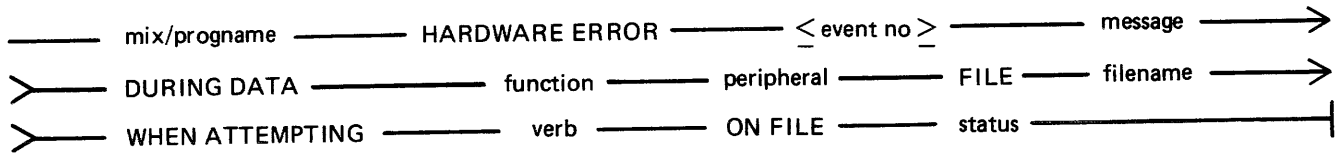
SUBSECTION A
OUTPUT MESSAGES FOR B 90/B 900/CP 9500 SYSTEMS

Events 1-5

Software Information

These messages are information indicating error conditions. If any operator action is required, other message(s) will immediately follow.

Message Syntax:



Where:

message	given in the table below
function	"INPUT FROM" or "OUTPUT TO"
verb	specifies the communicate that was being performed when the error occurred – for example READ, WRITE.
status	"DISK ADDRESS address" or "null"

NOTE

If there is an input/output error on a disk device during a hardware search, the disk address given is not valid, it shows where the search started from. The error is somewhere beyond this point.

Event	Message	Possible Causes	Suggested Action
1	DATA FORMAT CORRUPTION	Ending label is missing or invalid.	None. Program has indicated it can handle the error.
2	PARITY	Hard parity error found on this device.	None. Program has indicated it can handle the error.
3	DEVICE MALFUNCTION	For tape device, no data was found for quite a distance on the tape (exact length depends on unit). For disk, cylinder specified could not be used.	None. Program has indicated it can handle the error.
4	DATA NOT LOCATED	Sector could not be found on disk (cylinder was found). Software has indicated an invalid sector or sector address on disk is corrupt.	None. Program has indicated it can handle the error.

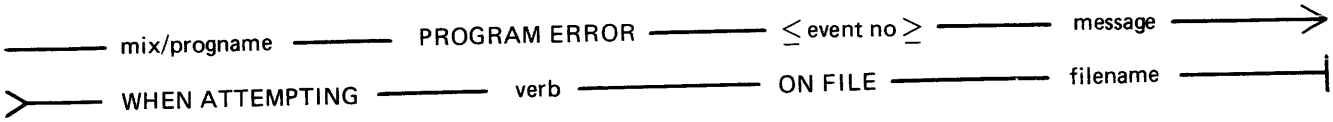
(continued)

Event	Message	Possible Causes	Suggested Action
5 MEDIA WRITE PROTECTED		The device is not write enabled.	Write enable the device and retry program.

Events 6, 22-27, 29-37, 40-41, 69, 79-81, 84, 87, 89-90

Invalid Requests on Class A or B Communicate to MCP (1)

Message Syntax:



These messages normally indicate program errors. The program in error should be DS'ed or DP'ed (see DS or DP intrinsics), if necessary. The operator should then attempt to run the program again. If the same error is encountered, request technical assistance.

Event	Message	Possible Causes	Suggested Action
6	DUAL PACK FILES NO LONGER SUPPORTED	Use of a disk pack file is no longer supported.	
22	UNEXPECTED REMOVAL OF DEVICE	The medium which contained the file has been illegally, physically removed when it was still required.	Replace the medium into the same drive it was removed from and ready the drive using the RY intrinsic.
23	FILE WAS OPENED WITH ATTRIBUTES THAT ARE INCOMPATIBLE	The attributes specified by the program are conflicting. For example a write to an input only file.	
24	INVALID SEQUENCE OF FILE OPERATIONS	The program has attempted to perform an invalid sequence of communicates, such as: 1. A rewrite not immediately preceded by a successful read. 2. An overwrite immediately preceded by an open or start communicate. 3. An overwrite or rewrite preceded by a conditional read which failed.	
25	SPECIFIED RECORD WORK AREA UNUSABLE	The work area segment specified within the FIB cannot be used, either because it indicates an FIB segment or it is a read only segment; but the communicate requests a data transfer to that segment.	

(continued)

Event	Message	Possible Causes	Suggested Action
26	ILLEGAL INDEX KEY (VALUE ZERO OR CONTAINING @FF@) USED	The key provided on a write communicate to an indexed file was equal to binary zero or a byte of @FF@.	
27	REQUEST FOR NON-EXISTENT DEVICE	There is no device of the required kind present on the system.	Do not use the program requesting this device.
29	DUAL PACK FILE INCONSISTENCY	The two parts of a dual pack file have been found to be inconsistent.	Check both disks contain the correct version of the file, replacing the file if necessary.
30	FILE ALREADY OPEN	The communicate requested an open of a already open file.	
31	FILE ALREADY CLOSED	The communicate requested a close of an already closed file.	
32	FPB OPEN OPTIONS INCOMPATIBLE OR INVALID	The adverb to open was determined to be illegal for any of the following reasons: 1. Myuse equal to zero. 2. Myuse incompatible with device, for example input/output for line printer. 3. Access mode random for non-disk device. 4. Access mode not equal to sequential, random or stream. 5. Shared access mode not random or sequential.	
33	INVALID RECORD OR BLOCK SIZE SPECIFIED	The record and/or block size has been determined to be incompatible or illegal for any of the following reasons: 1. Buffer or record length equal to zero for new disk or tape files. 2. Record length exceeds physical block size. 3. Buffer length not an integer multiple of record length.	
34	MAXIMUM FILESIZE EXCEEDED	The maximum file size exceeds 1048576 – 16 * number of sectors in an allocation unit, or 65536 – number of sectors in	

(continued)

Event	Message	Possible Causes	Suggested Action
		one allocation unit if a single area file.	
35	MORE THAN 16 BUFFERS REQUESTED	The number of buffers specified exceeds 16.	
36	INVALID DEVICE REQUESTED	The device requested is not supported by the CMS operating systems.	
37	WRONG FILE TYPE SPECIFIED	A mismatch between the file type fields of the FPB and the DFH has been found during an open, or user attempted to access a protected file.	
40	Invalid INDEX KEY SIZE SPECIFIED	An attempt has been made to open an indexed file with key size greater than 28, or with a key size equal to zero and access not sequential input, or input/ output.	
41	INDEX KEYFILE INCONSISTENCY	The implementation number in a key file KFPB is greater than this implementation (this is currently 1).	
69	A DISK FILE HEADER MISMATCH DUE TO CORRUPTION WAS DETECTED	There is an area-assignment mismatch on an extendable pseudo pack.	
79	INDEX KEYFILE CORRUPTION	The MCP has detected an inconsistency in the format of a key file during an indexed open.	
80	INDEX KEYFILE NOT WHOLLY IN DATA RECORD	The MCP has detected that the key for an indexed file is not wholly contained within the data record.	
81	PART OF FILE MISSING	The file containing the second half of a dual pack file or the file containing the data file of an indexed pair cannot be located. The operating system will cancel the suspension when the file is made available.	Make the required file available to the system.

(continued)

Event	Message	Possible Causes	Suggested Action
84	INVALID USE OF COMMUNICATE	A valid communicate was issued to the wrong device type. For example, test status ,which is valid for any disk file or ICMD file, and unconditional variant valid for tape file, so if either of these is issued to a card reader, this error would produced. Error may also be issued for other reasons.	
87	ACCESS BEYOND END OF FILE	A sequential read has been attempted after notification of end of file.	
89	FPB AND TAPE NAME MISMATCH	The tape currently in use has the wrong name, perhaps due to wrong reel being mounted.	
90	TAPE LABEL AND FPB ATTRIBUTE MISMATCH	Tape is not of the requested format.	

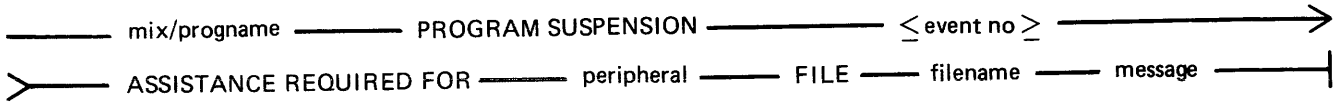
Events 10-19, 28, 39, 42, 82, 85, 88

Software Suspensions

When a running program encounters a condition that prevents it from continuing, the MCP suspends the program and informs the operator as to the reason for the suspension. When the condition is cleared, the MCP normally allows the program to continue running. If the program does not continue automatically, the operator should issue a "GO" command (see "GO" intrinsic in section 3) to the program.

All these messages requests the use of the "HELP" command (see "HELP" intrinsic in section 3) for further details.

Message Syntax:



Event	Message	Possible Causes	Suggested Action
10	FILE DOES NOT EXIST ON peripheral	Disk or tape file this program needs has not been found.	Check for correct disk or tape; supply program with backup copy of file requested.
11	THIS FILE ALREADY EXISTS ON peripheral	While attempting to place a certain file on disk or tape, program has discovered a file with the same name already exists on the device. The program halts as more than one file with the same name cannot reside on the same device.	Normally, use RM to remove the existing file from disk. If in doubt refer to program instructions. For non-disk devices, either power off, or save the unwanted device.
12	number DISK SECTORS NEEDED FOR FILE	The space available on disk is insufficient to hold the file this program is attempting to write. Or disk is checkerboarded.	Remove with RM any unnecessary files and as soon as there are sufficient sectors available on the disk the program will continue. Or, if the disk is checkerboarded, terminate the suspended program using the DS intrinsic, then use the SQ utility to consolidate disk space. Then re-execute the program.
13	NO FREE DISK DIRECTORY AVAILABLE	When the disk was initialized, the disk directory was constructed to contain a fixed number of file names. The directory has now reached its capacity.	Remove with RM any unnecessary files and program will continue. Or, DS the suspended program, replace disk with another disk having sufficient directory space, and re-execute the program.
14	peripheral NOT CURRENTLY AVAILABLE	Device that program needs in order to continue processing is either saved, not ready or in use by another program.	RY required device, or assign to alternative device using the AD intrinsic.

(continued)

Event	Message	Possible Causes	Suggested Action
15	SPECIAL peripheral FORMS REQUESTED	Program is waiting for operator to insert correct forms into the output device before it will continue processing.	Insert correct forms into output device. Then use AD intrinsic to assign device to program.
16	INDEX KEYFILE-DATAFILE GENERATION NUMBER DISCREPANCY – RECREATE KEYFILE	While opening an old indexed file, a discrepancy has been found between the generation fields of the keyfile and the data file. The operating system will cancel the suspension when the discrepancy no longer exists.	DS the program and either run the "KEY.CHECK" utility, or SORT the data file to produce a new keyfile. Then re-execute.
17	peripheral NOT ONLINE	Program needs a disk or tape that is not available to the system.	Mount the correct media and ready the device using the RY intrinsic. Or allow program currently using the device to go to end of job and then the suspended program will automatically be able to use the device.
18	MULTIPLE DEVICES OF THE NAME SPECIFIED ARE ONLINE	Two or more disks or tapes having the same names have been found on-line. Only one disk or tape of a given name may be on-line at a time.	Check for correct disk or tape, replace or relabel one of the duplicates.
19	ACCESS DENIED UNTIL OTHER PROGRAMS CEASE USING FILE – PLEASE WAIT	The specified file cannot be opened because it has already been opened by the maximum number of users allowed for the desired mode of use.	Wait until the file is not in use.
28	USE "AD" COMMAND TO ASSIGN DEVICE	An open of an unlabelled file requires a device to be assigned.	Assign the required device using the AD intrinsic.
39	INVALID FILENAME – USE "SF" COMMAND TO CORRECT NAME	OPEN or CLOSE has detected an illegal special character embedded in the filename.	Use the SF intrinsic if implemented.

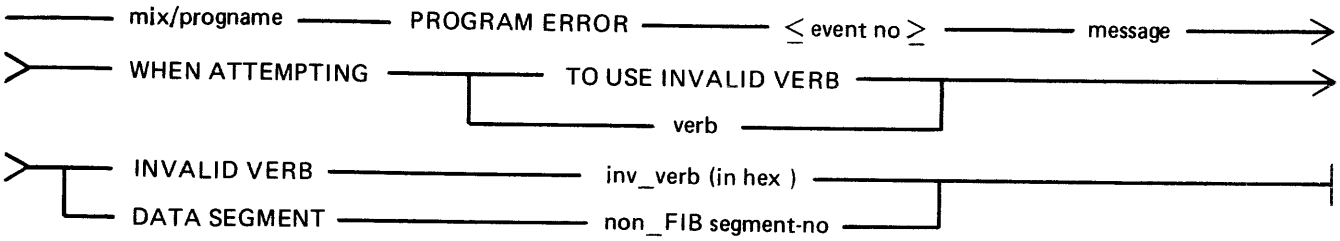
(continued)

Event	Message	Possible Causes	Suggested Action
42	ANOTHER PROGRAM EXCLUSIVELY USING DISK – PLEASE WAIT	The disk requested by the program is in use by another program which has "locked" access of the disk.	Wait for the disk to become available.
82	DISK CONTAINING PART OF FILE MISSING	The disk containing the second half of a dual pack file or containing the data file of an indexed pair cannot be located. The operating system will cancel the suspension when the file is made available.	Ready the required disk using the RY intrinsic or load the required disk onto the system.
85	FILE NOT FOUND ON TAPE – TRY ANOTHER TAPE	The tape does not have the required file.	Replace the tape with one which has the required file. Then assign the drive to the program using the AD intrinsic.
88	TAPE IS WRITE PROTECTED	The tape is write disabled.	Replace the tape with a write enabled tape. Then assign the drive to the program using the AD intrinsic.

Events 20-21

Invalid Requests on Class A or B Communicate to MCP (2)

Message Syntax:



These messages normally indicate program errors. The program in error should be DS'ed or DP'ed (see DS or DP intrinsics), if necessary. The operator should then attempt to run the program again. If the same error is encountered, request technical assistance.

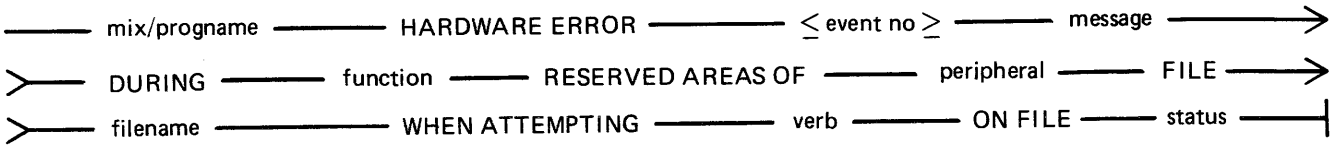
Event	Message	Possible Causes	Suggested Action
20	no message	An illegal verb has been used in a communicate. The "inv. verb" field contains the hexadecimal value of the illegal verb.	
21	WRONG DATA SEGMENT SUPPLIED	The data segment table did not contain an index to a valid FIB. The "non-FIB segment-no" field contains the segment number in decimal.	

Events 43-48

Device Errors

These error messages indicate terminations due to fatal device errors. These errors are normally encountered when attempting to read disks or magnetic tapes. The program which encountered these errors should be DS'ed or DP'ed (see DS and DP intrinsics). The operator should then attempt to run the program again. If the same error is encountered, the disk or tape involved should not be used again until field engineering has been notified and the media have been checked.

Message Syntax:



Where:

function	"INPUT FROM" or "OUTPUT TO"
verb	specifies the communicate that was being performed when the error occurred – for example READ, WRITE.
status	"DISK ADDRESS address" or "null".

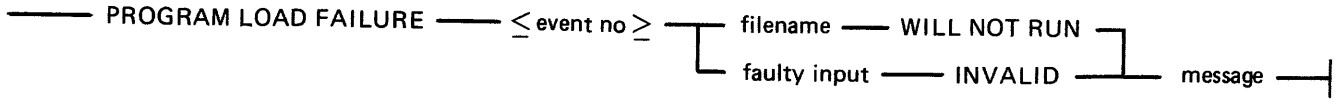
Event	Message	Possible Causes	Suggested Action
43	END OF MEDIA	An unexpected end of the tape has been encountered.	
44	DATA FORMAT CORRUPTION	The end label is missing or a labelled tape is invalid.	
45	PARITY	A hard parity error was discovered by the device during label processing.	
46	DEVICE MALFUNCTION	When reading a tape device no data was found for quite a considerable distance along the tape, the exact length is device dependent. For disk, the sector specified could not be found.	
47	DATA NOT LOCATED	The specified cylinder was found but the required sector was not found. Either the software has indicated an invalid sector or the disk/sector address is corrupt.	
48	MEDIA WRITE PROTECTED	The peripheral is found to be write inhibited.	

Events 49-68, 76-78

Loader Detected Failures

These messages indicate that the MCP failed to begin the processing of a particular program for some reason. The operator should correct the condition which caused the load failure and then try running the program again.

Message Syntax:



Event	Message	Possible Causes	Suggested Action
49	PROGRAM TYPE NOT SUPPORTED	This type of program is not supported by the system.	
50	DISK MISSING	The specified disk is either not ready, or not on-line.	Check that the correct disk has been loaded. Ready the disk with the RY intrinsic if it is not ready.
51	PROGRAM FILE MISSING	The specified program file was not found on disk.	Check that the correct program and disk name were entered and re-input if necessary. Ensure the disk contains the program required.
52	NO FREE MIX SLOT	The program specified cannot be run at the present time for one of the following reasons: 1. No mix slots are available for a program in this priority class. 2. Another program present will not allow any other programs to enter the mix.	Wait for a suitable time to execute the program, or perform an orderly termination of some of the current programs.
53	INSUFFICIENT FREE DISK SPACE AVAILABLE	There is not enough disk space for the program's Virtual Memory file (VMFILE) or the file directory of the disk has no entries for the creation of new files.	Make disk space available for program by either removing unwanted files from the disk, or copying files not required at the moment to backup, then remove them from the disk.
54	INTERPRETER FILE MISSING	The required interpreter is not on the system disk with its executing name.	Ensure that the necessary interpreter (for example, BILINTERP, COBOLINT, SORTINTRINS) is on the system disk with its executing name.
55	OTHER USERS DENYING ACCESS TO	A request to exceed the maximum number of concurrent users of a file has been attempted. The maximum	Wait until one or more of the programs currently using the file terminates.

(continued)

Event	Message	Possible Causes	Suggested Action
	PROGRAM	number of concurrent users will not exceed seven.	
56	PROGRAM FILE CORRUPT	The file type for the file specified was not a code file type. For example, the file was data or key file type.	Check input and re-enter.
57	UNABLE TO DETERMINE PROGRAM NAME DUE TO ERROR IN LOAD REQUEST	The program name specified in EX command was invalid. For example it contained too many characters.	Check input and re-enter.
58	INSUFFICIENT MEMORY AVAILABLE	There is not enough memory space to hold this program's Task Control Block (TCB) and its Program Control Block (PCB).	Wait for other programs to terminate. If this condition persists, request technical assistance.
59	MAXIMUM ALLOWED MCS PROGRAMS ALREADY RUNNING	The maximum number of MCS programs is already reached.	None.
60	DUPLICATE DISKS ONLINE	Two or more disks with the same name are online. The system requires all disks on the system to have a unique name.	Remove unnecessary pack or rename one of the disks using the RL intrinsic.
61	OTHER PROGRAMS RUNNING	The specified program may only run in a mix in which only the datacomm, SYS-SUPERUTL and SPIM (if running) are present. ALSO SYS-SUPERUTL must be in a non-executable state. executable state.	Wait for the mix to become suitable for executing this program.
62	NDL PROGRAMS CANNOT BE LOADED DIRECTLY	An NDL task can only be loaded as part of the data comm loading sequence. Any other attempt to load an NDL task will fail.	Use the required NDL loading sequence.
63	DISK HARDWARE ERROR ENCOUNTERED	There has been an irrecoverable disk error while attempting to load a program.	Use backup copy of the required program. If the problem continues, request technical assistance.
64	SUPERUTILITY FUNCTION ALREADY	An attempt has been made to use one of the functions of SYS-SUPERUTL while this	Wait until the current SYS-SUPERUTL function has completed before entering the required function.

(continued)

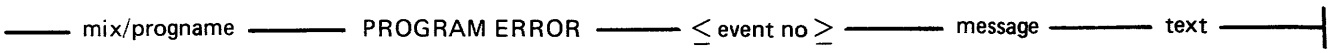
Event	Message	Possible Causes	Suggested Action
	RUNNING	utility is performing another function. SYS-SUPERUTL functions are IR, LB, LF, CH, KX, PD and RM.	
65	INSUFFICIENT MEMORY REQUESTED	The program structure cannot be constructed within the amount of memory specified in the real store field in the the EX intrinsic.	
66	DUAL ALPHABET/ REVERSE ESCAPEMENT PRINTER FEATURES NOT SUPPORTED ON THIS SYSTEM	The program requires the interpreter and MCP to support dual alphabet or reverse escapement and the current system does not.	Do not use this program with this implementation of CMS.
67	ERROR IN FILE NAME EQUATION	The file equate in the EX command referenced an internal file which did not exist.	Check input and re-enter the execute command (EX intrinsic).
68	ANOTHER PROGRAM HAS EXCLUSIVE CONTROL OF SPECIFIED DISK	Another program has locked the required disk.	Wait until the program holding the lock on the required disk terminates, then re-execute the program.
76	INTERPRETER VERSION AND CURRENT MCP INCOMPATIBLE	The interpreter and the MCP levels are incompatible.	Copy the required interpreter from backup media, then re-execute the program.
77	IT IS NOT A PROGRAM FILE	The file type for the file specified was not a program file type.	Check input and re-enter.
78	SPECIFIED MCS ALREADY RUNNING	The MCS specified has already started. Enter "MX" for details.	

Events 70-75

Run Structure Problems

These messages are output to indicate failure on execution of a program. The program encountering these errors should be DS'ed or DP'ed (see DS DP intrinsics) if necessary. The operator should then attempt to run the program again. If the same error is encountered, request technical assistance.

Message Syntax:



Where "text" is:

70, 71, 73	"PROBABLE COMPILER/INTERPRETER FAULT"
72, 74	"INSUFFICIENT RESOURCES REQUESTED"
75	"AMEND PROGRAM"

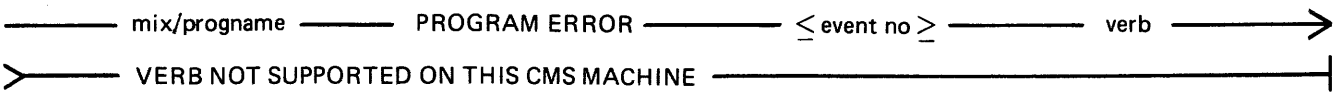
Event	Message	Possible Causes	Suggested Action
70	ACCESS ATTEMPTED TO NON-EXISTENT SEGMENT	The segment number exceeds the number of segments declared in the program.	
71	ACCESS ATTEMPTED BEYOND END OF SEGMENT	The length of the segment exceeds the declared size of the segment.	
72	ACCESS ATTEMPTED BEYOND END OF CONTROL STACK	The amount of control stack declared in the program has been exceeded during execution.	
73	ACCESS ATTEMPTED TO NON-EXISTENT DATA ON CONTROL STACK	The specified program has attempted to retrieve more information from the control stack than is present.	
74	INSUFFICIENT MEMORY REQUESTED AT PROGRAM LOAD	The program has attempted to exceed the memory size specified at execution (load) time in the real store field. Refer to EX intrinsic.	
75	PROGRAM REQUIRES MORE MEMORY THAN SYSTEM CAN PROVIDE	The program has attempted to exceed the physical memory size.	

Event 83

Unimplemented Communicates

This message is output to indicate that this CMS machine does not support the referenced communicate.

Message Syntax:

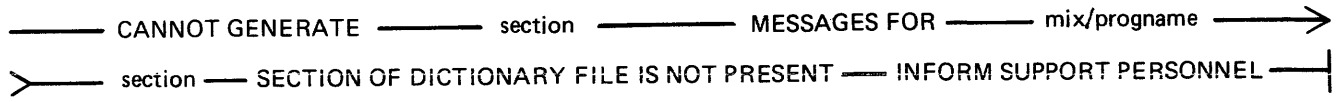


Event 91

Message Dictionary Section

This message is output to indicate that the MCP cannot generate the requested message because it has not found the appropriate section in the dictionary file.

Message Syntax:

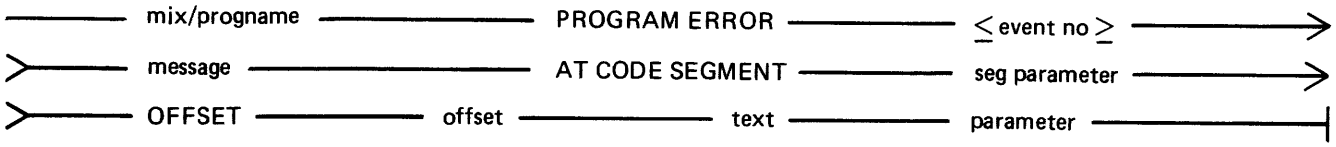


Events 100-169

Interpreter Detected Errors

These messages indicate that a program interpreter has encountered an error on execution of the program. The program in error should be DP'ed (see DP intrinsic) if necessary. The source program, object program, dumpfile produced by DP and any data files used by the program should be saved. The operator should then try to run the program again. If the same error is encountered, request technical assistance and supply all relevant data saved to the technician.

Message Formats:



100-105 "PROC/DEBUG LINE" for MPL and COBOL interpreter detected errors.
110-132 "PROCEDURE" for MPL-interpreter detected errors.
140-169 "DEBUG LINE" for COBOL-interpreter detected errors.

Events 100-105 MPL/COBOL Interpreter Detected Errors

Event	Message	Possible Causes	Suggested Action
100	COMMUNICATE ERROR	MCP returned @80@ in Byte 0 of Fetch Message on a communicate.	
101	COMMUNICATE EOF ERROR	MCP returned an End of File indication in Fetch message (@20 10 00@) and the user has not specified any action if EOF occurs.	
102	COMMUNICATE I/O ERROR	MCP returned an I/O error other than EOF (@20 10 00@) in Fetch message and user has not specified any action to be taken if error occurs.	
103	SEGMENT NUMBER ERROR	Interpreter detected an invalid data segment number.	
104	WRITE ERROR	Interpreter detected an attempted WRITE to a Read-Only Segment or, if COBOL, a literal.	

(continued)

Event	Message	Possible Causes	Suggested Action
105	SEGMENT BOUNDARY VIOLATION	Interpreter, on trying to resolve an address, has discovered that the address of the data or code is out of range.	

Events 106-109 reserved for future use.

Events 110-132 MPL Interpreter Detected Errors

Event	Message	Possible Causes	Suggested Action
110	INVALID OP	Code file has become corrupt or, an error exists in MPLII compiler or interpreter, or an old level of interpreter with same feature not implemented.	
Events 111-114 reserved for future use.			
115	DESCRIPTOR ACCESS	Program tried to store the fetch value to a non-character field; to convert to a non-character field; or to store to a self-relative descriptor. Or, an error occurred in the SETNAME procedure, or in the parameter in decimal arithmetic. Other possible errors are: key not character (8) in authenticate, message not character in authenticate, or monitor parameter out of range.	
116	SEGMENT SIZE ERROR	The length assigned at run time to a segment or segmap page must be less than equal to the maximum length specified, and the segment must not have been used by the program before the SEGMENT/SEGMAP statement has been encountered at run-time.	
117	ADDRESS ERROR	SETNAME extent error; identifier has become out of scope.	
118	MESSAGE REFERENCE ERROR	Size of message reference field not divisible by 4, or illegal access to message reference field.	
119	STRING STORAGE ERROR	Illegal destination in store string instruction.	
120	REMAP ERROR	Program tried to re-map a bit descriptor.	
121	SUBSTRING ERROR	Attempt made to sub-string into a non-character area, or length of substring greater than 255, or substring not wholly contained in area being substringed.	

(continued)

Event	Message	Possible Causes	Suggested Action
122	INDEX ERROR	Program tried to index a self-relative descriptor, or to bit index a self-relative descriptor.	
123	EXIT ERROR	Local data returned from a function.	
124	CPA ERROR	Error in communicate parameter area: for example, message reference expected, or character field of 3 bytes expected.	
125	DIVIDE ERROR	Divide by zero attempted.	
126	ZIP ERROR	Error returned after ZIP (not used on B 80).	
127	BIT DESCRIPTION ERROR	Bit field overlaps more than one byte boundary.	
128	FPB ERROR	Error in file parameter block encountered, or subfield of a non-FPB segment requested.	
129	CONTROL STACK ERROR	Control Stack overflow in communicate instruction, or control stack underflow in exit instruction.	
130	DATA STACK ERROR	Illegal descriptor encountered. Usually caused by overwriting descriptors on the data stack by data in an assignment statement, where the left hand side has a bad offset value. For example A(X):=VAL; where X is too big – this will write beyond the end of A and may corrupt descriptors.	
131	DECLARATION MODE ERROR	Size of character field greater than 255.	
132	DATA STRUCTURE ERROR	Insufficient room for structure nesting or size of character field greater than 255, or insufficient room for array bound evaluation, or non-character descriptor encountered when character descriptor expected.	

Events 133-139 reserved for future use.

Events 140-169 COBOL Interpreter Detected Errors

Event	Message	Possible Causes	Suggested Action
140	INVALID S-OP CODE	Invalid S-Op-Code. Code file is corrupt or error exists in COBOL/RPG compiler or interpreter.	
141	INVALID COPX – GREATER THAN SIZE OF COP TABLE	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
142	ALPHANUMERIC FIELD TYPE NOT 8-BIT UNSIGNED	Corrupt code file or error exists in COBOL/RPG RPG compiler or interpreter.	
143	INVALID EDIT MICRO OPERATOR	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
144	INLINE EDIT MASK NOT CORRECTLY TERMINATED	Corrupt code file or error exists in COBOL/RPG RPG compiler or interpreter.	
145	EXAMINE SOURCE FIELD ERROR	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
146	EXAMINE PARAMETER FIELD NOT 8-BIT UNSIGNED, ONE CHARACTER	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
147	EXAMINE CONTROL BYTE ERROR	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
148	COMPARE FOR CLASS – CLASS AND FIELD TYPE INCOMPATIBLE	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
149	SUBSCRIPTED OR INDEXED SUBSCRIPTED OR INDEX	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
150	INDEXED/SUBSCRIPT VARIABLE IS INDEXED/ SUBSCRIPTED BY MORE THAN 3 VARIABLES	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
151	FETCH COMMUNICATE RESPONSE FIELD NOT OF LENGTH 3 BYTES	Corrupt code file or error exists in COBOL/RPG compiler or interpreter. exists in COBOL/ RPG RPG compiler or interpreter.	
152	INVALID EXAMINE SPECIFICATION	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	

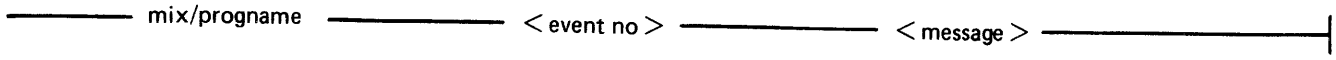
(continued)

Event	Message	Possible Causes	Suggested Action
Events 153-157 reserved for future use.			
158	PERFORM STACK UNDERFLOW	Attempt to exit from a module when not in a called module. Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
159	CANNOT OPEN NEW OUTPUT FILE WHEN VALUE OF FILESIZE IS ZERO	Program attempted to create a file on disk with a filesize of zero. Invalid code as no records could ever be output to the file created.	Perform suitable alterations to the program which allow the program to create a file with non-zero filesize.
160	PERFORM STACK OVERFLOW	Indicates that the number of nested PERFORM statements exceeds that allowed within the Perform Stack at compile time (if not specified then the default value is used). If this did not result from programming error, the Perform Stack should be increased.	The size of the Perform Stack may be increased by MODIFY utility, using CONTROL.STACK option to change the PPB (see section 4).
161	NON POSITIVE SUBSCRIPT	Subscripts must be a value greater than zero.	
162	ARRAY BOUND VIOLATION	Subscript outside the upper bound of OCCURS clause.	
Event 163 reserved for future use.			
164	TRANSLATION SOURCE ERROR	In an MPL TRANSLATE statement the translation text or translation table not of type character (1).	
165	NO STOP RUN ENCOUNTERED	End of code file has been encountered but no STOP RUN statement has been found.	
166	INVALID SIGN CODE		
167	I/O ERROR	Invalid READ/WRITE to a file encountered.	
168	SORT OR MERGE ERROR	An error has been encountered in SORT or MERGE.	
169	ZIP FAILURE	Zip of another task has failed. For example zipped program not present, zipped file is not a code-file.	

Events 170-199

Sort/Merge Information

Message Format:



The following messages may be output while SORT or SORTINTRINS is running. They will display the name of the program which executed or attempted to execute SORTINTRINS, the MCP event number and a message to explain why the event was displayed.

Event	Message	Possible Causes	Suggested Action
170	DUPLICATE RECORD record number	Only for keyfile creation. Another record in the file has the same key as this record.	If duplicates are desired, specify "DUPLICATES" on input.
171	ILLEGAL INDEX KEY IN RECORD key	Only for keyfile creation. The key field contains either all binary zeros or has one or more bytes with @FF@. This record will not be referenced from keyfile.	None.
172	RECORDS LOST/GAINED BY SORT-MERGE	Probably indicates an error in SORTINTRINS.	See introductory paragraph concerning errors.
173	number DUPLICATE RECORDS	Normal message. Tells the operator the total number of records that have duplicates. (See Event 170).	None.
174	number RECORDS CONTAINING INVALID INDEX KEYS	Normal message. Tells operator the total number of records with invalid index keys. (See Event 171).	None.
175	number DELETED RECORDS	Informs the operator that a number of records were deleted as they contained only hex FF in every byte.	None.
176	number RECORDS MERGED	Normal message for MERGE only. Tells the operator the total number of records merged from all files.	None.
177	number FILES MERGED	Normal message for MERGE only. Tells the total number of files merged. Should be the same as the number requested in the Sort Spec.	None.

(continued)

Event	Message	Possible Causes	Suggested Action
178	SORT-MERGE OUTPUT FILE NOT CREATED	SORTINTRINS was DS'ed. May indicate corrupt SORT or SORTINTRINS programs.	See introductory paragraph concerning SORT errors.
179	SORT-MERGE ABNORMAL EOJ	Early termination due to errors.	
180	SORT-MERGE SOFTWARE ERROR <SEG number OFFSET number>	Error in SORTINTRINS.	See introductory paragraph concerning SORT errors.
181	number RECORDS REFERENCED BY KEYFILE/ TAGFILE	Only for keyfile/tagfile creation. Tells the number of records referenced by a keyfile/tagfile.	None.
182	NO INITIATING MESSAGE	SORT intrinsic requires a properly coded initiating message. This should be properly formatted by SORT or Sorts within programming languages such as COBOL. Probably indicates an attempt to execute SORTINTRINS directly.	None.
183	number RECORDS SORTED	Normal message. Tells operator the number of records sorted by a successful data file SORT.	None.
184	FILE ERROR <number> NEAR RECORD number ON filename	The message informs the operator that an error has been encountered within the specified file near the specified record number. The <number> means: 1. EOF on output file. 2. Parity error on input file. 3. EOF on Sort Workfile. 4. Bad disk address. 5. SORT workfile error. 6. Input file error. 7. Output file error. Except for 2, this probably indicates an error in SORTINTRINS.	See introductory paragraph concerning SORT errors.
185	UNORDERED MERGE INPUT FILE filename NEAR RECORD number	Files to be merged did not have an increasing/decreasing key value. Either the file is incorrect or the key position has been incorrectly specified.	SORT files for correct order, then retry.
186	TOO MANY RECORDS FOR SORT-MERGE	Machine dependent limitation.	See introductory paragraph concerning SORT errors.

(continued)

Event	Message	Possible Causes	Suggested Action
187	DUPLICATE RECORDS – KEYFILE NOT BUILT	DUPLICATES was not specified on input and they were found to exist. This message age will be displayed and no keyfile will be built.	Specify DUPLICATES on input.
188	INITIATING MESSAGE NOT VALID	Initiating message supplied to SORTINTRINS is not in proper format. This could be caused by a fault in the program that zipped the SORTINTRINS.	See introductory paragraph concerning SORT errors.
189	SORT-MERGE version INITIATED FROM mix number/program name	Normal message to inform the operator which version of the SORTINTRINS is in use and by which program.	None.
193	INPUT RECORD SIZES UNEQUAL – BAD FILE filename	For a merge sort, the input files do not have the same record size. The offending file is named.	None.
194	IN/OUT RECORD SIZES BAD – OUTPUT SIZE CHANGED	The output file has been specified with a record size different to that of the input file. This has been corrected automatically.	
195	BAD RECORD/BLOCK SIZE FOR OUTPUT DEVICE	The output file buffer size must be less than 32768 for disk file.	
196	KEY OVER-RUNS RECORD END	Key limit is beyond the end of record.	
197	CANNOT SPLIT INDEX FILE	An index file cannot reside on more than one disk.	
198	number PARITY BLOCKS	Indicates the number of blocks in which parity errors were found.	
199	INDEX SORT INPUT FILE NOT TYPE DATA	Input files must be of type data.	

Events 200-349

Data Communications Errors

These messages indicate invalid conditions encountered by the data communications sub-system. For full descriptions of these events see the CMS Data Communications Subsystems Reference Manual, form no 1090909.

Event Number	Breakdown
200-254	Common Datacomm Messages B 90/B 900/CP 9500
255-269	B 90 Dependent Datacomm Messages
300-305	B 900/CP 9500 Dependent Datacomm Messages

Events 200-229, 231-237

Datacomm Handler Detected Errors (For B 90/B 900/CP 9500)

Message Format:

Event	Message	Possible Causes	Suggested Action
200	OPERATION DENIED – INVALID MESSAGE TYPE	The type field in the message header contains a value greater than 12. Error is returned after QUEUE.	
201	OPERATION DENIED – INVALID LOGICAL STATION NUMBER	A reference has been made to an LSN greater than STATION.COUNT-1. Error is returned after ALLOW.OUTPUT CLEAR CONTINUE.STATION DISALLOW.OUTPUT QUEUE REDEFINE.STATION ROUTE.OUTPUT SET.INPUT.LIMIT STATION.DESCRIPTION STATION.STATUS	
202	OPERATION DENIED – INVALID QUEUE REFERENCE	A reference has been made to an invalid queue. Certain communicates are restricted to certain queues, therefore the queue referenced may exist while being invalid in this context. Error is returned after CLEAR DEQUEUE QUEUE QUEUE.DEPTH RECALL ROUTE.INPUT ROUTE.OUTPUT SET.QUEUE.LIMIT	
203	OPERATION DENIED – INVALID LOGICAL SUBNET NUMBER	A reference has been made to a subnet greater than SUBNET.COUNT-1. Error is returned after ALLOW.INPUT	

(continued)

Event	Message	Possible Causes	Suggested Action
		DISALLOW.INPUT SUBNET.DESCRIPTION SUBNET.STATIONS	
204	WRITE HEADER DENIED – TEXT LENGTH EXCEEDS ACQUIRED DATA COMM MESSAGE SPACE	The text length in the message header was set to a length greater than the message length field. Error is returned after WRITE.HEADER	
205	OPERATION DENIED – MESSAGE REFERENCE DOES NOT CONTAIN A MESSAGE POINTER	An attempt has been made to perform a function on a null message reference, that is, one which does not reference a message space. Error is returned after COPY.TEXT QUEUE READ.HEADER READ.TEXT WRITE.HEADER WRITE.TEXT	
206	WRITE TEXT DENIED – STARTING BYTE LIES OUTSIDE OF ACQUIRED DATA COMM MESSAGE SPACE	The starting byte index of a text transfer is illegal. For the source message the index must be less than text length. For the destination message the index must be less than message length. Error is returned after COPY.TEXT READ.TEXT WRITE.TEXT	
207	OPERATION DENIED – INVALID MIX NUMBER	The task number referenced is not currently in the mix or is outside the range of the mix table. Error is returned after ALLOW.INPUT ALLOW.OUTPUT CONTINUE.TASK DISALLOW.INPUT DISALLOW.OUTPUT SET.OUTPUT.LIMIT TASK.NAME TASK.STATUS	
208	OPERATION DENIED – INVALID LOGICAL LINE NUMBER	The LLN reference is greater than LINE.COUNT-1. Error is returned after QUEUE LINE.DESCRIPTION LINE.STATIONS	

(continued)

Event	Message	Possible Causes	Suggested Action
		LINE.STATUS REDEFINE.LINE REDEFINE.STATION	
209	OPERATION DENIED – INVALID MODEM NUMBER	A reference has been made to an LMN greater than MODEM.COUNT-1. Error is returned after MODEM.DESCRIPTION REDEFINE.LINE REDEFINE.STATION	
210	OPERATION DENIED – INVALID TERMINAL NUMBER	A reference has been made to an LTN greater than TERMINAL.COUNT-1. Error is returned after TERMINAL.DESCRIPTION REDEFINE.STATION	
211	OPERATION DENIED – INSUFFICIENT DATA COMM BUFFER SPACE AVAILABLE	No message space is available to execute the communicate. Error is returned after CLEAR RECALL	
212	REDEFINITION DENIED – STATION NOT ATTACHED	An attempt has been made to make an unattached station ready. Error is returned after QUEUE	
213	REQUESTED DATA COMM COMMUNICATE NOT IMPLEMENTED	The last communicate issued is not implemented on this CMS system.	
214	OPERATION DENIED – LIMIT NOT ALLOWED	A queue limit of zero has been specified. Error is returned after SET.INPUT.LIMIT SET.OUTPUT.LIMIT SET.QUEUE.LIMIT	
215	OPERATION DENIED – INVALID TALLY	A tally number not in the range 3-30 has been specified in the ACCESS.TALLIES COMMUNICATE	
216	OPERATION DENIED – STATION FORMAT DOES NOT PROVIDE REQUESTED TALLIES	The station specified in the ACCESS.TALLIES COMMUNICATE is not of type tallies.	
217	OPERATION DENIED – FUNCTION PERMITTED ONLY BY PRIVILEDGED MCS	An attempt was made to perform an operation which is not allowed by a non-privileged MCS.	

(continued)

Event	Message	Possible Causes	Suggested Action
218	REDEFINITION DENIED – SPO CAPABILITY CONFLICTS WITH OTHER ATTRIBUTES	A REDEFINE.STATION COMMUNICATE has been issued which would result in a station which is either 1. SPO capable and type bits 2. SPO capable but not myuse input and output	
219	REDEFINITION DENIED – STATION IS IN SCL MODE OR SCL MESSAGES ARE QUEUED	A REDEFINE.STATION COMMUNICATE has been issued while a SPO capable station is in SCL mode, or has SCL output messages queued.	
220	OPERATION DENIED – LOGICAL STATION ALREADY ATTACHED	The LLN of an attached station has been set to a value other than @FF@. Error is returned after REDEFINE.STATION	
221	REDEFINITION DENIED – INCOMPATIBLE ATTRIBUTES PROPOSED	The new attributes of the station or line are inconsistent with the existing network definition. Error is returned after REDEFINE.STATION REDEFINE.LINE	
222	OPERATION DENIED – DIRECT CONNECT LINE	An attempt has been made to assign a MODEM to a direct connect line. Error is returned after REDEFINE.LINE	
223	OPERATION DENIED – FULL DUPLEX MISMATCH	Attribute mismatch of full duplex terminal. Error is returned after REDEFINE.LINE REDEFINE.STATION	
224	OPERATION DENIED – INCOMPLETE VARIABLE	The length of the parameter area to be used for reconfiguration is insufficient. Error is returned after REDEFINE.LINE REDEFINE.STATION	
225	OPERATION DENIED – IMPROPER LINE CONDITION	The line being redefined is not in the required state of NOT READY and, for a switched line, not switched busy or not connected. Error is returned after REDEFINE.LINE	
226	OPERATION DENIED – MESSAGES QUEUED	Messages are queued for output to the station referenced by	

(continued)

Event	Message	Possible Causes	Suggested Action
		REDEFINE.STATION. Messages are queued for output to a station on the line referenced by REDEFINE.LINE. Error is returned after REDEFINE.STATION REDEFINE.LINE	
227	OPERATION DENIED – NO VACANCY ON LINE	The MAXSTATIONS statement in the line section of the NDL defines the maximum number of stations which may be attached to a particular line. An attempt has been made to attach a station to a line which already has MAXSTATIONS.	
228	OPERATION DENIED – SPEED MISMATCH	The speed specified for a station when either redefining the station or attaching the station to a line, does not match the speeds of the other stations on that line.	
229	OPERATION DENIED – QUEUE FULL	The MCS has attempted to queue a message which would cause the queue count field of the station or subnet table to overflow.	
231	OPERATION DENIED – OWNERSHIP IMPROPER FOR STATION OR SUBNET QUEUE ACCESS	The MCS tried to access a station or subnet which it does not own.	
232	LINE OR DCP ALTERATION DENIED – ALL ASSOCIATED STATIONS MUST BE OWNED BY THE ISSUING MCS	The MCS tried to attach a station to a switched line on which stations owned by a different MCS are already attached.	
233	OPERATION DENIED – NETWORK RECONFIGURATION IS IN PROGRESS	The following operations cannot be performed at this time because network reconfiguration is taking place: LINE STATIONS STATION STATUS	
234	RELINQUISH DENIED – IMPROPER STATION, LINE OR SUBNET QUEUE STATE	An attempt by an MCS to release control of a station cannot be accomplished for one of the following reasons: 1. The station or subnet to be relinquished is queued.	

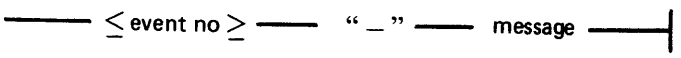
Event	Message	(continued) Possible Causes	Suggested Action
		2. The station is READY. 3. The station is attached to a switched line. 4. The mix number passed is not that of an MCS.	
235	OPERATION DENIED – SPECIFIED MIX NUMBER DOES NOT REFER TO AN MCS	An MCS has tried to relinquish a station/subnet to an MCS which is not currently in the mix.	
236	OPERATION DENIED – STATION INPUT OR OUTPUT IS NOT ROUTED TO THIS MCS	Invalid routing of message in multi MCS environment.	

Event 237 reserved for future use.

Events 230, 245-254

Datacomm Loader Detected Errors

Message Format:



Event	Message	Possible Causes	Suggested Action
230	DC LOAD FAILURE – NDLSYS FILE AND DCP CODEFILE LEVELS ARE INCOMPATIBLE	The data comm loader has detected an inconsistency between the NDL code file and the DCP code file. Possibly the DCP code file was not generated from this NDL code file.	
245	DC WARNING – SPO CAPABILITY OVERRIDDEN FOR STATIONS WITH INCOMPATIBLE ATTRIBUTES	The station is defined as both SPO = TRUE and is type BITS, or both SPO = TRUE and is not MYUSE INPUT and OUTPUT (See MTS guide).	
246	DC LOAD FAILURE – MCS IS NOT LISTED IN THE NDLSYS FILE	An attempt was made to load an MCS that has not been identified in the NDL "MCSLIST" statement.	
247	MCS LOAD FAILURE – THIS MCS WAS ALREADY LOADED	An attempt was made to load an MCS which has the same name as an MCS already executing.	
248	DC LOAD FAILURE – MCP VERSION AND DCP CODEFILE LEVELS ARE INCOMPATIBLE	The DCP codefile is not generated by an NPC900 running under this MCP level, so it does not match.	
249	DC LOAD FAILURE – NDLSYS FILE PRIORITY CLASS INVALID	The NDLSYS file does not have the correct value in the priority class field of the PPB (@3180@).	
250	DC SUBSYSTEM ERROR – HARDWARE ERROR DURING DISK ACCESS	The NDLSYS or DCP file cannot be read because of a disk I/O failure.	
251	DC LOAD FAILURE – CORRUPT NDLSYS FILE	The NDLSYS file either has a line with an invalid address for B 800 or specifies an amount of required memory which is insufficient for the tables and buffers declared.	
252	DC SUBSYSTEM ERROR – INSUFFICIENT DCP MEMORY	The memory space required field of the present data in the NDLSYS file specifies more	

Event	Message	(continued) Possible Causes	Suggested Action
		space than the MCP can provide.	
253	DC SUBSYSTEM ERROR – HARDWARE ERROR DURING CLOSE OF NDLSYS FILE	Performing the close communicate on the NDLSYS has failed.	
254	DC LOAD FAILURE – CANNOT OPEN NDLSYS FILE	Performing the open communicate on the NDLSYS has failed. For example: – File is not on disk. – File has wrong file type.	

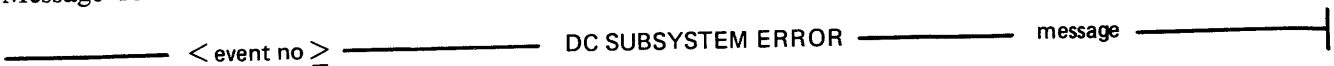
B 80/B 90 Dependent Datacomm Messages

Event 255

Event	Message	Possible Causes	Suggested Action
255	DC INVALID	The operator has entered a DC message when no MCS was running. This event does not set fetchvalue or fetchmessage.	

Events 256-263

Message format.



Event	Message	Possible Causes	Suggested Action
256	TOO MANY SUBNETS IN NDL	This event is returned and data comm load aborted if: (SUBNET.COUNT * 16) + 1 is greater than 2000. That is, if there is insufficient space for the number of subnet queues defined.	
257	TOO MANY STATIONS IN NDL	This event is returned and data comm load aborted if: (STATION.COUNT * 12) is greater than 2000. That is, if there is insufficient space for the number of stations defined.	
258	CANNOT ALLOCATE REQUIRED SPACE	This event is returned and data comm load aborted if there is insufficient space declared in the NDL preset data for the system queue header and at least one message. Insufficient space is declared to be less than (STATION.COUNT + SUBNET.COUNT + 2) * 12 + 176	
259	CANNOT EXECUTE NDL PROGRAM	This occurs if the load of the NDL interpreter was not caused by an MCS load.	
260	INVALID DATACOMM CONTROLLER	This occurs if the system detects that a line channel/subchannel does not contain a valid data comm controller, or if the transmission method of the controller is incompatible with that declared for the line in NDL (for example, an async controller declared as sync in NDL). The load of data comm is aborted.	
261	TOO MANY LINES IN NDL	Insufficient space has been allocated to the NDL interpreter by the MCP. Load of data comm is aborted.	

NOTE

The error messages for events 262 and 263 are outside the range of B 90-dependent errors. They refer to restrictions which will be lifted in the future.

Event	Message	Possible Cause	Suggested Action
262	FULL DUPLEX LINE NOT IMPLEMENTED	This event is returned and the load of data comm aborted if the NDLSYS contains a full duplex line.	
263	TELEX LINE NOT IMPLEMENTED	This event is returned and the load of data comm aborted if the NDLSYS contains a telex line.	

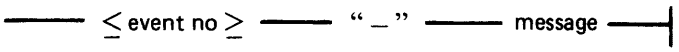
Event 269

Event	Message	Possible Causes	Suggested Action
269	DC LOAD FAILURE – CODEFILE WAS NOT GENERATED FROM THIS NDL	This NDL.INTERPX file is not the latest codefile to be generated by NPC90 from this NDLSYS.	

B 900/CP 9500 Dependent Datacomm Messages

Events 300-305

Message Format



Event	Message	Possible Causes	Suggested Action
300	DC LOAD FAILURE – INSUFFICIENT DATACOMM BUFFER MEMORY	The amount of memory specified in the SYSCONFIG file for use by data comm buffers is less than the minimum specified in the NDLSYS file.	
301	DC LOAD FAILURE – CANNOT OPEN SYSRECON FILE	The system encountered an error while attempting to open the SYSRECON file.	
302	DC LOAD FAILURE – CANNOT CLOSE SYRECON FILE	The system encountered an error while attempting to close the SYSRECON file.	
303	DC WARNING – LOGICAL DCP number NOT LOADED	This message will be displayed after detection of a DCP related load error, to indicate that the DCP is in error.	
304	DC LOAD FAILURE – NO DATACOMM PROCESSORS LOADED	This message is displayed if no DCPs have been loaded. This condition is fatal to the data comm load.	
305	DC SUBSYSTEM ERROR – NDLSYS FILE LINE ADDRESS IS NOT SPECIFIED CORRECTLY FOR THIS SYSTEM	Line address in NDL is different from the hardware line address.	

Events 350-499

Implementation Dependent System Messages

These messages may or may not be output depending on the specific CMS system being used.

Unless otherwise shown in the message, these messages do not output an event number. The event number shown in the column called EVENT is the message's relative position in the MCP section of SYSLANGUAGE.

Events 350-369 – Common Messages

These messages may be output from any CMS operating system.

Event	Message	Possible Causes	Suggested Action
350	DISK pack-id IN DRIVE peripheral IS CORRUPT CEASE USING DISK	Label and directory information could not be read at disk load time.	Try re-inserting disk, if the same error occurs, stop using this disk.
351	DISK pack-id IN DRIVE peripheral SHOULD BE BACKED UP SOON DUE TO EXCESSIVE ERRORS	An excessive number of retries is occurring on this disk.	Copy all required files from this disk and re-initialize the disk.
352	DISK pack-id IN DRIVE peripheral AT END-OF-LIFE IF REMOVABLE, REPLACE DISK IF FIXED, INFORM SUPPORT PERSONNEL	The hardware has automatically removed as many bad sectors as it has the capacity for.	Replace the disk if it is a floppy disk, otherwise ask for technical assistance.
353	DISK pack-id IN DRIVE peripheral NEARING END OF LIFE. BACKUP IMMEDIATELY.	The hardware will shortly have no more room left in which to relocate bad sectors.	Copy all required files from this disk. If this is a floppy disk, replace it, otherwise ask for technical assistance.
354	REMOVABLE DISK IN DRIVE peripheral NOT CENTRED CORRECTLY (MTR DATA = number) RE-INSERT DISK	Drive failed to recalibrate due to circumferential accuracy.	Remove and re-insert disk.
355	DISK IN DRIVE peripheral CANNOT BE ACCESSED DUE TO EITHER A DRIVE FAILURE OR CORRUPTED DISK (MTR DATA = number)	Drive failed to recalibrate due to positional tribit accuracy.	Try another disk in the drive with the reported failure. If the drive operates with no reported failure, retry the original disk.
356	DISK DRIVE peripheral FAILURE (MTR DATA = number) CEASE USING DRIVE AND INFORM SUPPORT PERSONNEL	The drive failed read/write confidence test.	Ask for technical assistance.

(continued)

Event	Message	Possible Causes	Suggested Action
357	DISK IN DRIVE peripheral MAY HAVE BEEN CORRUPTED WHILST WRITING (MTR DATA = number) CHECK CONTENTS OF DISK	Drive failed to: 1. Recognize any sector identifiers. 2. Detect end of sequence.	Use CHECK.DISK and KA to check contents of the disk.
358	INVALID CONTROL CARD IN READER peripheral. REMOVE AND REPUNCH CARD	A program from a card-reader cannot be executed due to an invalid control card.	Remove and repunch the control card.
359	DISK IN DRIVE peripheral IS NOT IN STANDARD CMS FORMAT	The disk label may be corrupted.	Reformat or re-initialize the disk.
364	WARNING: mix/progname HAS ACCESSED file-name WHICH IS A DUAL PACK FILE. SUCH FILES WILL NO LONGER BE SUPPORTED IN FUTURE RELEASES USE "COPY" TO CONVERT FILE.	In future releases dual pack files will no longer be supported, so it would be better to eliminate them now.	Use COPY utility to convert the file, or run DUMPADISK.
365	TOO MANY SMALL UNUSED AREAS ON DISK pack-id/BECOMING AVAILABLE RESULTING IN DIRECTORY OVERFLOW AND LOST SPACE. RUN "SQ" SOON TO RECOVER DISK AREAS AND CONSOLIDATE INTO LARGER FREE AREAS.	It is possible for all entries in available table (non-file directory) to be in use. If a file then releases its disk space, it may not be possible to enter it in the available table. Disk checkerboarded.	Run SQ utility.
366	DEVICE peripheral LOCKED BY mix/ progname. USE "RY" TO BRING IT ON LINE WHEN REQUIRED	A non disk device is closed with lock.	Run RY intrinsic to make ready the peripheral.
367	mix/progname UNABLE TO PURGE WRITE-PROTECTED TAPE IN DEVICE peripheral	An attempt has been made to purge a write disabled magnetic tape device.	Write enable the media.

Events 370-399 – B 90 Dependent Messages

Event	Message
370	ENTER DATE AS MM/DD/YY
371	ENTER DATE AND TIME AS MM/DD/YY HHMM
373	mix/progname priority status
374	* * *

Events 430-469, 630-634 – B 900 System Dependent Messages

Event	Message
430	filename CREATED
431	SYSTEM DUMP INITIATED
432	SYSTEM DUMP ABORTED AN AVAILABLE DUMP FILE DOES NOT EXIST
433	SYSTEM DUMP ABORTED DISK ERRORS WERE ENCOUNTERED
434	filename CLOSED
435	SYSDMFILE NOT CREATED DISK ERRORS WERE ENCOUNTERED ON OPEN
436	SYSDMFILE NOT CREATED INSUFFICIENT DISK SPACE
437	SYSDMFILE NOT CREATED MAXIMUM NUMBER OF SYSTEM DUMP FILES ALREADY EXISTS
438	ENTER DATE AND TIME USING DT COMMAND
439	SYSTEM I/O ERROR WHEN ACCESSING A PROGRAM STRUCTURE FOR mix/progname
440	CLEAR/START hexnumber
441	MEMORY PARITY ERROR DETECTED AT: hexnumbers
442	decnumber text decnumber
443	decnumber DCP decnumber
444	mix/progname KEYBUILD NOT FOUND
445	mix/progname MERGER NOT FOUND
446	mix/progname ALL INPUT RECORDS DELETED
447	PROCESSOR decnumber LOGICALLY NOT READY
448	mix/progname REQUIRES AN INTERPRETER WHICH IS NOT PRESENT IN ANY OF THE PROCESSORS
449	mix/progname HAS REQUESTED MORE MEMORY THAN IS AVAILABLE ON THE APPROPRIATE TASK PROCESSORS
450	mix/progname SWAPPED OUT. THRASHING DETECTED
451	mix/progname EXECUTING
452	input IGNORED – DEVICE ERROR event-no ON peripheral – THE LP TRANSLATION TABLE text SPECIFIED IN THE SYSCONFIG FILE CANNOT BE LOCATED IN FILE SYSTRANSLATE
453	input IGNORED – DEVICE ERROR event-no ON peripheral – THE REQUESTED TRANSLATION TABLE text SPECIFIED IN THE SYSCONFIG FILE CANNOT BE ACCESSED
454	mix/progname <454> LOC: hexnumber, FETCH.VALUE hexnumber
455	mix/progname <455> FILE ERROR decnumber ON SORT-MERGE FILE

(continued)

Event	Message
456	text decnumber
457	mix/progname event-no CANNOT OPEN filename – FILETYPE NOT DATA OR SOURCE
458	AUTOMATIC SYSTEM RECOVERY INITIATED BY OPERATOR
459	AUTOMATIC SYSTEM RECOVERY INITIATED BY CLEAR/START hexnumber
460	SYSTEM DUMP CONTAINED IN FILE text
461	SYSTEM DUMP COMPLETED SUCCESSFULLY
462	SYSTEM DUMP INCOMPLETE DISK ERRORS WERE ENCOUNTERED
463	END CF
464	mix/progname DEVICE ERROR event-no – THE LP TRANSLATION TABLE text SPECIFIED IN THE SYSCONFIG FILE CANNOT BE LOCATED IN FILE SYSTRANSLATE WHEN ATTEMPTING verb ON FILE filename.
465	mix/progname DEVICE ERROR event-no – THE REQUESTED TRANSLATION TABLE IN FILE SYSTRANSLATE CANNOT BE ACCESSED WHEN ATTEMPTING verb ON FILE filename
466	mix/progname DUMP FILE NOT CREATED – A FILE NAMED pack-id DMFILmix ALREADY EXISTS
468	DEVICE ERROR event-no ON peripheral – THE REQUESTED TRANSLATION TABLE IN FILE SYSTRANSLATE CANNOT BE ACCESSED
630	"PO FINAL" IGNORED – MUST BE PRECEDED BY "PO SYSTEM"
631	input IGNORED – THE SYSTEM SPO IS IN THE PROCESS OF POWERING DOWN THE SYSTEM
632	input IGNORED – SPECIFIED DEVICE IS THE SYSTEM DISK – USE "PO SYSTEM"
633	WARNING – PROGRAMS ARE IN THE MIX ENTER "PO FINAL" TO TERMINATE THEM AND POWER OFF THE SYSTEM ENTERING ANY OTHER COMMAND WILL CAUSE THE "PO SYSTEM" TO BE IGNORED
634	ERROR ENCOUNTERED WHEN ATTEMPTING TO READ THE LABEL ON peripheral

Events 564-567 – B 90/B 900 SDI Disk Messages

These messages are produced when the SDI disk drive detects an error condition.

Message syntax:

where the hex number has the following format:

BYTE 1 (Common information)

Note bit 1 only applies to the 3/6 disk drive (B 9489-21/27).

Bit 0	Disk speed incorrect.
Bit 1	Drive servo phase offsets outwith specification.
Bit 2	Servo fault.
Bits 3-7	To be specified.

BYTE 2 and BYTE 3 (Byte 2 relates to Drive 01 and Byte 3 to Drive 02)

NOTE
For 211 (B 9493-20/40/80) disk drive only use Byte 2.

Bit 0	Drive failed to recalibrate successfully due to positional tribit accuracy.
Bit 1	Drive failed to recalibrate successfully due to circumferential accuracy.
Bit 2	Drive failed to read/write relocation table or maintenance log successfully.
Bit 3	Drive failed to recognize any sector identifiers.
Bit 4	Drive failed to detect end of sector sequence.
Bit 5	Drive failed to Read/Write confidence test.
Bits 6-7	To be specified.

Event	Message	Possible Cause	Suggested Action
564	REINSERT DISK	Drive failed to recalibrate due to circumferential accuracy.	Remove and reinsert disk.
565	TRY ANOTHER DISK	Drive failed to recalibrate due to positional tribit accuracy.	Try another disk in the drive with the reported failure. If the drive operates with no reported failure, then try the original disk.
566	DRIVE FAILED – SERVICE REQUIRED	The drive failed to read/ write confidence test.	Call local Burroughs support.

(continued)

Event	Message	Possible Cause	Suggested Action
567	DISK MAY BE CORRUPT	The drive failed to: 1. recognize any sector identifiers 2. detect end of sector sequence.	Use another disk and if error recurs, call local Burroughs support.

SUBSECTION B

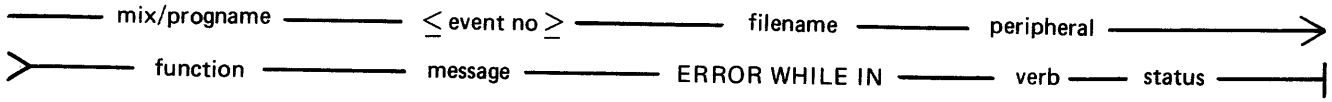
OUTPUT MESSAGES FOR B 1800/B 1900 SYSTEMS

Events 1-6

Software Information

These messages are information indicating error conditions. If any operator action is required, other message(s) follow immediately.

Message syntax:



Where:

function	INPUT or OUTPUT.
message	given in the following table.
verb	for example, READ, WRITE.
status	gives additional information such as the disk address for disk failures.

NOTE

If there is an input/output error on a disk device during a hardware search, the <sector> address given is not valid, it shows where the search started from. The error is somewhere beyond this point.

Event	Message	Possible Causes	Suggested Action
1	TAPE FORMAT	Ending label on a labelled tape is missing or invalid.	None Program has indicated it can handle the error.
2	PARITY <status>	Hard parity error found on this device (disk). Non-disk, either data could not be located, or the device has a malfunction.	None. Program has indicated it can handle the error. <status> shows disk sector start address for disk errors.
3	TIMEOUT <status>	For tape device, no data was found for quite a distance on the tape (exact length depends on unit). For disk, sector specified could not be used.	None. Program has indicated it can handle the error. <status> shows disk sector start address for disk errors.
4	ADDRESS <status>	Sector could not be found on disk (cylinder was found). Software has indicated an invalid sector or sector address on disk is corrupt.	None. Program has indicated it can handle the error itself. <status> shows disk sector start address for disk errors.
5	REWIND	Problem encountered on tape rewind.	Retry program.
6	DESCRIPTOR	Identifies I/O error that is not PARITY, TIMEOUT, ADDRESS or REWIND.	None. Program has indicated it can handle the error.

Events 7-9

Device Messages

The event number shown for these messages is only a guide to their position in the file SYS-LANGUAGE, as these messages do not produce an event number in the output.

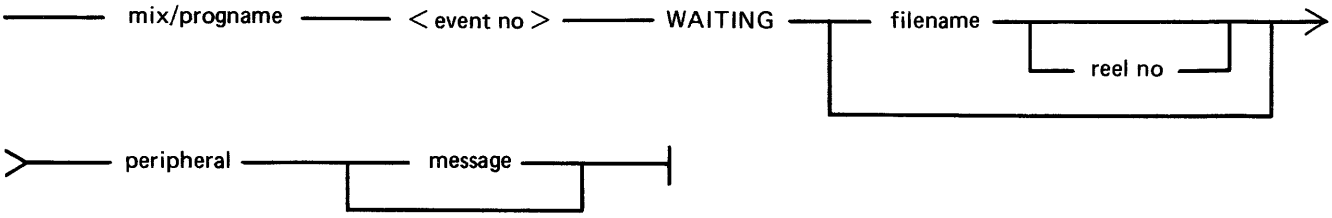
Event	Message	Possible Causes	Suggested Action
7	packid/ AVAILABLE TABLE FULL – PLEASE RUN SQ	It is possible for all entries in available table (non-file directory) to be in use. If a file then releases its disk space it may not be possible to enter it in the available table. Disk is checkerboarded.	Run SQ utility.
8	mix/progname peripheral LOCKED	A non disk device is closed with lock.	None.
9	mix/progname peripheral DEVICE NOT PURGED – NO WRITE PERMIT	An attempt has been made to close purge a write disabled magnetic tape device.	Write enable the media.

Events 10-19, 27-29, 39, 42, 69, 81-82, 85 And 88

Software Suspensions

When a running program encounters a condition that prevents it from continuing, the MCP suspends the program and informs the operator as to the reason for the suspension. When the condition is cleared, the MCP normally allows the program to continue running. If the program does not continue automatically, the operator should issue a "GO" command (see "GO" intrinsic) to the program.

Message Syntax:



Event	Message	Possible Causes	Suggested Action
10	NO FILE	Disk or tape file this program needs has not been found.	Check for correct disk or tape; supply program with backup copy of file requested.
11	DUPLICATE FILE	While attempting to place a certain file on disk or tape, program has discovered a file with the same name already exists on the device. The program halts as more than one file with the same name cannot reside on the same device.	Normally, use RM or CH to remove the existing file from disk. If in doubt, refer to program instructions. For non-disk devices, either power off, or save the unwanted drive.
12	NO DISK SPACE	There is no more available space on disk; space available is insufficient to hold file this program is attempting to write; or disk is checkerboarded.	With KA analyse amount of available space remaining. If the disk is full, remove any unnecessary files and program will continue. Or, if the disk is checkerboarded, terminate the suspended program using the DS intrinsic, then use the SQ utility to consolidate disk space. Re-execute the program that encountered the suspension.
13	DIRECTORY FULL	When the disk was initialized, the disk directory was constructed to contain a fixed number of file names. The directory has now reached its capacity.	Use RM to remove any unnecessary files and program will continue. Or, DS the suspended program. Replace disk with another disk having sufficient directory space, and re-execute the program.
14	DEVICE NOT AVAILABLE	Device that program needs in order to continue processing is either saved, not ready or in use by another program.	RY required device, or assign program to alternative device (see AD intrinsic).

(continued)

Event	Message	Possible Causes	Suggested Action
15	FORMS REQUIRED	Program is waiting for operator to insert correct forms into the output device before it will continue processing.	Insert correct forms into output device. Then use AD intrinsic to assign device to program (see AD intrinsic)
16	GENERATION NUMBER MISMATCH	While opening an old indexed file, a discrepancy has been found between the generation fields of the keyfile and the data file. The operating system will cancel the suspension when the discrepancy no longer exists.	DS the program and either run the utility KEY.CHECK (see KEY.CHECK utility), or Sort the data file to produce a new keyfile. Re-execute program.
17	NO FILE	Program needs a disk or tape that is not available to the system.	Mount the correct media and ready the device using the RY intrinsic. Or allow program currently using the device to go to end of job and then the suspended program will automatically be able to use the device.
18	DUPLICATE PACK	Two or more disks or tapes having the same names have been found on-line. Only one disk or tape of a given name may be on-line at a time.	Check for correct disk or tape. Replace or relabel one of the duplicates.
19	FILE IN USE	The specified file cannot be opened because it has already been opened by the maximum number of users allowed for the desired mode of use.	Wait until the file is not in use.
27	DEVICE NOT ON SYSTEM	There is no device of the required kind present on the system.	Fit requested device to the system, or change program so that it does not request this device.
28	DEVICE ASSIGNMENT REQUIRED	An open of an unlabelled file requires a device to be assigned.	AD the required device to the program (see AD intrinsic).
29	DUAL PACK MISMATCH	The two parts of a dual pack file have been found to be inconsistent.	Check both disks contain the correct version of the file, replacing the file if necessary.
39	BAD FILENAME	OPEN or CLOSE has detected an illegal special character imbedded in the filename.	Use SF intrinsic.
42	DISK LOCKED	The disk requested by the program is in use by another program which has "locked" access of the disk.	Wait for the disk to become available.
69	DISK FILE HEADER MISMATCH	Possible disk or disk drive problem.	Backup as many files as possible and reformat or re-initialize disk.

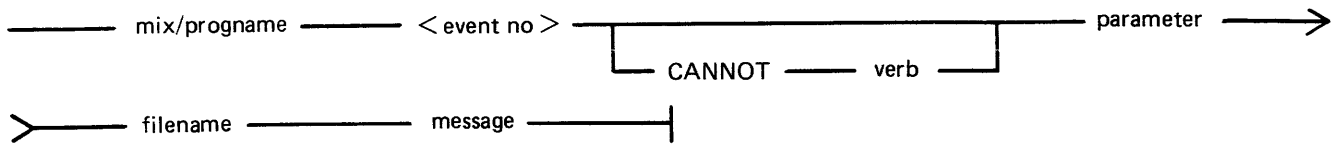
(continued)

Event	Message	Possible Causes	Suggested Action
81 or 82	FILE INCOMPLETE	The disk containing the second half of a dual pack file or the disk containing the data file of an indexed pair cannot be located. The operating system will cancel the suspension when the file is made available.	Make the required file available to the system, as follows: RY the required disk (see the RY intrinsic) or load the required disk onto the system.
85	NO FILE – DEVICE RELEASED	The tape does not have the required file.	Replace the tape with one which has the required file. Then AD the drive to the program (see AD intrinsic).
88	NO WRITE PERMIT – DEVICE RELEASED	The tape is write disabled.	Replace the tape with a write enabled tape then AD the drive to the program (see AD intrinsic).

Events 20-26, 30-38, 40, 41, 79, 80, 84, 87, 89, 90

Invalid Requests on Class A or B Communicate to MCP

Message Syntax:



These messages normally indicate program errors. The program in error should be DS'ed or DP'ed (see DS or DP intrinsics), if necessary. The operator should then attempt to run the program again. If the same error is encountered, request technical assistance.

Event	Message	Possible Causes	Suggested Action
20	no message	An illegal verb has been used in a communicate. The parameter field contains the hexadecimal value of the illegal verb.	
21	NOT FIB	The DST did not contain an index to a valid FIB. The parameter field contains the FIB segment number in decimal.	
22	FILE LOST	The medium which contained the file, has been illegally, physically removed when it was still required.	Replace the medium in the drive from which it was removed and ready the drive using the RY intrinsic.
23	ATTRIBUTE MISMATCH	The attributes specified by the program are conflicting. For example, a write to an input only file.	
24	BAD SEQUENCE	The program has attempted to perform an invalid sequence of communicates. The following are bad sequences: 1. A rewrite not immediately preceded by a successful read. 2. An overwrite immediately preceded by an open or start communicate. 3. An overwrite or rewrite preceded by a conditional read which failed. 4. An overwrite after a file test status.	
25	BAD WORK AREA	The work area segment specified within the FIB cannot be used,	

(continued)

Event	Message	Possible Causes	Suggested Action
		either because it indicates an FIB segment or because it is a read only segment; but the communicate requests a data transfer to that segment.	
26	ILLEGAL KEY	The key provided on a write communicate to an indexed file was equal to binary zero or a byte of @FF@.	
30	ALREADY OPEN	The communicate requested an open of an already open file.	
31	ALREADY CLOSED	The communicate requested a close of an already closed file.	
32	BAD ADVERB	The adverb to open was determined to be illegal for any of the following reasons: 1. Myuse equal to zero. 2. Myuse incompatible with device, for example input/output for line printer. 3. Access mode random for non-disk device. 4. Access mode not equal to sequential, random or stream. 5. Shared access mode not random or sequential.	
33	BAD BLOCK OR RECORD SIZE	The record and/or block size has been determined to be incompatible or illegal for any of the following reasons: 1. Buffer or record length equal to zero for new disk or tape files. 2. Record or buffer length exceeds physical block size. 3. Buffer length not an integer multiple of record length.	
34	BAD FILE SIZE	The maximum file size exceeds 1048576 – 16 * number of sectors in an allocation unit, or 6553 – number of sectors in one allocation unit, if a single area file.	
35	BAD NUMBER OF BUFFERS	The number of buffers exceeds 16.	

(continued)

Event	Message	Possible Causes	Suggested Action
36	BAD DEVICE	The device requested is not supported by the CMS operating systems.	
37	BAD FILE TYPE	A mismatch between the file type fields of the FPB and the DFH has been found during an open, or user attempted to access a protected file.	
38	PROTECTION ERROR	A privileged user attempted to close lock/purge a VMFILE (virtual memory) or firmware file while the file was still in use.	
40	BAD KEY SIZE	An attempt has been made to open an indexed file with key size greater than 28, or with a key size equal to zero and access not sequential input, or input/ output.	
41	BAD LEVEL NUMBER	The implementation number in a key file KFPB is greater than this implementation. (This is currently 1.)	
79	INVALID KEY FILE	The MCP has detected an inconsistency in the format of a key file during an indexed open.	
80	BAD KEY POSITION	The MCP has detected that the key for an indexed file is not wholly contained within the data record.	
84	INVALID COMMUNICATE	A valid communicate was issued to the wrong device type. For example, test status which is valid for any disk file, ICMD file and unconditional variant valid for tape file, so if issued to a card reader, this error would be produced. Error may also be issued for other reasons.	
87	SEQUENTIAL ACCESS BEYOND EOF	A sequential read has been attempted after notification of end of file.	
89	FILE NAME MISMATCH	The tape currently in use by the program has the wrong name, perhaps due to wrong reel being mounted.	

(continued)

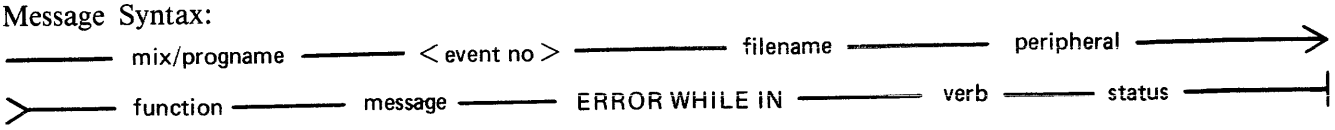
Event	Message	Possible Causes	Suggested Action
90	TAPE NOT OF REQUESTED FORMAT		

Events 43-49

Device Errors

These error messages indicate terminations due to fatal device errors. These errors are normally encountered when attempting to read disks or magnetic tapes. The program which encountered these errors should be DS'ed or DP'ed (see DS and DP intrinsics). The operator should then attempt to run the program again. If the same error is encountered the disk or tape involved should not be used until Burroughs Customer Service Engineering has been notified and the media have been checked.

Message Syntax:



Where:

function is INPUT or OUTPUT

verb specifies the communicate that was being performed when the error occurred – for example OPEN, CLOSE, WRITE.

status is the disk sector address only produced for disk devices (B 90).

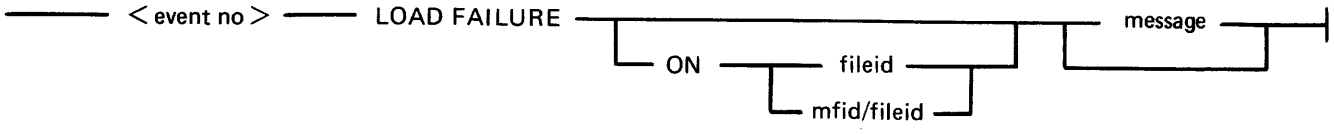
Event	Message	Possible Causes	Suggested Action
43	END OF TAPE	An unexpected end of tape has been encountered.	
44	TAPE FORMAT	The end label is missing or a labelled tape is invalid.	
45	PARITY	A hard parity error was discovered by the device during label processing.	
46	TIMEOUT	When reading a tape device no data was found for quite a considerable distance along the tape, the exact length is device dependent. For disk, the sector specified could not be found.	
47	ADDRESS	The specified cylinder was found but the required sector was not found. Either the software has indicated an invalid sector or the disk/sector address is corrupt.	
48	REWIND	A problem has been encountered when rewinding the tape.	
49	DESCRIPTOR	Identifies an input/output error other than PARITY, ADDRESS or REWIND.	

Events 50-68, 76

Loader Detected Failures

These messages indicate that the MCP failed to begin the processing of a particular program for some reason. The operator should correct the condition which caused the load failure and try running the program again.

Message Syntax:



Event	Message	Possible Causes	Suggested Action
50	DISK NOT FOUND	The specified disk is either not ready, or not on-line.	Check that correct disk has been loaded. Ready the disk with the RY intrinsic if it is not ready.
51	PROGRAM NOT FOUND	The specified program file was not on disk.	Check that the correct program and disk name were entered and re-input if necessary. Ensure disk does contain the program required.
52	FULL MIX	The program specified cannot be run at the present time for one of the following reasons: 1. No mix slots are available for a program in this priority class. 2. Another program present will not allow any other programs to enter the mix.	Wait for a suitable time to execute the program, or perform an orderly termination of some of the current programs.
53	NO DISK SPACE	There is not enough disk space for the program's Virtual Memory file (VMFILE) or the file directory of the disk has no entries for the creation of new files.	Make disk space available for program by either removing unwanted files from the disk, or copying files not required at the moment to backup, then remove them from the disk.
54	INTERPRETER NOT FOUND	The required interpreter is not on system disk with its executing name.	Ensure that the necessary interpreter (for example, BILINTERP, COBOLINT, SORTINTRINS) is on the system disk with its executing name.
55	USER COUNT ERROR	A request to exceed the maximum number of concurrent users of a file has been attempted. The maximum number of concurrent users will not exceed 32.	Wait until one or more of the programs currently using the file terminates.

(continued)

Event	Message	Possible Causes	Suggested Action
56	CODE FILE ERROR	The file type for the file specified was not a code file type. For example, the file was data or key file type.	Check input, correct it and re-enter.
57	INVALID LOAD REQUEST	The program name specified in EX command was invalid. For example it contained too many characters.	Check input, correct it and re-enter.
58	INSUFFICIENT MEMORY	There is not enough memory space to hold this program's Task Control Block (TCB) and its Program Control Block (PCB).	Wait for other programs to terminate. If this condition persists, request technical assistance.
59	MCS ALREADY PRESENT	Only one MCS can be in the mix at a time.	None.
60	DUPLICATE PACK	Two or more disks/packs packs with the same name are on-line. The system requires all disks on the system to have a unique name.	Remove unnecessary pack or re-name one of the disks using the RL intrinsic.
61	NULL MIX REQUIRED	The specified program may only run in a suitable mix. A suitable mix is one in which only the datacomm, SYS-SUPERUTL and SPIM (if running) are present. Also SYS-SUPERUTL must be dormant.	Wait for the mix to become suitable for execution of this program.
62	ILLEGAL DATA COMM LOAD REQUEST	An NDL task can only be loaded as part of the data comm loading sequence. Any other attempt to load an NDL task will fail.	Use the required NDL loading sequence.
63	DISK ERROR	There has been an irrecoverable disk error whilst attempting to load a program.	Use backup copy of the required program. If the problem continues, request technical assistance.
64	SUPERUTILITY BUSY	An attempt has been made to use one of the function of SYS-SUPERUTIL while this utility is performing another function. SYS-SUPERUTL functions are IR, LB, LF, CH, KX, PD and RM.	Wait until the current SYS-SUPERUTIL function has completed before entering the required function.
65	INSUFFICIENT REAL STORE	The program structure cannot be constructed within the amount of memory specified in the real store field in the EX intrinsic.	

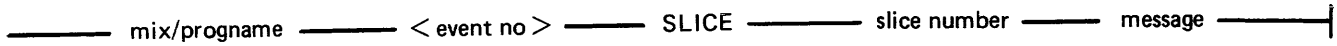
(continued)

Event	Message	Possible Causes	Suggested Action
66	DUAL ALPHABET NOT SUPPORTED	The program requires the interpreter and MCP to support dual alphabet or reverse escapement and the current system does not.	Do not use this program with this implementation of CMS.
67	FAULTY FILE EQUATE	The file equate in the EX command referenced an internal file which did not exist.	Check input and re-enter execute command (EX intrinsic).
68	DISK LOCKED	Another program has locked the required disk.	Wait until the program holding the lock on the required disk terminates then re-execute the required program.
76	INTERPRETER RELEASE LEVEL MISMATCH	The interpreter and MCP levels are incompatible.	Copy the required interpreter from backup media then re-execute the program.

Events 70-75

Run Structure Errors

Message Syntax:



These messages are output to indicate failure on execution of a program. The slice number, when present, is only meaningful when applied to the specific implementation. It does not indicate what function of the virtual machine was executing. The program encountering these errors should be DS'ed or DP'ed (see DS and DP intrinsics) if necessary. The operator should then attempt to run the program again. If the same error is encountered, request technical assistance.

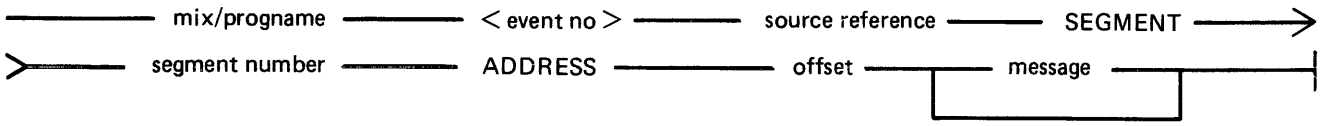
Event	Message	Possible Causes	Suggested Action
70	SEGMENT OUT OF RANGE	The segment number exceeds the number of segments declared in the program.	
71	SEGMENT SIZE ERROR	The length of the segment exceeds the declared size of the segment.	
72	STACK OVERFLOW	The amount of control stack declared in the program has been exceeded during execution.	
73	STACK UNDERFLOW	The specified program has attempted to retrieve more information from the control stack than is present.	
74	INSUFFICIENT REAL STORE	The program has attempted to exceed the memory size specified at execution (load) time in the real store field. See EX intrinsic.	
75	PROGRAM TOO LARGE	The program has attempted to exceed the physical memory size.	

Events 100-169

Interpreter Detected Errors

These messages indicate that a program interpreter has encountered an error on execution of the program. The program in error should be DP'ed (see DP intrinsic) if necessary. The source program, object program, dumpfile produced by DP and any data files used by the program should be saved. The operator should then try to run the program again. If the same error is encountered, request technical assistance and supply all relevant data saved to the technician.

Message Formats:



NOTE

Source reference information is only provided if the appropriate compiler control instruction is included during the compile:

COBOL: Source line number
RPG: Debug line number
MPL: Procedure number

Event	Message	Possible Causes	Suggested Action
100	COMMUNICATE ERROR	MCP returned @80@ in Byte 0 of Fetch Message on a communicate.	
101	COMMUNICATE EOF ERROR	MCP returned an End of File indication in Fetch message (@20 10 00@) and the user has not specified any action if EOF occurs.	
102	COMMUNICATE I/O ERROR	MCP returned an I/O error other than EOF (@20 10 00@) in Fetch message and user has not specified any action to be taken if error occurs.	
103	SEGMENT NUMBER ERROR	Interpreter detected an invalid data segment number.	
104	WRITE ERROR	Interpreter detected an attempted WRITE to a Read-Only Segment or, if COBOL, a literal.	
105	SEGMENT BOUNDARY VIOLATION	Interpreter, on trying to resolve an address, has discovered that the address of the data or code is out of range.	

Events 106-109 reserved for further use.

Event	Message	Possible Cause	Suggested Action
110	INVALID OP	Code file has become corrupt or, an error exists in MPLII compiler or interpreter, or an old level of interpreter with same feature not implemented.	
Events 111-114 reserved for future use.			
115	DESCRIPTOR ACCESS	Program tried store the fetch value to a non-character field; to convert to a non-character field; or to store to a self-relative descriptor. Or, an error occurred in the SETNAME procedure, or in the parameter in decimal arithmetic. Other possible errors are: key not character (8) in authenticate, message not character in authenticate, or monitor parameter out of range.	
116	SEGMENT SIZE ERROR	The length assigned at run time to a segment or segmap page must be less than equal to the maximum length specified, and the segment must not have been used by the program before the SEGMENT/SEGMAP statement has been encountered at run-time time.	
117	ADDRESS ERROR	SETNAME extent error; identifier has become out of scope.	
118	MESSAGE REFERENCE ERROR	Size of message reference field not divisible by 4, or illegal access to message reference field.	
119	STRING STORAGE ERROR	Illegal destination in store string instruction.	
120	REMAP ERROR	Program tried to re-map a bit descriptor.	
121	SUBSTRING ERROR	Attempt made to sub-string into a non-character area, or length of substring greater than 255, or substring not wholly contained in area being substringed.	

(continued)

Event	Message	Possible Cause	Suggested Action
122	INDEX ERROR	Program tried to index a self-relative descriptor, or to bit index a self-relative descriptor.	
123	EXIT ERROR	Local data returned from a function.	
124	CPA ERROR	Error in communicate parameter area: for example, message reference expected, or character field of 3 bytes expected.	
125	DIVIDE ERROR	Divide by zero attempted.	
126	ZIP ERROR	Error returned after ZIP.	
127	BIT DESCRIPTION ERROR	Bit field overlaps more than one byte boundary.	
128	FPB ERROR	Error in file parameter block encountered, or subfield of a non-F.P.B. segment requested.	
129	CONTROL STACK ERROR	Control Stack overflow in communicate instruction, or control stack underflow in exit instruction.	
130	DATA STACK ERROR	Illegal descriptor encountered. Usually caused by overwriting descriptors on the data stack by data in an assignment statement, where the left hand side has a bad offset value. For example A(X):=VAL; where X is too big – this will write beyond the end of A and may corrupt descriptors.	
131	DECLARATION MODE ERROR	Size of character field greater than 255.	
132	DATA STRUCTURE ERROR	Insufficient room for structure nesting or size of character field greater than 255, or insufficient room for array bound evaluation, or non-character descriptor encountered when character descriptor expected.	

Events 133-139 reserved for future use.

Event	Message	Possible Cause	Suggested Action
140	INVALID S-OP CODE	Invalid S-Op-Code. Code file is corrupt or error exists in COBOL/RPG compiler or interpreter.	
141	INVALID COPX – GREATER THAN SIZE OF COP TABLE	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
142	ALPHANUMERIC FIELD TYPE NOT 8-BIT UNSIGNED	Corrupt code file or error exists in COBOL/RPG RPG compiler or interpreter.	
143	INVALID EDIT MICRO OPERATOR	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
144	INLINE EDIT MASK NOT CORRECTLY TERMINATED	Corrupt code file or error exists in COBOL/RPG RPG compiler or interpreter.	
145	EXAMINE SOURCE FIELD ERROR	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
146	EXAMINE PARAMETER FIELD NOT 8-BIT UNSIGNED, ONE CHARACTER	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
147	EXAMINE CONTROL BYTE ERROR	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
	148 COMPARE FOR CLASS – CLASS AND FIELD TYPE INCOMPATIBLE	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
149	SUBSCRIPTED OR INDEXED SUBSCRIPTED OR INDEX	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	

(continued)

Event	Message	Possible Cause	Suggested Action
150	INDEXED/ SUBSCRIPT VARIABLE IS INDEXED/ SUBSCRIPTED BY MORE THAN 3 VARIABLES	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
151	FETCH COMMUNICATE RESPONSE FIELD NOT OF LENGTH 3 BYTES	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
152	INVALID EXAMINE SPECIFICATION	Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
Events 153-157 reserved for future use.			
158	PERFORM STACK UNDERFLOW	Attempt to exit from a module when not in a called module. Corrupt code file or error exists in COBOL/RPG compiler or interpreter.	
159	CANNOT OPEN NEW OUTPUT FILE WHEN VALUE OF FILESIZE IS ZERO	Program attempted to create a file on disk with a filesize of zero. Invalid code as no records could ever be output to the file created.	Perform suitable alterations to the program which allow the program to create a file with non-zero filesize.
160	PERFORM STACK OVERFLOW	Indicates that the number of nested PERFORM statements exceeds that allowed within the Perform Stack at compile time (if not specified then the default value is used). If this did not result from programming error, the Perform Stack should be increased.	The size of the Perform Stack may be increased by MODIFY utility, using CONTROL.STACK option to change the PPB (see section 4).
161	NON POSITIVE SUBSCRIPT	Subscripts must be a value greater than zero.	
162	ARRAY BOUND VIOLATION	Subscript outside the upper bound of OCCURS clause.	
Event 163 reserved for future use.			

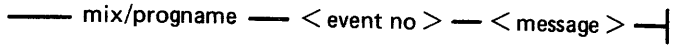
(continued)

Event	Message	Possible Cause	Suggested Action
164	TRANSLATION SOURCE ERROR	In an MPL TRANSLATE statement the translation text or translation table not of type character (1).	
165	NO STOP RUN ENCOUNTERED	End of code file has been encountered but no STOP RUN statement has been found.	
166	INVALID SIGN CODE		
167	I/O ERROR	Invalid READ/WRITE to a file encountered.	
168	SORT OR MERGE ERROR	An error has been encountered in SORT or MERGE.	
169	ZIP FAILURE	Zip of another task has failed. For example zipped program not present, zipped file is not a code-file.	

Events 170-199

Sort/Merge Information

Message Format:



The following messages may be output while SORT or SORTINTRINS are running. They will display the name of the program which executed or attempted to execute SORTINTRINS, the MCP event number and a message to explain why the event was displayed.

Event	Message	Possible Causes	Suggested Action
170	DUPLICATE RECORD record number	Only for keyfile creation. Another record in the file has the same key as this record.	If duplicates are desired, specify "DUPLICATES" on input.
171	ILLEGAL INDEX KEY IN RECORD key	Only for keyfile creation. The key field contains either all binary zeros or has one or more bytes with @FF@. This record will not be referenced from keyfile.	None.
172	RECORDS LOST/ GAINED BY SORT-MERGE	Probably indicates an error in SORTINTRINS.	See introductory paragraph concerning errors.
173	number DUPLICATE RECORDS	Normal message. Tells the operator the total number of records that have duplicates. (See Event 170).	None.
174	number RECORDS CONTAINING INVALID INDEX KEYS	Normal message. Tells operator the total number of records with invalid index keys. (See Event 171).	None.
175	number DELETED RECORDS	Informs the operator that a number of records were deleted as they contained only hex FF in every byte.	None.
176	number RECORDS MERGED	Normal message for MERGE only. Tells the operator the total number of records merged from all files.	None.
177	number FILES MERGED	Normal message for MERGE only. Tells the total number of files merged. Should be the same as the number requested in the Sort Spec.	None.

(continued)

Event	Message	Possible Causes	Suggested Action
178	SORT-MERGE OUTPUT FILE NOT CREATED	SORTINTRINS was DS'ed. May indicate corrupt SORT or SORTINTRINS programs.	See introductory paragraph concerning SORT errors.
179	SORT-MERGE ABNORMAL EOJ	Early termination due to errors.	
180	SORT-MERGE SOFTWARE ERROR <SEG number OFFSET number>	Error in SORTINTRINS.	See introductory paragraph concerning SORT errors.
181	number RECORDS REFERENCED BY KEYFILE/ TAGFILE	Only for keyfile/tagfile creation. Tells the number of records referenced by a keyfile/tagfile.	None.
182	NO INITIATING MESSAGE	SORT intrinsic requires a properly coded initiating message. This should be properly formatted by SORT or Sorts within programming languages such as COBOL. Probably indicates an attempt to execute SORTINTRINS directly.	None.
183	number RECORDS SORTED	Normal message. Tells operator the number of records sorted by a successful data file SORT.	None.
184	FILE ERROR <number> NEAR RECORD number ON filename	The message informs the operator that an error has been encountered within the specified file near the specified record number. The <number> means: 1. EOF on output file. 2. Parity error on input file. 3. EOF on Sort Workfile. 4. Bad disk address. 5. SORT workfile error. 6. Input file error. 7. Output file error. Except for 2, this probably indicates an error in SORTINTRINS.	See introductory paragraph concerning SORT errors.
185	UNORDERED MERGE INPUT FILE	Files to be merged did not have an increasing/decreasing key value. Either the file is incorrect	SORT files for correct order, then retry.

(continued)

Event	Message	Possible Causes	Suggested Action
	filename NEAR RECORD number	or the key position has been incorrectly specified.	
186	TOO MANY RECORDS FOR SORT-MERGE	Machine dependent limitation.	See introductory paragraph concerning SORT errors.
187	DUPLICATE RECORDS – KEYFILE NOT BUILT	DUPLICATES was not specified on input and they were found to exist. This message age will be displayed and no keyfile will be built.	Specify DUPLICATES on input.
188	INITIATING MESSAGE NOT VALID	Initiating message supplied to SORTINTRINS is not in proper format. This could be caused by a fault in the program that zipped the SORTINTRINS.	See introductory paragraph concerning SORT errors.
189	SORT-MERGE version INITIATED FROM mix number/ program name	Normal message to inform the operator which version of the SORTINTRINS is in use and by which program.	None.
193	INPUT RECORD SIZES UNEQUAL – BAD FILE filename	For a merge sort, the input files do not have the same record size. The offending file is named.	None.
194	IN/OUT RECORD SIZES BAD – OUTPUT SIZE CHANGED	The output file been specified with a record size different to that of the input file. This has been corrected automatically.	
195	BAD RECORD/ BLOCK SIZE FOR OUTPUT DEVICE	The output file buffer size must be less than 32768 for disk file.	
196	KEY OVER-RUNS RECORD END	Key limit is beyond the end of record.	
197	CANNOT SPLIT INDEX FILE	An index file cannot reside on more than one disk.	

(continued)

Event	Message	Possible Causes	Suggested Action
198	number PARITY BLOCKS	Indicates the number of blocks in which parity errors were found.	
199	INDEX SORT INPUT FILE NOT TYPE DATA	Input files must be of type data.	

Events 200-349

Data Communications Errors On B 1000 Systems

These messages indicate invalid conditions encountered by the data communications sub-system. For full descriptions of these events see the CMS Data Communications Subsystem Reference Manual, form 1090909.

Event Number	Breakdown
200-254	Implementation Independent Errors
272-274	Implementation Independent Errors
282-284	Implementation Independent Errors
285-299	B 1800 Dependent Errors

Events 200-247

Message Format:

Event	Message	Possible Causes	Suggested Action
200	DC ERROR BAD MSG TYPE	The type field in the message header contains a value greater than 12. Error is returned after QUEUE.	
201	DC ERROR BAD LSN	A reference has been made to a Logical Station Number (LSN) greater than STATION.COUNT-1. Error is returned after: ALLOW.OUTPUT CLEAR CONTINUE.STATION DISALLOW.OUTPUT QUEUE REDEFINE.STATION ROUTE.OUTPUT SET.INPUT.LIMIT STATION.DESCRIPTION STATION.STATUS	
202	DC ERROR BAD QUEUE REF	A reference has been made to an invalid queue. Certain communicates are restricted to certain queues, therefore the queue referenced may exist while being invalid in this context. Error is returned after: CLEAR DEQUEUE QUEUE QUEUE.DEPTH RECALL ROUTE.INPUT ROUTE.OUTPUT SET.QUEUE.LIMIT	
203	DC ERROR BAD SUBNET NO	A reference has been made to a subnet greater than SUBNET.COUNT-1. Error is returned after: ALLOW.INPUT DISALLOW.INPUT SUBNET.DESCRIPTION SUBNET.STATIONS	
204	DC ERROR TEXT SIZE TOO BIG	The text length in the message header was set greater than the message length field. Error is returned after: WRITE.HEADER	

(continued)

Event	Message	Possible Causes	Suggested Action
205	DC ERROR NULL MREF	An attempt has been made to perform a function on a null message reference, that is, one which does not reference a message space. Error is returned after: COPY.TEXT QUEUE READ.HEADER READ.TEXT WRITE.HEADER WRITE.TEXT	
206	DC ERROR BYTE INDEX TOO BIG	The starting byte index of a text transfer is illegal. For the source message the index must be less than text length. For the destination message, the index must be less than message length. Error is returned after: COPY.TEXT READ.TEXT WRITE.TEXT	
207	DC ERROR BAD TASK NO	The task number referenced is not currently in the mix or is outside the range of the mix table. Error is returned after: ALLOW.INPUT ALLOW.OUTPUT CONTINUE.TASK DISALLOW.INPUT DISALLOW.OUTPUT SET.OUTPUT.LIMIT TASK.NAME TASK.STATUS	
208	DC ERROR BAD LINE NO	The Logical Line Number (LLN) reference is greater than LINE.COUNT-1. Error is returned after: QUEUE LINE.DESCRIPTION LINE.STATIONS LINE.STATUS REDEFINE.LINE REDEFINE.STATION	
209	DC ERROR BAD MODEM NO	A reference has been made to a Logical Modem Number (LMN) greater than MODEM.COUNT-1. Error is returned after:	

(continued)

Event	Message	Possible Causes	Suggested Action
		MODEM.DESCRIPTION REDEFINE.LINE REDEFINE.STATION	
210	DC ERROR BAD TERMINAL NO	A reference has been made to a Logical Terminal Number (LTN) greater than TERMINAL.COUNT-1. Error is returned after: TERMINAL.DESCRIPTION REDEFINE.STATION	
211	DC ERROR NOSPACE	No message space is available to execute the communicate. Error is returned after: CLEAR RECALL	
212	DC ERROR STATION NOT ATTACHED	An attempt has been made to make an unattached station ready. Error is returned after: QUEUE	
213	DC ERROR COMM NOT IMPLEMENTED	The last communicate issued is not implemented on this CMS system.	
214	DC ERROR LIMIT NOT ALLOWED	A queue limit of zero has been specified. Error is returned after: SET.INPUT.LIMIT SET.OUTPUT.LIMIT SET.QUEUE.LIMIT	
218	DC ERROR SPO ATTRIBUTE MISMATCH	A REDEFINE.STATION COMMUNICATE has been issued which would result in a station that is either 1. SPO capable and type bits or 2. SPO capable but not MYUSE INPUT and OUTPUT.	
219	DC ERROR IMPROPER SPO STATION CONDITION	A REDEFINE.STATION COMMUNICATE has been issued while a SPO capable station is in SCL mode, or has SCL output messages queued.	
220	DC ERROR STATION ALREADY ATTACHED	The LLN of an attached station has been set to a value other than @FF@. Error is returned after: REDEFINE.STATION	
221	DC ERROR ATTRIBUTE MISMATCH	The new attributes of the station or line are inconsistent with the existing network definition. Error	

(continued)

Event	Message	Possible Causes	Suggested Action
		is returned after: REDEFINE.STATION REDEFINE.LINE	
222	DC ERROR DIRECT CONNECT LINE	An attempt has been made to assign a MODEM to a direct connect line. Error is returned after: REDEFINE.LINE	
223	DC ERROR FULL DUPLEX MISMATCH	Attribute mismatch of full duplex terminal. Error is returned after: REDEFINE.LINE REDEFINE.STATION	
224	DC ERROR INCOMPLETE VARIABLE	The length of the parameter area to be used for reconfiguration is insufficient. Error is returned after: REDEFINE.LINE REDEFINE.STATION	
225	DC ERROR IMPROPER LINE CONDITION	The line being redefined is not in the required state of NOT READY and, for a switched line, not switched busy or not connected. Error is returned after: REDEFINE.LINE	
226	DC ERROR MESSAGES QUEUED	Messages are queued for output to the station referenced by REDEFINE.STATION. Messages are queued for output to a station on the line referenced by REDEFINE.LINE. Error is returned after: REDEFINE.STATION REDEFINE.LINE	
227	DC ERROR NO VACANCY ON LINE	The MAXSTATIONS statement in the line section of the NDL defines the maximum number of stations which may be attached to a particular line. An attempt has been made to attach a station to a line which already has MAXSTATIONS.	
228	DC ERROR SPEED MISMATCH	The speed specified for a station when either redefining the station or attaching the station to a line, does not match the speeds of the other stations on that line.	

(continued)

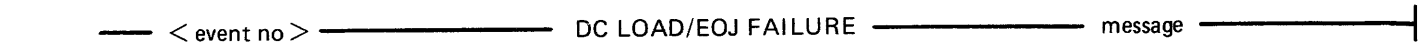
Event	Message	Possible Causes	Suggested Action
229	DC ERROR QUEUE FULL	The MCS has attempted to queue a message which would cause the queue count field of the station table or subnet table to overflow.	
230	DC ERROR NDL DCP MISMATCH	The data comm loader has detected an inconsistency between the NDL code file and the DCP code file. Possibly the DCP code file was not generated from this NDL code file.	Re-NPC the NDL code file to produce a consistent DCP code file.
231	DC ERROR ACCESS DENIED	The MCS tried to access a station or subnet which it does not own.	
232	DC ERROR DENY MULTIPLE OWNERSHIP	The MCS tried to attach a station to a switched line on which stations owned by a different MCS are already attached.	
233	DC ERROR RECON PENDING	Reconfiguration in progress.	
234	DC ERROR INVALID RELINQUISH	An attempt by an MCS to release control of a station cannot be accomplished for one of the following reasons: 1. The station or subnet to be relinquished is queued. 2. The station is READY. 3. The station is attached to a switched line. 4. The mix number passed is not that of an MCS.	
235	DC ERROR MCS NOT RUNNING	An MCS has tried to relinquish a station/subnet to an MCS which is not currently in the mix.	
236	DC ERROR BAD ENVIRONMENT	Invalid routing of message in multi MCS environment.	
245	DC ERROR SPO CAPABILITY OVERRIDDEN FOR STATIONS WITH	The station is defined as both SPO = TRUE and is type BITS, or both SPO = TRUE and is not MYUSE INPUT and OUTPUT.	

(continued)

Event	Message	Possible Causes	Suggested Action
	INCOMPATIBLE ATTRIBUTES		
246	DC ERROR INVALID MCS NAME	An attempt was made to load an MCS that has not been identified in the NDL "MCSLIST" statement.	
247	DC ERROR MAX THIS MCS	An attempt was made to load an MCS which has the same name as an MCS already executing.	

Events 249-254

Message Format



Event	Message	Possible Causes	Suggested Action
249	DC LOAD/ EOJ FAILURE BAD NDL PRIORITY CLASS	The NDLSYS file does not have the correct value in the priority class field of the PPB (@3180@).	
250	DC LOAD/ EOJ FAILURE DISK ERROR	The NDLSYS or DCP file cannot be read because of a disk I/O failure.	
251	DC LOAD/ EOJ FAILURE NDL DATA ERROR	The NDLSYS file either has a line with an invalid address for B 800 or specifies an amount of required memory which is insufficient for the tables and buffers declared.	
252	DC LOAD/ EOJ FAILURE INSUFFICIENT MEMORY	The memory space required field of the preset data in the NDLSYS file specifies more space than the MCP can provide.	
253	DC LOAD/ EOJ FAILURE CANNOT CLOSE NDL FILE	Performing the close communicate on the NDLSYS has failed.	
254	DC LOAD/ EOJ FAILURE CANNOT OPEN NDL FILE	Performing the open communicate on the NDLSYS has failed. For example: 1. File is not on disk. 2. File has wrong file type.	

Events 272-274

Message Format:

Event	Message	Possible Causes	Suggested Action
272	DC ERROR PROCESSOR NUMBER INVALID	A Load/Reload request specifies an invalid DCP.	
273	DC ERROR PROCESSOR BUSY	A reload specifies a DCP which is busy.	
274	DC ERROR PROGRAM FILE NAME INVALID	The DCP file name specified in the reload is not defined in the NDLSYS file.	

Events 282-284

Message format

Event	Message	Possible Causes	Suggested Action
282	DC LOAD/ EOJ FAILURE CANNOT CLOSE DCP FILE	Performing a close communicate on the DCP file has failed.	
283	DC LOAD/ EOJ FAILURE CANNOT OPEN DCP FILE	Performing the open communicate on the the DCP has failed for one of the following reasons: 1. The file is not on disk. 2. The file has a bad file type. 3. The file is larger than DCP memory.	
284	DC LOAD/ EOJ FAILURE DC number NOT ON SYSTEM	The specified DCP has not been warmstarted.	

Events 285-286

Event	Message	Possible Causes	Suggested Action
285	ERROR number ON DC number	An error has occurred on on the DC processor indicated.	
286	DC LOAD FAILURE – SPO LINE REFERENCED IN NDL	NDL specifies a physical line address which is the SPO line – this is not permitted.	

Events 350-499

Implementation Dependent System Messages

These messages may or may not be output, depending on the specific CMS system being used.

Unless otherwise shown in the message, these messages do not output an event number. The event number shown in the Event column is the message's relative position inside the MCP section of SYSLANGUAGE.

Events 350-369

These messages may be output from any CMS operating system.

Message format:

Event	Message	Possible Causes	Suggested Action
350	BAD DISK – CANNOT BE LOADED	Label and directory information could not be read at disk load time.	Try re-inserting disk.
351	SHOULD BE REINITIALIZED SOON	An excessive number of retries are occurring on this disk.	Copy all required files from this disk and reinitialize the disk.
352	BAD DISK – RELOCATION TABLE FULL	The hardware has automatically removed as many bad sectors as it has the capacity for.	Replace the disk if it is a floppy disk, otherwise contact local Burroughs support.
353	EXPIRING – BACK-UP	The hardware will shortly have no more room left in which to store bad sectors.	Copy all required files from this disk. If this is a floppy disk replace it, otherwise contact local Burroughs support.

Events 354-369 Reserved for future use.

Events 470-499

Event	Message
470	CANNOT READ LABEL ON peripheral
471	AVR FAILURE ON peripheral
472	peripheral NOT CMS DISK
473	peripheral NOT STANDARD LABEL
474	peripheral WRONG DENSITY
475	peripheral NO WRITE PERMIT
476	LT REQUIRED FOR peripheral
477	peripheral UNIT CHECK

(continued)

Event	Message
478	mix/progname <478> filename peripheral RESULT DESCRIPTOR ERROR WHILE IN verb RD = hex-number hex-number DISK ADDRESS = hex-number
479	mix/progname <479> filename peripheral FILE BOUNDARY VIOLATION ERROR WHILE IN verb DISK ADDRESS = hex-number
480	SYSTEM DISK PO'ED
481	LOAD FAILURE ON filename INTERPRETER TABLE OVERFLOW
482	ENTER TIME AS HHMM

Events 483 to 499 reserved for future use.

SECTION 8
B 90 DEPENDENT SYSTEM SOFTWARE

INTRODUCTION

This section covers those items in the CMS software which are operationally different on the B 90 series from other CMS implementations. These differences are mainly a result of the different hardware features involved. The software covered includes:

Powering the B 90 on and off.
The B 90 CMS Warmstart and Coldstart "bootstrap" feature.
Stand-Alone Utilities (SAU), operating without MCP control.
COLDSTART utility, operating without MCP control.
Loading the MCP.
Particular features of the B 90 MCP.
Taking memory (system) dumps.
B 90 system errors and symptoms.
Utilities released only for B 90.
Intrinsics released only for B 90.

POWER ON

Ensure no disks, tapes or cassettes are in the system (failure to do this may result in subsequent media corruption). Ensure the fixed disk is powered off.

Turn the system power on. It is then under the control of ROM which performs a mini-test of critical machine components to verify it is capable of starting. The successful completion of this test is verified by the PK lights lighting and then turning off sequentially, with PK1 and PK2 remaining lit.

For ROM levels less than 5, PK1 permits the loading of a cassette into the system. Some examples of cassettes to load would be:

1. AE 500 firmware to cause the B 90 to perform as an AE 500, or
2. ACSYS SL5 emulator firmware cassettes to cause the B 90 to process Series L cassette programs on disk.

B 90 cassette loads are not used in CMS.

For ROM level 5, PK1 permits loading of the utility COLDSTART from coldstart medium (see COLDSTART utility for a full description).

PK2 permits the loading of information from a disk into the system. Some examples of disks to load would be:

1. ACSYS (SL5 emulator) firmware disks, or
2. CMS disks to cause the B 90 to load CMS firmware.

NOTE

ACSYS (SL5 emulator) can only be run on a console system.

Load the CMS disk in any available disk drive.

A 1 MB BSMD mini-disk is considered loaded immediately the drive unit door is closed, and the blue indicator is lit (disk properly inserted and up to speed). For MCP control, the disk must be write-enabled (red indicator lit).

For a 3 MB BSMII mini-disk, wait until the drive has calibrated, and the white number indicator is lit. For MCP control, the disk must be write-enabled (red indicator lit).

For cartridge disk, wait about 20 seconds for the cartridge to come up to speed (a click is heard as the heads access the disk). Ideally, they should be run initially for a few minutes before use to achieve the correct running temperature. For MCP control, the disk must be write-enabled.

For fixed disk, wait until the 'READY' half of the "BUSY/READY" button is on.

Depress PK2 to enter CMS Bootstrap Mode (see below). The various states, including Initial State and Bootstrap Mode, are shown in figures 8-1 and 8-2.

Possible errors in the power-on sequence are given later in this section under System Load Errors.

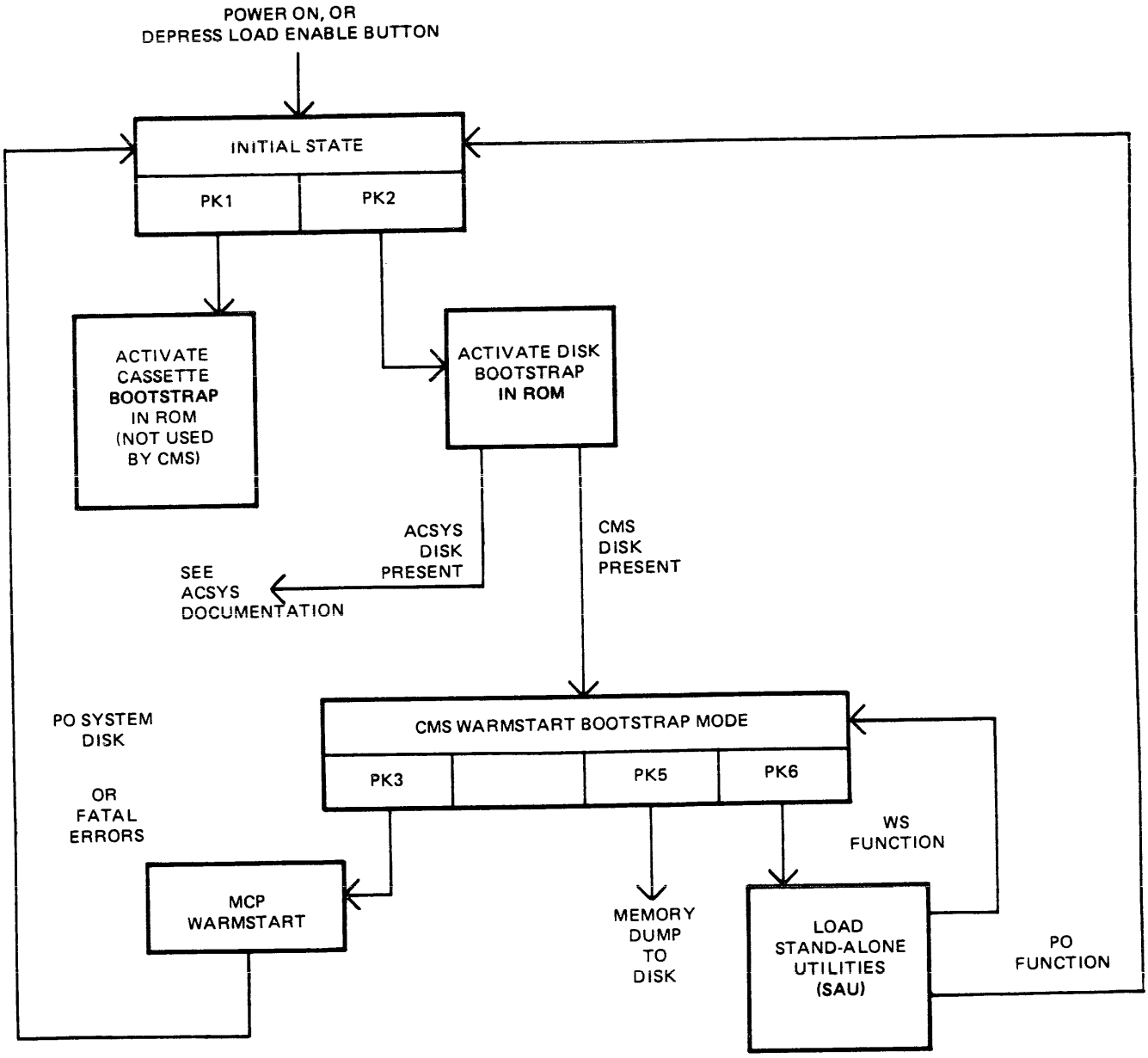


Figure 8-1. B 90 Warmstart ROM Level 4

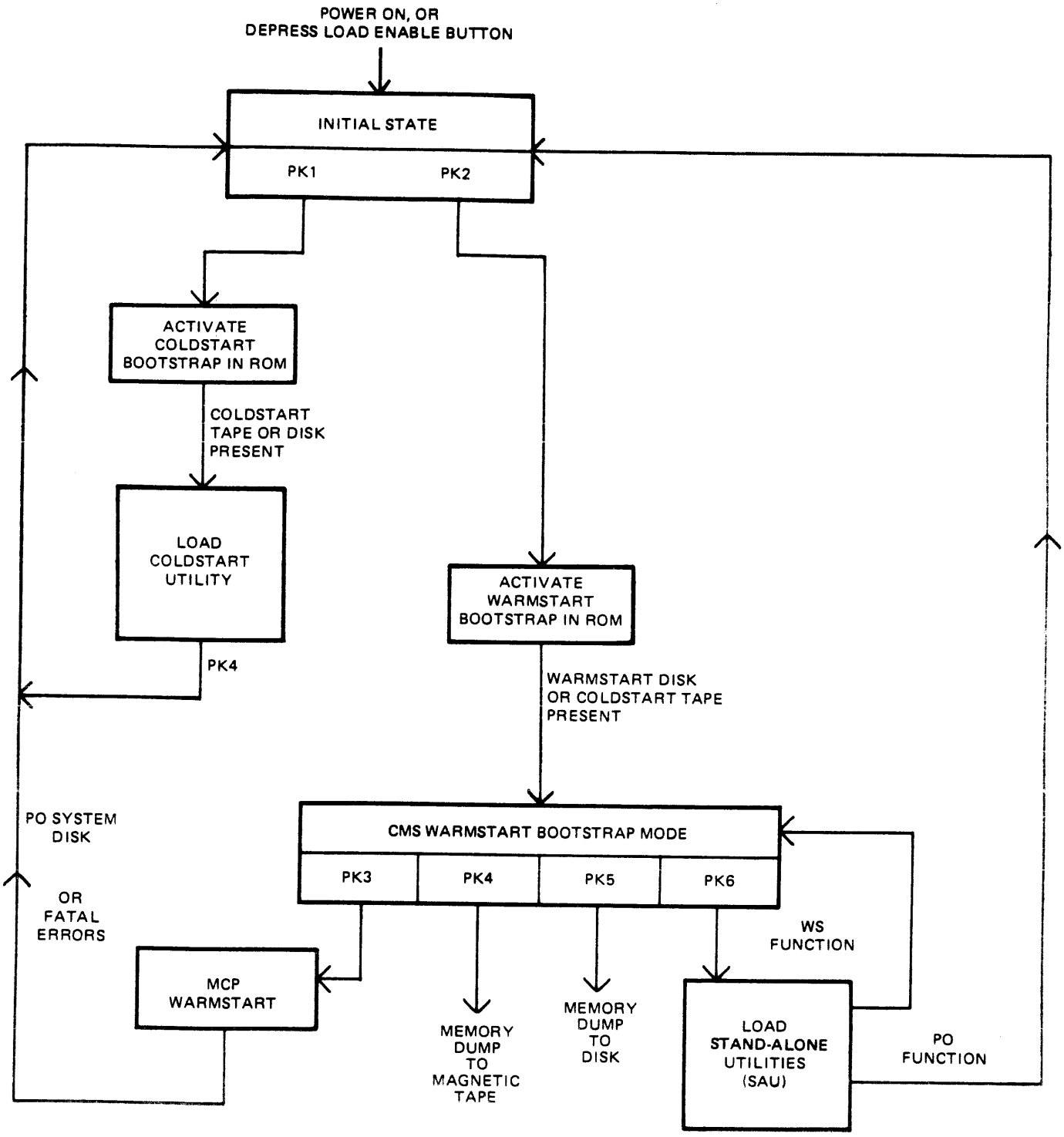


Figure 8-2. B 90 Coldstart and Warmstart ROM Level 5

CMS BOOTSTRAP MODE

From the Initial State, depression of PK2 initiates a ROM load routine which searches through all present disks until it finds the bootstrap code. Refer to ROM Scanning Algorithm for details of the search algorithm.

Successful load of CMS bootstrap: PK3 through PK6 will be lit, to provide the following facilities:

PK3 – warmstart MCPX.
PK4 – dump contents of memory to magnetic tape (ROM level 5 only).
PK5 – dump contents of memory to disk.
PK6 – enter Stand-Alone Utilities (SAU).

Forcing System Initialization

When the system hangs (that is, it is not performing any functions or responding to any input from an operator, but has not returned to the initial state), it is necessary to force the system to initialize.

This is done by pressing the Load Enable button in the main cabinet. Never switch the cabinet off, or unload disks or cassettes in use, as this can cause media corruption of various kinds.

If the correct procedure is followed, then although disk or cassette files may be only partially created or updated, the system when recovered should still be able to access the media.

COLDSTART (B 90 ONLY)

The principal function of this utility is to load files from Coldstart media to a 211 fixed disk in CMS format on a B 90 consoleless system. As the 211 has to be in CMS format, the COLDSTART utility provides an option to initialize the highest 211 drive on the highest channel of the B 90 consoleless system.

The Coldstart medium may be one of the following:

1. 1 MB disk with COLDSTART bootstrap.
2. BSMII (3/6 disk) with COLDSTART bootstrap.
3. Magnetic tape in LD format, with the Tape name "CLDSTRT".

All the above must contain COLDSTART utility and CMSBOOT. Additional files are contained in the above as required.

Restrictions on Use

1. The system ROMs must support the COLDSTART utility. A minimum of level 5 ROM is required to support the Customer utility for use on a B 90.

2. Coldstart tapes can be produced on B 90s, B 900s or B 1900s. However, on a B 900, files should not be dumped from an unrestricted pseudo-pack, as the COLDSTART utility will not be able to load such tapes.

3. Dual pack files cannot be handled by COLDSTART. If the COLDSTART encounters a dual pack file, it halts and informs the operator via a non-fatal error message. To continue loading the rest of the files, the operator is required to press the RESET key.

4. Multi-reel tape cannot be handled by COLDSTART. If a multi-reel tape is loaded, the utility loads the files from the first reel until it reaches the last file. It will not load the last file, since part of the file will be on the next reel. This results in a fatal error. The COLDSTART utility can handle an unlimited number of single-reel tapes; that is, tapes which do not have a file which is continued on another tape, during one execution of the utility.

Creation of a Coldstart Tape

The coldstart magnetic tape must be in LD utility format and called "CLDSTRT".

For reasons of efficiency, it is recommended that CMSBOOT be the first and COLDSTART the second file on the tape. However, the ROM, BOOTSTRAP and COLDSTART utility scans along the tape for these files.

When dumping files to tape using the CMS utility LD, all the files to be dumped must reside on the same disk. This may or may not be the system disk.

Ensure that a purged magnetic tape is on line.

Use LD with the initiating message below to dump the files.

```
LD DUMP PRINT.DIR FROM <disk-name> TO CLDSTRT CMSBOOT, COLDSTART,
<file-list>
```

Specification of PRINT.DIR in the initiating message causes an accurate tape directory print to be produced at the end of the dump. This directory, whether produced by LD or TAPELR, contains useful information including file names and file numbers. Since the COLDSTART utility communicates with the operator through the MTR keyboard, it is only able to display file numbers, not file names, when encountering an error.

If the files are being dumped from the system disk, the FROM <disk-name> part can be omitted from the initiating message.

The above initiating message causes the files CMSBOOT and COLDSTART to appear first and second, followed by the other files specified, on the tape.

If the files contained in <file-list> contain CMSBOOT and/or COLDSTART, a warning message of the form:

DUPLICATE – <file-name> ALREADY BEING DUMPED

will be displayed for each of CMSBOOT and/or COLDSTART. These messages can be ignored.

NOTE

Only single reel tapes can be handled by the COLDSTART utility. Also log files will not be dumped to a coldstart tape.

For further details, refer to the LD utility.

Creation of a Coldstart Disk

A coldstart disk is made in one of the following ways:

1. Use the CI function in the B 90 SAU which initializes the disk in the normal manner for that type of disk. CI then writes the coldstart bootstrap, from CMSBOOT, into track zero of the disk.

2. Use the RF or IN option in DSKUTL with "<COLDSTART>", on a B 90, specified in the initiating message. The RF or IN option performs as before, except that the coldstart bootstrap, from CMSBOOT, is written to track zero of the disk.

3. The methods described above cause the loss of any files present on the disk. By using the BOOT option in DSKUTL, with "<COLDSTART>" in the initiating message on a CMS standard disk, the bootstrap in track zero of the disk is replaced by the coldstart bootstrap without the loss of any files on the disk.

NOTE

The file CMSBOOT must be version 3.04 or greater. Also, all the above methods must be carried out on a B 90 system.

For further information and syntax, refer to the relevant utility.

The COLDSTART utility and CMSBOOT are now copied to the disk, which has a coldstart bootstrap, along with the required files.

Outline of COLDSTART Utility

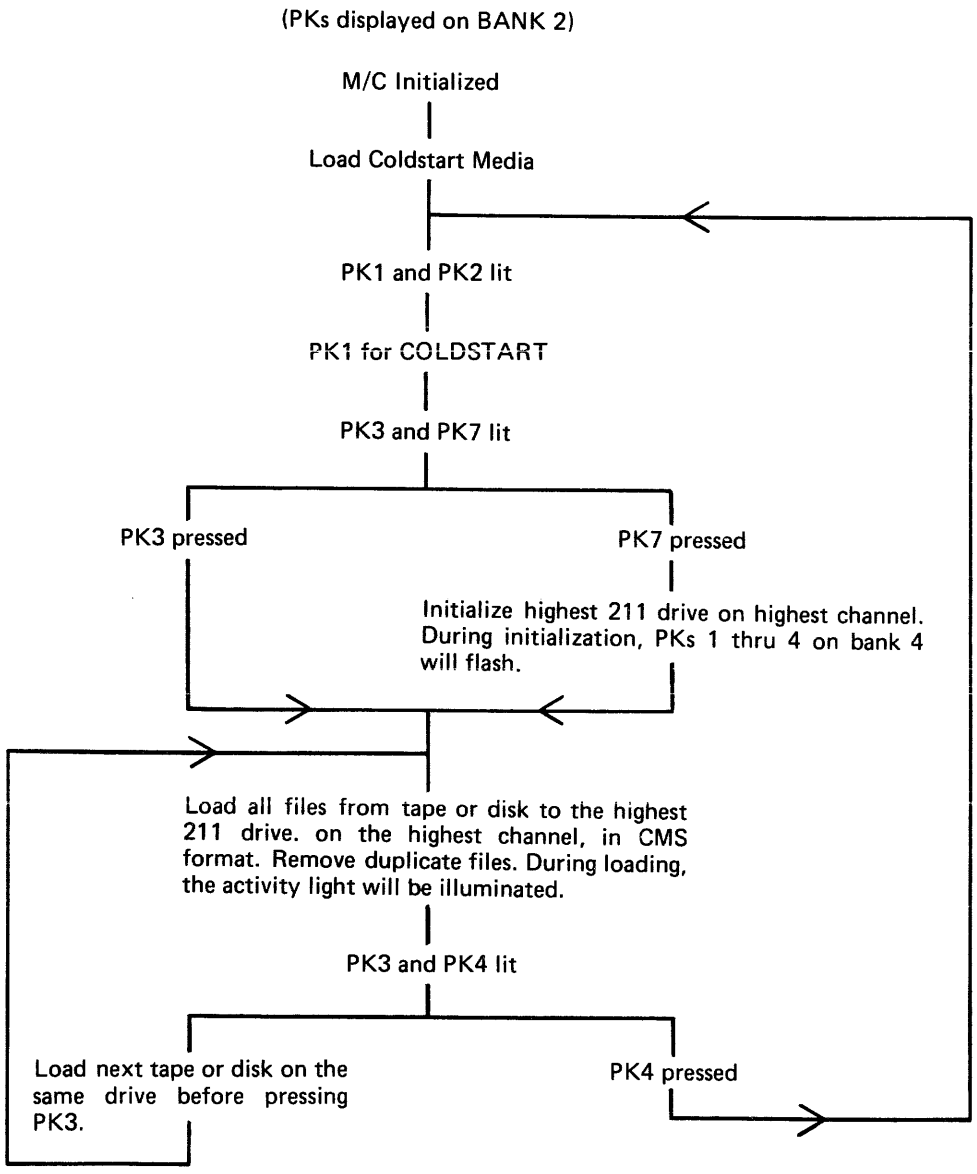


Figure 8-3. Outline of COLDSTART Utility

Initialization Parameters for 211 Fixed Disk

If the 211 is to be initialized, the following pre-defined set of parameters is used :

Disk name	SYSTEM
Number of files	500 – 5MB (Allocation Unit 1)
	750 – 10MB (Allocation Unit 1)
	1000 – 20MB (Allocation Unit 2)
	2000 – 40MB (Allocation Unit 4)
	2804 – 80MB (Allocation Unit 8)
Date	11/11/11
Owner	CMSB90
Serial number	111111

NOTE

The coldstart media must contain the file "CMSBOOT" (release 3.04 or greater) if the initialize option is required.

Error Messages

If, during the initialization of the 211 fixed disk or during the loading of the files onto the 211, any errors occur, the error light on BANK 6 is illuminated. BANK 4 displays the error number.

All non-fatal errors are signified by only one light illuminated on BANK 4. The file number is also displayed on BANK 2 in binary for non-fatal errors.

Fatal error numbers are displayed on two or three lights on BANK 4. The error messages displayed on BANK 4 are as follows (X represents a light illuminated):

Bank 4 Lights 1 2 3 4 5 6 7 8	Description	Recovery
X O O O O O O O	Read error on Coldstart medium	See note 1
O X O O O O O O	Blocksize in label is smaller than expected (Magnetic tape only)	See note 1
X X O O O O O O	Corrupt directory on I/P (Disks only)	See note 2
O O X O O O O O	Temporarily Not Available (BSMDII only).	See note 1
X O X O O O O O	No CMSBOOT file found on Coldstart media.	See note 2
O X X O O O O O	Tape not LOAD/DUMP format (Magnetic tape only).	See note 2
X X X O O O O O	Directory I/O error on I/P (Disk only)	See note 2
O O O X O O O O	File not present on tape but appears in directory of tape.	See note 1

(continued)

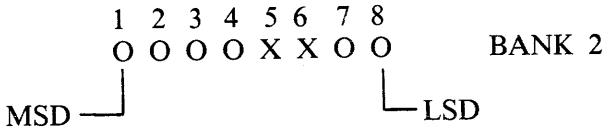
Bank 4 Lights 1 2 3 4 5 6 7 8	Description	Recovery
X O O X O O O O	No tape named "CLDSTRT" on-line (Magnetic tape only).	See note 2
O X O X O O O O	Coldstart medium that the bootstrap informed the utility to load from is not available.	See note 2
X X O X O O O O	Coldstart medium device controller problem.	See note 2
O O X X O O O O	Read error while reading CMSBOOT.	See note 2
X O X X O O O O	Multi-reel tape loaded. If this is the first reel of a multi-reel tape, all the files (except the last file, which will overlap onto the next reel), will be copied.	See note 2
O X X X O O O O	Disk bad – in "DANGER" state (BSMDII and 5.25 inch).	See note 2
X X X X O O O O	Disk expiring (BSMDII and 5.25 inch).	See note 2
O O O O X O O O	Read error on 211.	See note 1
O O O O O X O O	Cannot allocate required areas.	See note 1
O O O O X X O O	Initialize was not required, but no CMS standard 211 on-line.	See note 2
O O O O O O X O	Areas wrong for output.	See note 1
O O O O X O X O	Mandatory interrupt on 211.	See note 2
O O O O O X X O	Name list full.	See note 2
O O O O X X X O	Directory I/O error on 211.	See note 2
O O O O O O O X	Cannot load dual pack files.	See note 1
O O O O X O O X	Device Error : Drive inoperable Disk not present 211 switched off after initialization.	See note 2
O O O O O X O X	Output failure – Bootstrap not copied	See note 2
O O O O X X O X	Disk Bad – 211 in "DANGER" state.	See note 2
O O O O X O X X	211 fixed disk expiring.	See note 2
O O O O O X X X	211 fixed disk device controller problem.	See note 2

NOTE 1 – Non-fatal Error Handling

To continue with the loading of files, the user must press the RESET key after noting the file number displayed on BANK 2 (see below). The file which was being loaded when the error occurred is removed from the 211 fixed disk and the COLDSTART utility goes on to the next file on the tape or disk. If the utility fails to load a file which would be a duplicate, the original copy of the file remains on the 211 fixed disk.

The file number of the file in error is displayed, in binary, on BANK 2 of the MTR keyboard. (The file number is reset to zero for each new disk/tape presented to the utility for loading.) In the event of the file number exceeding 255, the file number is given mod 256.

Example: If BANK 2 displays the following:



(where X represents an illuminated PK)

This implies that an error occurred when reading or writing file 12.

To find the name of the file, refer to a directory listing for the tape or disk. Tape directories can be produced by LD at dump time or by TAPELR or TAPEPD. Disk directories can be produced by PD, KA, LR or by the LS option in SAU.

NOTE 2 – Fatal Error Handling

To clear the fatal error, the user is required to press the initialization button to terminate the utility and return to the initial state with PK lights 1 and 2 on BANK 2 illuminated.

STAND-ALONE UTILITIES (SAU)

SAU contains the necessary functions to prepare a disk for use on the B 90. These functions are used to initially transfer system software (MCP, interpreters, utilities and compilers) to a new disk (provided a removable media disk drive is present on the system) at a new installation, or to install a new level of system software at an existing installation.

SAU provides the facility to initialize (IN) or reformat (RF) disks and to copy (COPY) files from disk to disk.

SAU functions are loaded into memory from disk and operate independently of the MCP.

Loading Stand-Alone Utilities

From the initial state (PK1 and PK2 lit), ensure there is a disk containing SAU on the system. Press PK2, which loads the Warmstart Bootstrap (see ROM Scanning Algorithm, level 4 or 5). PKs 3 through 6 will be lit when the Warmstart Bootstrap is loaded.

Press PK6. The system will search for a disk file called SAU. For details of the search algorithm used, refer to Warmstart Bootstrap Scanning Algorithm.

For failures in search, see System Load Errors.

SAU from a Console

If the system has a console, SAU uses the console printer to display messages and the console keyboard for input. When SAU has finished its internal initialization, the ALPHA and READY lights are illuminated.

All functions and parameters must be entered with no leading or trailing spaces, and must be terminated with the OCK1 key. The RESET key clears all keyboard input since the last OCK key.

SAU from a Terminal

If the system does not have a console, SAU looks for a Data Comm Power Pak on channel 5. SAU tries to establish a data comm link with a suitably configured terminal (see Terminal Configuration for SAU). When the data comm link is made, the ALPHA and READY lights on the MTR keyboard are illuminated. They remain illuminated until the user either issues a "PO" or a "WS" command.

When SAU is running from a terminal, it scrolls from the bottom of the screen.

All functions and parameters must be entered in the top left hand corner of the screen, with no leading or trailing spaces, and must be transmitted using the XMT key.

The operation of SAU on a terminal is similar to operation on a console printer, except that on a terminal the XMT key is used to perform the function of the OCK1 key of the console keyboard.

SAU Header Message

```
STAND-ALONE UTILITY
VERSION n.n.n yyddd
REQUEST "HELP" FOR FUNCTION SUMMARY
FUNCTION
```

NOTE

"REQUEST "HELP" FOR FUNCTION SUMMARY" appears on status line of terminals.

SAU Functions

1. IN — Initialize a disk and place the B 90 Warmstart Bootstrap on track zero.
2. RF — Reformat a disk to initial state and place the B 90 Warmstart Bootstrap on track zero.
3. LD — Load files to disk from cassette.
4. COPY — Copy files from disk to disk.
5. RM — Remove disk files.
6. LS — List the file names and their sizes (in sectors) of files resident on a disk.
7. RL — Relabel a disk.
8. OL — List the status of all drives on the system.
9. WS — Initialize the system and load the Warmstart Bootstrap (PKs 3 through 6 lit).
10. FE — Initialize a disk for MTR use (for Customer Service Engineers' use).
11. PO — To return to the initial state (PK1 and PK2 lit).
12. CH — Change the file-name of a disk file.
13. CLEAN — Clean the read/write heads of a BSMD drive.
14. COMPARE — To compare disk files.
15. PDX — Display hexadecimal dump of a disk directory or any sector on the disk.
16. DISCOPY — Duplicate a BSMII disk.
17. CI — Initialize a disk and place the B 90 Coldstart Bootstrap in track zero. For use with the B 90 Coldstarting procedures.

The syntax and a complete description of each of the functions is contained later in this section.

NOTE

Only one function can be run at any one time. When operating SAU, only input parameters may be entered or a function when SAU requests it

Terminal Configuration for SAU

If SAU is to be run from a terminal, a TD 830 device or equivalent (MT 983) must be installed on channel 5 using Burroughs Two Wire Direct Interface (TDI) and poll/select line discipline. The line speed configuration may be one of:

4800 bits per second
9600 bits per second
19200 bits per second
38400 bits per second

The required TD 830 system registers for use with SAU are as follows:

```
0080    0A nn 0A 00 17 4F 1F 1E
        10 3x  3y 04 nn 00 04  04

0090    04 2F 4C 17 4F FF 20 47
        17 40  00 08 03 50 0F FF

00A0    00
```

NOTE

x, y must lie in the range 0-9. These refer to the terminal address.

nn refers to the line-speed register code. The codes relating to particular line-speeds can be found in the relevant TD 830 or MT 983 documentation.

The required MT 983 system registers for use with SAU are as follows:

```
0080    0A A1 0A 00 17 4F 1F 1E
        10  30  31 04 08 00 04 04

0090    04 2F 4C 17 0F 00 20 47
        17 00  00 08 00 00 07 80

00A0    00
```

NOTE

MT 983 firmware levels 3.0 and above require byte 8D to be set for language option. Refer to MT 983 documentation.

Common SAU Output Messages

Message	Possible Causes	Suggested Action
FUNCTION ABORTED	Hardware error.	If this message is preceded by another message then follow that message, else notify customer service engineer.
Dxy DEVICE ERROR DRIVE INOPERABLE	Specified drive was accidentally made Not Ready; or drive unit malfunction.	Ready drive; or notify customer service engineer.
Dxy DEVICE ERROR OFF CYLINDER	Drive unit malfunction.	Notify customer service engineer.
Dxy DEVICE ERROR WRITE INHIBITED	Disk in specified drive has its write lockout indicator set.	Ensure that the Write Lockout Hole (BSMD) is covered, or for cartridge – the Write Lockout Plug is flush with surface of cartridge.
Dxy DEVICE ERROR SEEK TIMEOUT	Drive unit malfunction.	Notify customer service engineer.
Dxy DEVICE ERROR CONTROLLER PROBLEM	Drive unit malfunction.	Notify customer service engineer.
FUNCTION	Prompt to request operator to enter next desired function.	Enter next function, or "HELP" to display list of functions.

Note:

x identifies type of disk
(M = BSMD or BSMDII, F = FIXED, K = CARTRIDGE)

y identifies unit
(A, B, C etc.)

Disk I/O Errors During SAU

These errors are identified by the messages:

WRITE ERROR,
READ ERROR,
<unit> DIRECTORY I/O ERROR
<unit> DEVICE ERROR

which may be encountered while running the Stand-Alone Utilities.

Disk I/O errors indicate a failure to read from (READ error) or write to (WRITE error) disk. Such problems should not be allowed to persist on a disk which is to be used to store important information, especially where the disk is to be used as a systems disk. Therefore an explanation or fix is required before the drive and disk can be considered acceptable for live use. Even if the Stand-Alone Utility continues to run satisfactorily, there may be some form of disk corruption. After any of these errors, the media involved should be checked for any corruption which might cause future system problems (for example, the CHECK.DISK, KA or DA utilities under MCP control).

A Note On Dual Pack Files

A "dual pack file" is a file which resides on two separate disks or logically identified disks (for example, DFA and DFB).

A dual pack file consists of:
A disk file header on each of the two disks.
At least one and at most sixteen file areas, each of which may be allocated on either disk.

Both parts have the same file name.

Under MCP control, the file may be opened only if both parts are present.

Each file header contains a reference (the pack-id) to the other disk.

Therefore, if for any reason one part of the dual pack file is lost, or if the pack-id of one of the disk volumes is changed, the file will be inaccessible under MCP control.

Caution must therefore be exercised when using SAU to initialize, reformat, or relabel any disk containing part of a dual pack file. Dual pack files may be located with the LS function.

In addition to the file-name and area occupied, the LS function will give, for each dual pack file, the overflow pack-id and the total overflow area.

If one part of a dual pack file is lost for any reason, the SAU RM function may be used to remove the remaining part.

The terms "master file" and "overflow file" are sometimes used to distinguish the two parts of a dual pack file. It should be remembered, however, that either part of the file may be regarded as the master file. In this section the term "master" file is used to indicate the part of the file mentioned in a COPY or RM command.

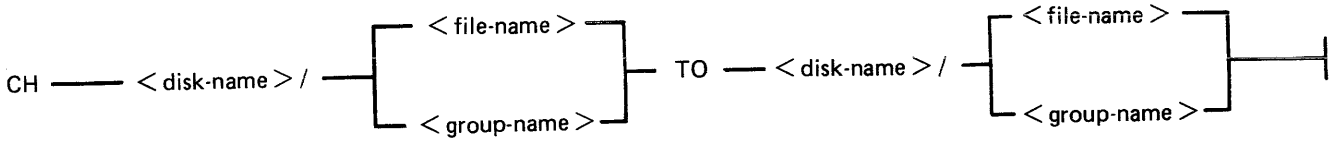
It is necessary that both "master" and "overflow" files reside on disks with the same allocation unit in order to copy them.

The 3.05 release will not allow the creation of new dual-pack files. Any access of an existing dual-pack file will result in a warning message.

CH (Change disk file name)

This function allows the operator to change the name of disk files.

Format:



Examples:

To change a file AR030 on a disk ARDISK to PR030X:

CH ARDISK/AR030 TO ARDISK/PR030X

To change a group of files starting with GL on a disk GLDISK to a group starting with AP:

CH GLDISK/GL= TO GLDISK/AP=

Output Messages

Message	Possible Cause	Suggested Action
file-name CHANGED TO file-name	Function successful.	None.
file-name NOT CHANGED ILLEGAL FILE-ID	file-name or group-name contains an illegal character.	Check input and re-enter.
file-name NOT CHANGED WRITE LENGTH OVERFLOW	The resulting file-name is longer than 12 characters (possibly during a group change).	Check the group-name and re-input.
file-name NOT CHANGED WRITE LENGTH UNDERFLOW	When changing a group-name to a smaller group-name, the resulting file has no characters (for example, changing F= to = when a file called "F" is present).	Correct the group-name and re-input if necessary; or enter another CH for the file not changed.
DUPLICATE FILE file-name	An attempt was made to change a file to the name of an already-existing file.	Correct the input and re-enter.
file-name NOT FOUND – file-name NOT CHANGED INCOMPLETE DUAL PACK FILE	The file is a dual-pack file and the overflow pack is mounted but the file is not present.	Investigate why the other part of the file is missing.
PACK disk-name NOT ON LINE – file-name NOT CHANGED – INCOMPLETE DUAL PACK FILE	The file is a dual pack file but the overflow pack is not present.	Make the overflow pack present and re-input.

(continued)

Message	Possible Cause	Suggested Action
file-name NOT CHANGED DUPLICATE FILE ON OVERFLOW PACK	An attempt was made to change the name of a dual-pack file to the name of a file already present on the overflow pack.	Correct the input and re-enter.

CI (Initialize a Disk with Coldstart Bootstrap)

The CI function only initializes mini disks (1 MB BSMD, 3 MB BSMII).

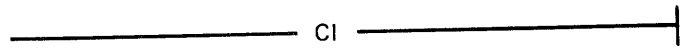
The CI function initializes mini disks for use with COLDSTARTing a system (see ROM Scanning Algorithm for level 5 ROMs and COLDSTART utility).

The CI function performs this disk initialization. When all disk parameters have been requested, CI checks the recording surface of the disk by writing and reading test patterns to each sector of the disk. Any "bad sectors" (unusable or unreadable) are removed from the disk's available area table, and the number of bad sectors is displayed to allow badly worn disks to be discarded.

The function also writes a disk label containing information supplied by the operator to the appropriate prompts (see Output Messages), and creates a disk directory of the appropriate size required for the number of files specified, plus a single, SYSMEM, entry. Sectors 1 through 31 on track zero are loaded with the Coldstart Bootstrap from a file called "CMSBOOTxxxxx" which can be on any on-line disk or the first file on any cassette labelled "SYSB90".

The "xxxxx" characters are ignored by the utility; only the first seven characters are compared when the on-line disks are searched (that is, the file specifications searched for are equivalent to CMSBOOT=).

Syntax:



Output Messages

Message	Possible Causes	Suggested Action
DRIVE <3 CHARACTERS OR NULL>	Prompt looking for drive containing disk to be initialized.	Enter Drive mnemonic or Enter null to terminate CI function.
DATE <MM/DD/YY>	Prompt looking for the date of initialization.	Enter date (format MM/DD/YY).
FILES <NUMBER LESS THAN 2805>	Prompt looking for maximum number of files for this disk.	Enter maximum number of files this disk will contain (less than 2805).
SERIAL <6 DIGITS>	Prompt looking for disk serial number.	Enter serial number (6 numerals).
PACK <UP TO 7 LEGAL CHARACTERS>	Prompt looking for name to be given to disk.	Enter disk-name (up to 7 legal characters). Legal characters include 0-9, A-Z, . (dot), – (dash).
OWNER <UP TO 14 CHARACTERS>	Prompt looking for user-defined information.	Enter valid information (up to 14 characters).
BOOTSTRAP VERSION n.nn.nn USED END CI	CI End of Job.	None.

(continued)

Message	Possible Causes	Suggested Action
NO OF BAD SECTORS number	Disk was successfully initialized. If any bad sectors were detected, the number is displayed.	None.
TRACK 0 BAD	Sector(s) in Track zero are bad and cannot be used. CI will end, as a CMS disk must contain reserved information in Track 0.	Select another disk to be initialized with CI.
CMSBOOT NOT PRESENT LOAD CMS BOOTSTRAP AND HIT OCK/XMT TO CONTINUE	System failed to find the file "CMSBOOTxxxxx".	Load another drive with a disk containing CMSBOOT or load a cassette with the name "SYSB90" containing CMSBOOT. Then press OCK1 or XMT. System will re-attempt search for CMSBOOT.
I/O FAILURE – BOOTSTRAP NOT COPIED	A hardware error has occurred.	Place disk with CMSBOOT into another drive and try CI again.
NO OF BAD SECTORS EXCEEDS 50	This disk is unusable.	Select another disk to be initialized.
NO COLDSTART BOOTSTRAP IN CMSBOOT, LOAD CORRECT FILE AND HIT OCK/XMT TO CONTINUE.	The first file found called CMSBOOT did not contain the COLDSTART bootstrap.	Load a disk which has a CMSBOOT level 3.04 or greater. Also check that no other file called CMSBOOTxxxxx is present on the system.

CLEAN (Clean BSM Drive Read/Write Heads)

The read/write heads of the one megabyte mini drive are cleaned by this function. This cannot be used on any other type of drive. Each head is cleaned in turn by the Burroughs head cleaning diskette.

The procedure the system follows to clean one head is as follows:

1. Load head onto cleaning surface.
2. Sequentially access the disk from the outermost track to the innermost.
3. Sweep heads from the outer to the inner track and back, ten times.
4. Unload head.

The current position of the head on the disk is visually displayed by illuminated PK lights.

Example:

```
CLEAN
```

Output Messages

Message	Possible Causes	Suggested Action
DRIVE	Prompt requesting operator to insert cleaning disk into proper drive.	Place cleaning disk into proper drive. Make certain disk is write inhibited (write inhibit hole is open and is positioned at top corner of disk).
LOAD WRITE INHIBITED CLEANING DISK AND TRANSMIT "OK" TO CONTINUE	Cleaning disk was inserted into drive.	Enter "OK" plus OCK1 to start cleaning procedure; or press OCK1 to terminate the function.
HEAD 1 CLEANED. FLIP WRITE INHIBITED CLEANING DISK AND TRANSMIT "OK" TO CONTINUE.	Head 1 has been cleaned.	Turn disk upside down (write inhibit hole to bottom) so that cleaning surface faces other side; then enter "OK" and press OCK1 to begin cleaning functions.
HEAD 0 CLEANED END CLEAN	Both heads have been cleaned. Function ends. Remove cleaning disk and store.	

Any attempt to clean head 0 before head 1 will result in the "LOAD/FLIPDISK" message being prompted.

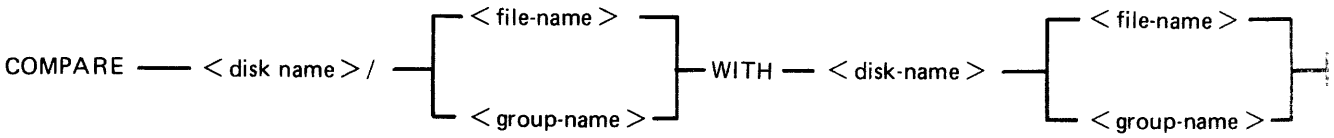
WARNING

Any attempt to use a write-enabled disk or any disk other than the cleaning disk during execution of this function will result in serious damage to the disk drive.

COMPARE (Compare two disk files)

This function allows the operator to compare files present on disk.

Format:



Examples:

To compare a file PR200 on disk PRDISK with another file PR200 on disk USER:

COMPARE PRDISK/PR200 WITH USER/PR200

To compare a group of files starting with IN on the disk INDISK with a group of files starting with XY on the disk XYZ:

COMPARE INDISK/IN= WITH XYZ/XY=

Before the files are compared, a check is made that the file sizes and number of areas are consistent. If any conflicts are found here, the function will be terminated.

Output Messages

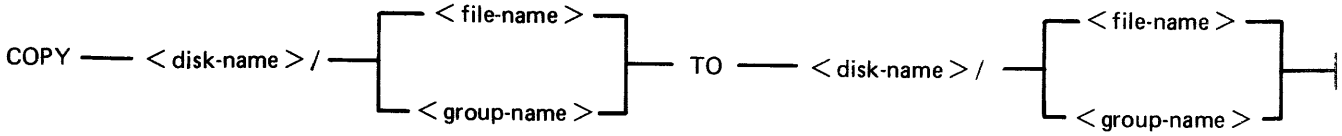
Message	Possible Cause	Suggested Action
file-name NOT FOUND	Self-explanatory.	Check input.
FILE SIZE CONFLICTS – END COMPARE	One file contains more records than the other.	None.
CONFLICTIONS IN NUMBER OF AREAS – END COMPARE	The number of areas assigned to the two files are different.	None.
disk-name/file-name COMPARED WITH disk-name/file-name – NO CONFLICTS FOUND	Successful file comparison.	None.
disk-name/file-name COMPARED WITH disk-name/file – CONFLICTS FOUND	The compare was unsuccessful.	Check SPO and error logs for any errors during copying of either file; re-copy file(s) from backup disks and re-run the COMPARE function.
PACK disk-name NOT ON LINE – file-name NOT COMPARED – INCOMPLETE DUAL PACK FILE	Comparison of dual pack file was required and the overflow pack was not on line.	Make the overflow pack present and re-input COMPARE function.
file-name NOT FOUND file-name NOT COMPARED	Comparison of dual pack file was required and the overflow	Investigate why the file was not present on the overflow pack.

(continued)

Message	Possible Cause	Suggested Action
INCOMPLETE DUAL PACK FILE	pack was present but did not contain the required file.	
END COMPARE	End-of-job of this SAU function.	None.

COPY (Copy files disk to disk)

This function allows the operator to copy files from one disk to another. Overflow files will not be copied.



Examples:

To copy a single file:

COPY PR1/PR200 TO PRBU/PR200

To copy a group of files:

COPY PR1/PR= TO PRBU/PR=

To copy all files on one disk to another:

COPY PR1/= TO PRBU/=

To copy a file and change its name:

COPY PR1/PR200 TO PRBU/PR200B

Output Messages

Message	Possible Causes	Suggested Action
DUPLICATE FILE NAME	System detected a file name on disk identical to that which the operator is attempting to copy. No two files with the same name may reside on a disk. Therefore specified file was not copied.	Normally, remove the existing file with RM; reattempt the COPY. Check input for correct specification of file names.
NAME LIST FULL	When this disk was initialized, the disk directory was constructed to contain a fixed number of file names. The directory has now reached its capacity.	The COPY function ends automatically. If possible, remove with RM any unwanted file(s) to make room in the disk directory. Then re-attempt to COPY the file which encountered the error; Replace the disk with a disk that has sufficient room in the disk directory.
Dxy DIRECTORY I/O ERROR	A read or write error was encountered while the system was attempting to access the directory of the specified disk.	1. Change the position of the disks and retry. 2. For a group copy, some files may be recovered using single

(continued)

Message	Possible Causes	Suggested Action
	The directory structure of the input (source) disk may be corrupted.	file copy. If these attempts fail, then type of error should be analyzed with DA utility and disk re-initialized before re-use.
WRITE DISK IS NOT WRITE PERMIT	Destination disk (disk to which system is copying) is write inhibited.	For cartridges: ascertain that write lockout plug is flush with outer surface of cartridge; For BSMD: ascertain that the write lockout hole is covered; Retry the COPY.
file-name NOT FOUND	Specified file-name is not on disk.	Check input – re-enter if necessary; Check for correct disk.
PACK disk-name NOT ON LINE	Specified disk is not on-line to the computer.	Check input – re-enter if necessary; Check for correct disk.
INVALID REQUEST	Typing error.	Check input and re-enter.
END COPY	Copy successful (EOJ); Specified group name to be copied is not on disk.	None if copy successful; Check input – re-enter if necessary; Check for correct disk.
file-name COPIED	COPY successful.	None.
CANNOT ALLOCATE AREAS FOR – file-name	Insufficient available space on disk to which operator is attempting to copy this file.	Using KA utility, analyze disk available space.
file-name NOT CHANGED – O/P LENGTH OVERFLOW	The resulting file-name is longer than 12 characters.	Correct the group name and re-input.
file-name NOT CHANGED – O/P LENGTH UNDERFLOW	The resulting file-name has no characters.	Correct the group name and re-input if necessary.
O/P ERROR – file-name	Error encountered while system was attempting to write specified file to disk.	Allow COPY to continue copying any remaining files. Then re-attempt copying the file in which the O/P error occurred.
READ ERROR – file-name	System encountered errors in the file from which it was reading.	Allow COPY to continue. Then use back-up copy of the specified file and re-attempt the COPY; switch the disks to the opposite drives.
file-name NOT COPIED: ILLEGAL FILE-ID	Specified file-name contains an illegal character (that is, / ?), therefore file will not be	Check input and re-enter if necessary.

(continued)

Message	Possible Causes	Suggested Action
	copied. Legal characters are 0-9, A-Z, . (dot), – (dash).	
file-name COPIED: ILLEGAL FILE-ID	Disk-to-disk copy is being made and this warning is given that the new file is copied but with an invalid file-name.	

Dual Pack Files

The copy function cannot create a dual pack file, however a complete dual pack file may be copied to a single disk.

The master file will be copied first. COPY will then look for a matching overflow file on the overflow pack. If a matching overflow file is found, it will be copied and the function will terminate with the message:

file-name COPIED FROM disk-name-1 AND disk-name-2

If the overflow pack is not on line, or the overflow file is not found, or an overflow file is found which does not match the master file, the function will print the following message, and wait for operator input:

disk-name-file-name IS AN INCOMPLETE DUAL PACK FILE
A MATCHING OVERFLOW FILE ON disk-name-2 IS NOT PRESENT.
PLEASE TAKE ONE OF THE FOLLOWING ACTIONS:
A. SUPPLY THE CORRECT OVERFLOW PACK AND TYPE "A" TO TRY AGAIN.
B. TYPE "B" TO SKIP THIS FILE.

At this stage the master file pack may be removed if necessary.

The operator should either power on the correct overflow pack and type "A" followed by OCK1 or type "B" followed by OCK1.

If the "A" option is selected, the function will repeat its search for the overflow file as above and either terminate normally or repeat the prompt to the operator.

If option "B" is selected, the output file will be purged and the function will terminate with the message:

file-name NOT COPIED, PART OF A DUAL PACK FILE

WARNING

If a level of SAU earlier than 3.00 is used to copy a dual pack file or a family containing a dual pack file, the output disk is liable to be seriously corrupted and to require re-initializing.

DISCOPY (Duplicate a BSMII Disk)

This function allows the operator to copy the contents of a Burroughs Super Minidisk II (BSMDII) to another BSMDII.

Format:

DISCOPY DMx TO DMx
where x = A, B, etc.

All information is copied from one BSMII to another, that is, disk label, bootstrap, directories and data. Thus a complete disk can be duplicated for backup purposes without initialization. The automatic relocation of bad sectors allows the duplicated directories to remain valid since all information on the duplicated disk will have the same logical, if not physical, address as that on the master disk.

Output Messages

Message	Possible Cause	Suggested Action
END DISCOPY	End of function.	None.
NON BSMII DISK SPECIFIED	Either of the disks specified is not a BSMII disk.	This function can only be performed on BSMII disks: use COPY for other kinds of disk, including BSM (1MB) disk.
PACK <up to 7 legal characters>	This is a prompt for the operator to give the seven character disk-name of the input disk.	Enter up to seven characters.
drive-mnemonic PURGED	An input error has occurred during the copy. The output disk is purged by overwriting track zero with "null" characters.	Investigate cause of error on the input disk, then re-run the DISCOPY function: the purged output disk may be re-used.
FUNCTION ABORTED – drive-mnemonic NOT PURGED	An I/O error has occurred on the output drive during the copy, and the copy is stopped.	Use the LS function to determine which files are successfully copied to the output disk: these may be recovered. Otherwise re-run the DISCOPY function using a different output disk.
drive-mnemonic COPIED TO drive-mnemonic	Function successful.	None.

FE (Initialize MTR Disk)

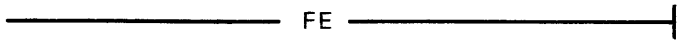
A virgin or formatted disk is initialized to CMS format with suitable sectors reserved for Maintenance Test Routines (MTRs). These sectors are denoted as bad sectors on a KA map of the disk. The surface is checked by writing and reading patterns to each sector. Bad sectors and MTR tracks are made unavailable. A disk label is written and the file directory is created with a single, SYSMEM, entry. Sectors 1 through 31 on track zero are loaded with the relevant MTR Bootstrap from a file called "CMSBOOTxxxxx" which can be on any on-line disk or the first file on any cassette labelled "SYSB90".

The "xxxxx" characters are ignored by the utility; only the first seven characters are compared when the on-line disks are searched (that is, the file specifications searched for are equivalent to CMSBOOT=).

In the case of a fixed disk, the CMS Bootstrap (Warmstart Bootstrap) is loaded from CMSBOOTxxxxx. When all other disks are initialized, FE loads the relevant MTR Bootstrap from CMSBOOTxxxxx.

The MTR tracks are not included in the number of bad sectors, but if the disk is reformatted (see RF), the total number of sectors (including those on MTR tracks) removed from the available table will be referred to as bad.

Syntax:



Output Messages

Message	Possible Causes	Suggested Action
DRIVE <3 CHARACTERS OR NULL>	Prompt looking for drive containing disk to be initialized.	Enter Drive mnemonic or Enter null to terminate FE function.
DATE <MM/DD/YY>	Prompt looking for the date of initialization.	Enter date (format MM/DD/YY).
FILES <NUMBER LESS THAN 2805>	Prompt looking for maximum number of files for this disk.	Enter maximum number of files this disk will contain (less than 2805).
SERIAL <6 DIGITS>	Prompt looking for disk serial number.	Enter serial number (6 numerals).
PACK <UP TO 7 LEGAL CHARACTERS>	Prompt looking for name to be given to disk.	Enter disk-name (up to 7 legal characters). Legal characters include 0-9, A-Z, . (dot), – (dash).
OWNER <UP TO 14 CHARACTERS>	Prompt looking for user-defined information.	Enter valid information (up to 14 characters).
BOOTSTRAP VERSION n.nn.nn USED END FE	FE End of Job.	None.

(continued)

Message	Possible Causes	Suggested Action
NO OF BAD SECTORS number	Disk was successfully initialized. If any bad sectors were detected, the number is displayed.	None.
TRACK 0 BAD	Sector(s) in Track zero are bad and cannot be used. FE will end, as a CMS disk must contain reserved information in Track 0.	Select another disk to be initialized with FE.
CMSBOOT NOT PRESENT LOAD CMS BOOTSTRAP AND HIT OCK/XMT TO CONTINUE	System failed to find the file "CMSBOOTxxxxx".	Load another drive with a disk containing CMSBOOT or load a cassette with the name "SYSB90" containing CMSBOOT. Then press OCK1 or XMT. System will re-attempt search for CMSBOOT.
BAD MTR TRACK	An MTR track contains a bad sector.	Select another disk to be initialized with FE.
CYL xxx BAD	In cases where certain cylinders are used for MTR purposes but are not removed from the available table, discovery of a bad sector will terminate the function.	Select another disk to be initialized by FE.
I/O FAILURE – BOOTSTRAP NOT COPIED	A hardware error has occurred.	Place disk with CMSBOOT into another drive and try FE again.
NO OF BAD SECTORS EXCEEDS 50	This disk is unusable.	Select another disk to be initialized.

IN (Initialize a Disk with Warmstart Bootstrap)

The Master Control Program (MCP) requires that any disk to be used on the system has a valid CMS disk label, disk directory, available area table, and a valid Bootstrap code in track zero. In addition, each sector of the disk must be initialized with its address.

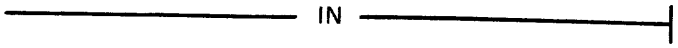
The IN function performs this disk initialization. When all disk parameters have been requested, IN checks the recording surface of the disk by writing and reading test patterns to each sector of the disk. Any "bad sectors" (unusable or unreadable) are removed from the disk's available area table, and the number of bad sectors is displayed to allow badly worn disks to be discarded.

The function also writes a disk label containing information supplied by the operator to the appropriate prompts (see Output Messages), and creates a disk directory of the appropriate size required for the number of files specified, plus a single, SYSMEM, entry. Sectors 1 through 31 on track zero are loaded with CMS bootstrap (Warmstart Bootstrap) from a file called "CMSBOOTxxxxx" which can be on any on-line disk or the first file on any cassette labelled "SYSB90".

The "xxxxx" characters are ignored by the utility; only the first seven characters are compared when the on-line disks are searched (that is, the file specifications searched for are equivalent to CMSBOOT=).

Since all fixed disks must be suitable for MTR purposes, the functions IN and FE are identical in this particular case. The relevant MTR tracks are checked and/or removed and the CMS Bootstrap (Warmstart Bootstrap) is written into sectors 1 through 31 of track zero.

Syntax:



Output Messages

Message	Possible Causes	Suggested Action
DRIVE <3 CHARACTERS OR NULL>	Prompt looking for drive containing disk to be initialized.	Enter Drive mnemonic or Enter null to terminate IN function.
DATE <MM/DD/YY>	Prompt looking for the date of initialization.	Enter date (format MM/DD/YY).
FILES <NUMBER LESS THAN 2805>	Prompt looking for maximum number of files for this disk.	Enter maximum number of files this disk will contain (less than 2805).
SERIAL <6 DIGITS>	Prompt looking for disk serial number.	Enter serial number (6 numerals).
PACK <UP TO 7 LEGAL CHARACTERS>	Prompt looking for name to be given to disk.	Enter disk-name (up to 7 legal characters). Legal characters include 0-9, A-Z, . (dot), – (dash).
OWNER <UP TO 14 CHARACTERS>	Prompt looking for user-defined information.	Enter valid information (up to 14 characters).

(continued)

Message	Possible Causes	Suggested Action
BOOTSTRAP VERSION n.nn.nn USED END IN	IN End of Job.	None.
NO OF BAD SECTORS number	Disk was successfully initialized. If any bad sectors were detected, the number is displayed.	None.
TRACK 0 BAD	Sector(s) in Track zero are bad and cannot be used. IN will end, as a CMS disk must contain reserved information in Track 0.	Select another disk to be initialized with IN.
CMSBOOT NOT PRESENT LOAD CMS BOOTSTRAP AND HIT OCK1/XMT TO CONTINUE	System failed to find the file "CMSBOOTxxxxx".	Load another drive with a disk containing CMSBOOT or load a cassette with the name "SYSB90" containing CMSBOOT. Then press OCK1 or XMT. System will re-attempt search for CMSBOOT.
I/O FAILURE – BOOTSTRAP NOT COPIED	A hardware error has occurred.	Place disk with CMSBOOT into another drive and try IN again.
NO OF BAD SECTORS EXCEEDS 50	This disk is unusable.	Select another disk to be initialized.

LD (Load files from cassette to disk)

This function allows the operator to load all files from a dump tape to a disk. A dump tape is one produced by the DUMP or UNLOAD functions of the utility "LD" (which runs under MCP control). Each sector of data written to disk is verified. The SAU LD function cannot create a dual pack file.

Format:

LD <disk-name> FROM <library-tape-name>

Example:

LD ARDISK2 FROM ARTAPE

Output Messages

Message	Possible Causes	Suggested Action
file-name LOADED	Displayed for each file loaded and verified.	None.
LOAD REEL number	Dump tape is a reel from a multi-reel dump and the next reel is required.	Supply next reel.
END LD	LD (EOJ)	None.
NOT DUMP TAPE	Specified tape is not a correctly formatted dump tape. Function ends.	Supply correctly formatted dump tape; re-initiate LD.
multi-file-name NOT PRESENT. LOAD CASSETTE AND PRESS OCK OR ENTER NULL TO TERMINATE	Specified tape is not installed and ready; function ends.	Install and ready tape; re-initiate LD; or press OCK1 to retry search for multi-reel name tape; or press any key plus OCK1 to terminate LD.
IRRECOVERABLE CASSETTE ERROR	Error was encountered while system was attempting to read cassette. Normally caused by accidentally opening cassette drive unit. Function ends.	Ready cassette drive unit; re-initiate LD.
PACK disk-name NOT FOUND	Specified disk is not ready; specified disk is not on-line to computer.	Ready the disk; Check for correct disk.
O/P DISK NOT WRITE PERMIT	Disk to which files are to be loaded is write protected. Function ends.	Ensure that write lockout hole is covered (for BSMD); For cartridge, ensure write lockout plug is flush with surface of disk cartridge; re-initiate LD.
DUPLICATE FILE NAME – file-name	File of specified name already exists on destination disk. A disk may not contain two files with the same name. LD continues loading other files, ignoring any duplicates.	Remove (with RM) the existing file from disk, then re-initiate LD.

(continued)

Message	Possible Causes	Suggested Action
CANNOT ALLOCATE AREAS FOR – file-name.	No appropriately sized available area on the destination disk for specified file.	If specified file is desired, replace this disk with a disk having more available area. Then re-initiate LD.
O/P ERROR – file-name	Disk write error encountered while system was loading specified file.	Place disk in opposite drive and re-initiate LD.
AREA SIZES TOO SMALL FOR – file-name	Insufficient available area on disk for specified multi-area file.	If specified file is desired, replace this disk with a disk having more available area. Then re-initiate LD.
NAME LIST FULL	No space remaining in disk directory for another file name. Function ends.	Use another disk, or remove unwanted files with the RM function.

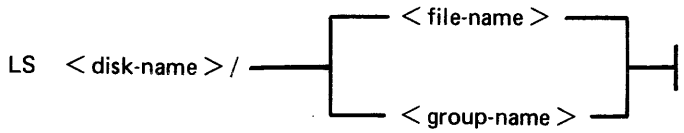
LS (List File Sizes)

This function allows the operator to print names and sizes in sectors of files on a specified disk. For dual pack files, both portions of the files and their sizes are printed. If all file sizes on a disk are printed by:

LS <disk-name>/=

then all available areas of that disk are printed at the end of the list.

Railroad diagram:



Example:

To list file sizes of all the files on the disk ARDISK2

LS ARDISK2/=

Output Messages

Message	Possible Causes	Suggested Action
PACK disk-name NOT ON LINE	Specified disk is not on line to the computer.	Check input – re-enter if necessary; Check for correct disk.
END LS	LS successful (EOJ)	None.
Dxy DIRECTORY I/O ERROR	A read or write error was encountered while the system was attempting to access the directory of the specified disk. The directory structure of the disk may be corrupted.	

OL (Print Status of Drives)

This function allows the operator to print the status of all cassette and disk drives.

Example:

OL

Output Messages

Message	Possible Causes	Suggested Action
END OL	OL successful (EOJ).	None.
disk drive disk-name	Specified disk is resident in specified drive.	None.
disk drive NOT READY	There is no disk in this drive; the disk in the drive has not been set to run; the disk in the drive has not come up to proper speed; system does not recognize disk.	If applicable – set disk to run; or allow disk to attain proper speed; make certain disk has a proper "label".
disk drive NOT CMS-STANDARD	Disk in this drive does not have a valid CMS label.	Disk must be initialized with IN to create valid CMS label.
cassette drive NOT READY	There is no cassette in this drive; cassette has not been loaded into drive properly.	Check for proper loading of cassette in drive.
cassette drive UNLABELLED	Cassette in specified drive does not have a valid label.	Use PG to create valid cassette label.
cassette drive multi-file-name file-name	Cassette in this drive has the specified name.	None.
disk-drive TEMPORARILY NOT AVAILABLE	The door of the disk drive has been opened.	Close the door.

MTR cassettes do not have labels corresponding to the correct CMS format and thus will appear "UNLABELLED".

PDX (Print Disk Directories)

This function allows the operator to print the disk directories, disk label and any sector in hexadecimal.

Format:

PDX

The function operates in interactive mode. The operator is prompted to supply the input.

Output Messages

Message	Possible Cause	Suggested Action
DRIVE	This is a prompt to the operator.	Enter the device mnemonic of the disk to be read; for example, DKA.
IS DIRECTORY REQUIRED <Y OR N>	This is a prompt to the operator.	Enter "Y" if a complete directory print is required, otherwise enter "N".
SECTOR <6 character hex value or null>	The operator is prompted to provide the hex address of a sector to be displayed. This prompt is repeated until a null response (just an OCK key) is given.	Enter the 6-character hex sector address followed by OCK1, or just OCK1 to terminate the function.
COMPLETE <Y OR N>	The operator is prompted to state if printing of all sectors in the disk directory is required or only the first sector of each entry.	Enter "Y" or "N" as appropriate.
LABEL CORRUPT	The check-string "SL9INTERNL" in sector zero has not been found.	None: use backup copy of disk for normal use.
END PDX	End of PDX function.	Enter desired SAU function.

PO (Power Off)

This function allows the operator to terminate the execution of the Stand-Alone Utilities.

Example:

PO

The utility displays the message:

END STAND ALONE UTILITY

and causes the B 90 to return to the initial state (PK1 and PK2 lit).

RL (Relabel a disk)

This function allows the operator to change a disk's name without affecting the remaining contents of the disk.

Format:

RL <disk-name>

Example:

To relabel a disk called AP2 to BQ3 enter:
RL AP2
Then enter "BQ3" after the utility has displayed/printed "PACK"

Output Messages

Message	Possible Causes	Suggested Action
END RL	RL successful (EOJ).	None.
PACK	Prompt to request operator to enter the new disk name.	Enter new disk name (up to 7 legal characters).
PACK disk-name NOT ON LINE	Specified disk is not on line to the computer.	Check input – re-enter if necessary; Check for correct disk.
Dxy DIRECTORY I/O ERROR	A read or write error was encountered while the system was attempting to access the directory of the specified disk. The directory structure of the disk may be corrupted.	

Warning

If RL is used on a disk containing part of a dual pack file, that file will be inaccessible under MCP control.

There are two ways to prevent this from occurring:

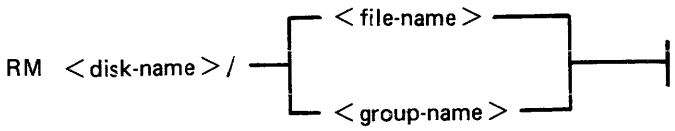
1. Remove the file before relabelling the disk.
or
2. Copy the entire file to a single pack file before relabelling.

There are two ways to resolve the problem if it does occur:

1. Use Stand-Alone Utility (RM) to remove the file.
or
2. Use Stand-Alone Utility (RL) to relabel the disk to its original name.

RM (Remove Disk Files)

This function allows the operator to remove files from disk.



Examples:

To remove a single file:

RM PR1/PR200

To remove a group of files:

RM PR1/PR=

To remove all files from disk:

RM PR1/=

Unlike the RM utility that runs under MCP control, the Stand-Alone RM will remove system software (MCPX, COBOLINTX, BILINTERPX, etc.) without a warning message, the same as it removes other programs.

Output Messages

Message	Possible Causes	Suggested Action
file-name REMOVED	RM successful.	None.
PACK disk-name NOT ON LINE	Specified disk is not on-line to the computer.	Check input – re-enter if necessary; Check for correct disk.
file-name NOT FOUND	Specified file-name is not on disk.	Check input – re-enter if necessary; Check for correct disk.
END RM	RM successful (EOJ); Specified group name to be removed is not on disk.	None if RM successful. Check input – re-enter if necessary. Check for correct disk.

Dual Pack Files

If a file to be removed has an overflow area or another disk volume:

– If both parts of the file are available, both are removed and the following message is printed:

file-name REMOVED FROM disk-name-1 AND disk-name-2

– If the overflow file is not found for any reason, or if a file is found with the same file-id but which does not match the master file, then the following message is printed:

disk-name/file-name IS AN INCOMPLETE DUAL PACK FILE
A MATCHING OVERFLOW FILE ON disk-name-2
IS NOT PRESENT. PLEASE TAKE ONE OF THE
FOLLOWING ACTIONS:
A) SUPPLY THE CORRECT OVERFLOW PACK
AND TYPE "A" TO TRY AGAIN
B) TYPE "B" TO SKIP THIS FILE
C) TYPE "C" TO REMOVE THE INCOMPLETE FILE

If option "A" is selected, then either the complete file will be removed or the above prompt will be repeated.

If option "B" is selected, then the following message is printed:

file-name NOT REMOVED, PART OF DUAL PACK FILE

If option "C" is selected, then the following message is printed:

file-name REMOVED ONLY FROM disk-name-1

WS (Warm Start)

This function causes a "branch" to the warmstart routine stored in the Read Only Memory (ROM) of the machine. It will cause the CMS operating system (file-name MCPX) to be loaded from disk into memory.

Operating Procedure

Type "WS" then press OCK1. Wait for PKs 3, 4, 5 and 6 to be illuminated.

Press PK3.

Enter today's date in the format:
MM/DD/YY
when prompted to do so.

For further information, refer to the section headed "Loading the MCP".

LOADING THE MASTER CONTROL PROGRAM (MCP)

This process is also called the "Warmstart procedure".

The B 90 MCP code file is named "MCPX" (this is its executing name).

From the initial state (PK1 and PK2 lit), press PK2. This causes the ROM to load the Warmstart Bootstrap (see relevant ROM Scanning Algorithm). For errors, see Hardware Error Conditions and System Load Errors.

When the Warmstart Bootstrap has been loaded and executed, PK3 through PK6 will be lit. Press PK3.

The Bootstrap searches for a disk file called "MCPX". The search for the MCP proceeds as outlined in Warmstart Bootstrap Scanning Algorithm later in this section.

For failures in the search, see Hardware Error Conditions and System Load Errors.

If the MCP search is successful, the MCP is loaded to memory and MCP initialization takes place. The activity during this process can be distinguished on the D-lights (bank 5).

NOTE
The references to banks are for consoleless systems.

When the initial part of the MCP is loaded, the D2 (bank 5 PK2) light flickers.

Any printers on the system are initialized.

Automatic Volume Recognition (AVR) (see later in this section) is performed on all peripherals. During AVR, D2 flickers, with D4 through D8 lit (bank 5 PK2 flickers with PK4 through PK8 lit).

The MCP communicates to the operator via the SPO. The SPO device can be the console, SELF-SCAN or terminal, according to data in the SYSCONFIG file. The following describe the differences between the different SPO devices.

NOTE
The version of the MCP is identified by mark/level/patch numbers. For example, version 3.04.09 is mark 3, level 3.04, patch 3.04.09. A new software release is denoted by a higher level number (for example, 3.05). Within a release, higher patch numbers indicate improved versions of that level. For example, the application of two patches to 3.04.22 will create an MCP version 3.04.24 (see PATCHMAKER for details of how to patch MCP files).

Console and SELF-SCAN SPO Device

The system prints the MCP version on the SPO device, followed by a list of on-line peripherals that are powered up.

The system requests the date. Press the Ready Request button and verify that the ON, READY and ALPHA lights are lit. Enter the date as requested (mm/dd/yy), followed by OCK1 (leading zeros are optional). The system prints a date message.

The MCP automatically loads the program SYS-SUPERUTL (D2, D3, D4, D5 and D7 lit).

The message "COMMENCING LOG FILE CONSOLIDATION" is displayed as SYS-SUPERUTL zips the TL utility. For a full explanation of logging, refer to Logging in the 3.04 Release in Section 4 of this manual.

Optionally, depending on SYSCONFIG, a user program is loaded and executed.

The warmstart is complete and the MCP enters idle state (see below).

It is inadvisable to enter system commands until the complete warmstart procedure is over, otherwise confusion can result. Also, such input is not entered in the system log (if system message logging is declared in SYSCONFIG).

Example:

```
          * * * * * * * * * * * * *
          *  COMPUTER MANAGEMENT  *
          *  SYSTEM (CMS) 83328   *
          *  MCP VERSION 03.05.08 *
          * * * * * * * * * * * * *
            DFA SYSTEMS/ SYS DISK  2 FILES OPEN
            ENTER DATE AND TIME AS "MM/DD/YY  HHMM"
6/18/84
            18 JUN 84  84170  MON 1113
            COMMENCING TRANSFER OF OLD LOG FILES
            LOG INFORMATION TRANSFERRED
            END TL
            COMMENCING CREATION OF NEW LOG FILES
            LOGGING IS INITIATED ON 06/18/84 AT 11:25:48 (MCP VERSION 03;05;08)
```

For possible errors, see System Load Errors later in this section.

Terminal SPO Device

The MCP performs the same functions as described previously in Console and SELF-SCAN SPO Device, except that it cannot request a date to be input, as at this point there is no SPO device.

By entering CTL ON on one of the terminals, this terminal becomes controlling SPO.

By pressing the SPCFY key the user may move from the SPO page to the application page.

All other terminals with TD 830 capabilities may be non-control SPO at the same time. To obtain this function, depress the SPCFY key.

BASIC OPERATION UNDER MCP CONTROL

D-Lights (Bank 5)

During MCP execution, D-lights D1 to D8 give an indication of system activity, as shown in figure 8-4.

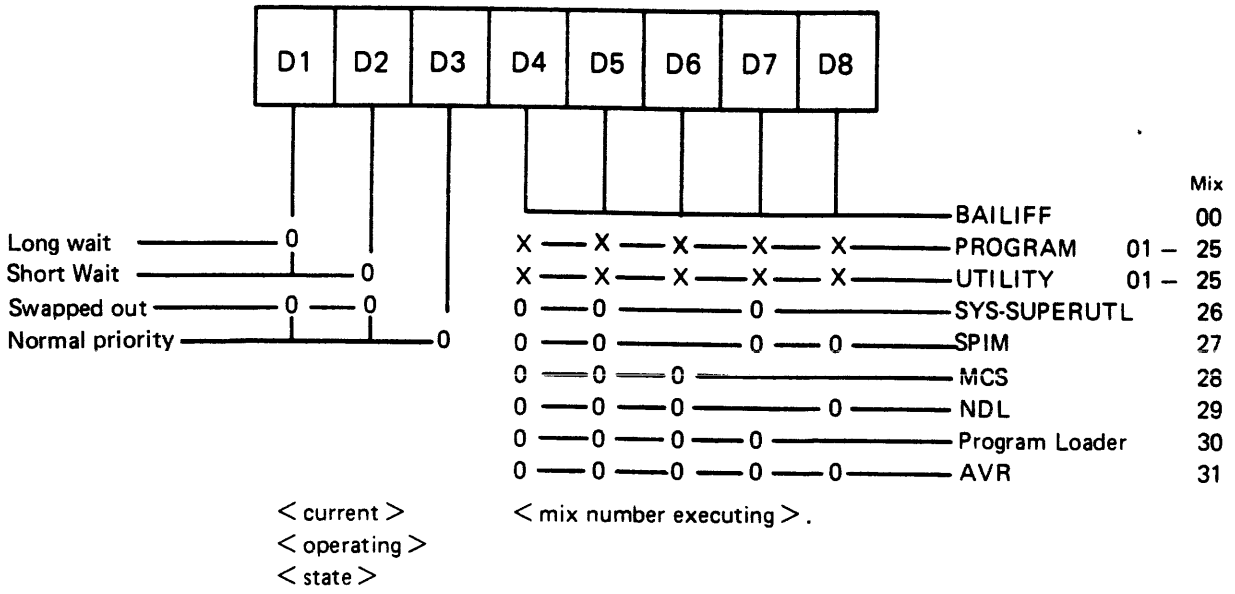


Figure 8-4. D-Light Display Interpretation

The light combination at any instant denotes the current function and state of that function. "0" denotes the light is ON and "x" denotes "DON'T CARE".

NOTE

The following tasks are not shown in the response to the MX intrinsic BAILIFF, Program Loader and AVR. Also, SYS-SUPERUTL is not shown in the response to the MX intrinsic unless a particular function is being performed. In this case, the MX response shows the name of the function, for example, 26/RM or 26/PD.

MCP States

When warmstart is complete, the MCP is either idle or executing a task. The idle state is identifiable by the absence of activity on the D-lights. In this state, three distinct patterns can be encountered:

D2, D3 only lit:
The last activity was a program which has gone to EOJ.

D1, D3, D4, D5, D7 only lit:
The last activity was a SYS-SUPERUTL function (IR, LB, LF, PD, RM, KX or CH).

D1, D3, D4, D5, D6, D7 only lit:
The last activity was a system intrinsic.

When the MCP is executing a task, the mix number is evident in D-lights D4 through D8 (see D-light display).

At any time when the system is idle (that is, when the mix is empty except for SPIM, MCS and NDL) it is valid to terminate the MCP by a PO of the system disk (see PO intrinsic).

Automatic Volume Recognition (AVR)

This procedure is carried out if any new media is loaded onto the system. If the procedure fails, the device is made not ready. The procedure varies for different media.

For fixed disk, disk cartridge, Burroughs Super Mini (BSM) disk, Burroughs Super Mini II disk and ICMD, AVR attempts to read the label.

For cassette, AVR searches for a CMS label or scratch label, otherwise the cassette is treated as unlabelled.

In all cases, if transient errors are suspected, Readying the peripheral (see RY intrinsic) re-initiates the AVR process.

Console Keyboard Under MCP Control

Under MCP control, the console keyboard may be used to enter system commands to the MCP, or solicited data to a program. Commands to the MCP may only be entered when the READY light is lit. Data to a program may only be entered on the alpha keyboard if the ALPHA light, but not the READY light, is lit. Input is terminated by an OCK key, or a PK key only if the corresponding PK light is lit. It can be seen that the keyboard can exist in one of three states: inactive, system enabled, or program enabled.

Since an understanding of the operation and behaviour of the keyboard and its indicators is essential, some important points are noted here.

If a disabled key is pressed at any time, or too much data is entered at once, then an error bell rings and the ERROR indicator lights. This error condition must be reset before any further keyboard input can be made.

The RESET key has two basic functions, resetting the error condition explained above, or clearing the information keyed since an OCK or PK key was last pressed. This key is necessarily enabled whenever any alpha or numeric key is enabled.

When some incomplete information has been keyed either to the MCP or a program, then it is necessary to terminate some information to this destination before entry to the other destination can be allowed. For example, once the READY light is lit an OCK key must be pressed to terminate input to the MCP before any input can be keyed to a waiting program. This restriction stands even if the RESET key is used.

If the keyboard is enabled for input, then the D-lights are not lit.

In order to enable keyboard input to the MCP, the READY ENABLE key must be pressed. This key is ignored if the SCL/LOADER routine of the MCP is currently executing a system command or loading a program (D4, D5, D6, D7 lit) and a wait of up to 30 seconds may be necessary before any keyboard input can be made. Similarly, if the MCP is very busy when a request for input is made, there

may be a delay of one or two seconds before the keyboard becomes enabled for input. A maximum of two characters can be entered in this time, or a keyboard error condition will arise.

If the system is being used with the self-scan screen operating as a SPO device as well as a console file, then the READY ENABLE key has an additional function. This key must be depressed in order to prompt the MCP to display either the complete screenful of messages on the screen, or the last message on the bottom line of the screen (scrolling the other information up one line). When the screen is being used as a console file, the screen displays information from the program using it until the READY ENABLE key is pressed.

If the screen is displaying SPO information, pressing an OCK returns the display to the console file information.

Interrupting the MCP

It is possible to interrupt the MCP while it is running. This informs the MCP that some action must be taken. The ability to interrupt the MCP is a good indication that the system is running satisfactorily. The only conditions where it may prove impossible to cause an interrupt is when the MCP is processing a system command or loading a program. Typical interrupts and the consequent processing include:

Press READY ENABLE button. This causes the MCP to enable the keyboard for input.

Opening or closing the serial printer cover causes the MCP to prevent or allow output to the printer accordingly.

Loading a disk or cassette causes the MCP to read the label and perform the AVR procedure.

POWER OFF

Logically power off all user disks (see PO command if under MCP control, or SAU PO command if under SAU control).

Logically power off the system disk (see PO command if under MCP control). Wait until the system returns to the initial state, that is, PK1 and PK2 are lit.

If the PO command cannot be used, due to some system error, then the system should be halted by pressing the Load Enable button, causing the system to return to the initial state with PK1 and PK2 lit.

Remove all removable disk media.

A mini disk can be removed as soon as the unit door is opened.
A disk cartridge can be removed only when the red stop light is lit, assuming that the drive is functioning correctly.

Power off the disk units (failure to remove disk media before this may result in subsequent media corruption).

Remove all cassettes from the system.

Power off the main cabinet (this must be the LAST action after all peripherals have been switched off).

Note on disk removal:

There are only two situations when it is valid to remove a disk:

where the MCP is not running and the disk is not in use.
where the MCP is running, but the disk is a user disk which is logically powered off after using the PO command: note that the PO command does not cause a disk to become logically powered off if it is in use, but the PO will be completed only after all activity on the disk is complete.

Note on power failures:

If the main cabinet is switched off accidentally (for example, by power failure), remove all disks and cassettes before it is switched back on.

ROM SCANNING ALGORITHM FOR ROM LEVEL 4

From the Initial State (PKs 1 and 2 illuminated), pressing PK2 initiates a ROM load routine which searches through all present disks until it finds a valid bootstrap code.

The ROM search consists of two passes; the first looking for removable devices (1 Mb, 3 Mb and cartridge disks) and the second looking for fixed devices (201I and 211 disks). Each pass scans from the highest channel to the lowest and from the bottom drive to the top drive in each channel.

If either a channel or a drive fault is encountered, the ERROR light is lit and the inverse of the channel address displayed on (Bank 2) PK1 through PK8. That is, PK1 extinguished and PK2 through PK8 lit indicates a fault on channel 0, PK2 extinguished and PK1 and PK3 through PK8 lit indicates a fault on channel 1, and so on. This type of error occurs after a two-minute timeout, if the scan encounters a powered-off SDI device (3 Mb or 211 disk).

Pressing OCK1 causes the Caelus Primary or SDI Host Status to be displayed on PK1 through PK8 (Bank 2) and the Caelus Secondary Status or SDI Device Status Byte 1 to be displayed on PK17 through PK24 (Bank 4).

Pressing OCK2 causes the SDI Device Status Byte 2 to be displayed on PK1 through PK8 (Bank 2), and the SDI Device Status Byte 3 to be displayed on PK17 through PK24 (Bank 4).

Pressing the Reset key causes the scan to resume from the next drive in the case of a drive fault, or the next channel in the case of a channel fault. If both passes of the scan have been exhausted without a valid bootstrap being found, then PK1 through PK8 are illuminated. To re-attempt a Cold Start, it is necessary to press the Load Enable button, ensure that a disk containing a valid bootstrap is present on the system and is powered on, and repeat the above procedure.

If either a channel or a drive fault is encountered, the ERROR light is lit and the inverse of the channel address is displayed on D1 through D8. The Caelus Primary or SDI Host status is displayed on PK1 through PK8 and the Caelus Secondary or SDI Device Status Byte 1 is displayed on PK9 through PK16. Pressing the right hand Form Feed key causes the SDI Device Status Byte 2 to be displayed on PK1 through PK8 and the SDI Device Status Byte 3 to be displayed on PK9 through PK16. From this state, pressing the left hand Form Feed key causes the Caelus Primary and Secondary Status, or the SDI Host Status and Status Byte 1, to be displayed.

Pressing the Reset key causes the scan to resume at the beginning of the algorithm (that is, at the bottom of the highest channel).

ROM SCANNING ALGORITHM FOR LEVEL 5 ROMS

These ROMs are fitted to B 96 systems as standard and are available for B 94s. They may be identified by pressing OCK3 while in the initial state (PKs 1 and 2 illuminated). The version number is then shown on Bank 4 lights as ".....o.o" where "o" represents an LED on and "." represents an LED off. This is the binary representation of 5. The Level 5 ROMs permit the loading of the Coldstart Bootstrap (COLDBOOT) via PK1, or the Warmstart Bootstrap (WARMBOOT) via PK2. OCK3 displays the ROM version number.

If a hardware error is encountered while the ROM is searching for, or loading, a bootstrap, the ERROR light is lit and the channel number is displayed on PK1 through PK8 (Bank 2) by extinguishing the appropriate PK. For example, an error on channel 0 causes the ERROR light and PK2 through PK8 to be lit. Similarly, if an error occurs on channel 3, the ERROR Lights PK1, PK2 and PK4 through PK8 are lit. See "Hardware Error Conditions".

Loading Coldstart Bootstrap – PK1

From the initial state (PKs 1 and 2 illuminated), pressing PK1 initiates a ROM routine which clears the first page of memory (Page 0). It also initiates a load routine which searches through all present Burroughs minidisks and magnetic tapes until it finds a valid Coldstart Bootstrap code.

This is accomplished as follows.

Peripheral Channels are scanned to find a coldstart medium. If necessary, two scans are made. Each scan is from channel 7 to channel 0.

Scan 1:

The first scan looks for a Coldstart Disk, that is, a 3 MB BSMII or 1 MB BSMD minidisk which has been initialized, reformatted or has had its Bootstrap replaced with the Coldstart Bootstrap (COLDBOOT).

If a Coldstart disk is found, COLDBOOT is loaded and control is transferred to COLDBOOT. (See Coldstart Bootstrap Algorithm.)

If a Coldstart Disk is not found, a second scan is made.

Scan 2:

The second scan looks for a Coldstart Tape at load point, that is, a tape created by the LD utility, whose name is "CLDSTRT".

If a Coldstart Tape is found, the Coldstart Tape is searched for a file called "CMSBOOT". If the file is found, COLDBOOT is loaded and verified.

If the verification is successful, control is transferred to COLDBOOT. (Note the Coldstart Tape is not rewound.)

If a Coldstart Tape is not found, the system halts with the ERROR light and all Bank 2 lights lit.

Loading Warmstart Bootstrap – PK2

From the initial state (PKs 1 and 2 illuminated), pressing PK2 initiates a ROM Load routine which searches through all present disks until it finds a valid CMS Warmstart Bootstrap code.

This is accomplished as follows.

The peripheral channels are scanned for a Warmstart medium. If necessary, three scans are made. Each scan is from channel 7 to channel 0.

Scan 1:

The first scan looks for a removable Warmstart Disk, that is, a 3 Mb BSMII disk, 1 Mb BSMD or cartridge which has been initialized, reformatted or has had its Bootstrap replaced with the Warmstart Bootstrap (WARMBOOT).

If a removable Warmstart Disk is found, WARMBOOT is loaded and control is transferred to WARMBOOT. (See Warmstart Bootstrap Scanning Algorithm.)

If a removable Warmstart Disk is not found, a second scan is made.

Scan 2:

The second scan looks for a fixed Warmstart Disk, that is, a 211 or 201I fixed disk which has been initialized, reformatted, or has had its Bootstrap replaced with the Warmstart Bootstrap (WARMBOOT).

If a fixed Warmstart Disk is found, WARMBOOT is loaded and the control is transferred to WARMBOOT. (See Warmstart Bootstrap Scanning Algorithm.)

If a fixed Warmstart Disk is not found, a third scan is made.

Scan 3:

The third scan looks for a Coldstart Tape at load point, that is, a tape created using the LD utility whose tape name is "CLDSTRT".

1. A search for a file whose file name starts with the seven characters "CMSBOOT" is made. If the file is found, the WARMBOOT portion of the CMSBOOT file is loaded from the tape and its checkstring is verified.
2. If the checkstring is correct, control is transferred to WARMBOOT. (The tape is not rewound.) If the checkstring is incorrect, the scan is resumed.

NOTE

WARMBOOT itself does not have the capability of searching for MCPX or SAU on magnetic tape. So, if no disk containing these files is ready, only Dump to Tape (PK4) and/or Dump to Disk (PK5) can be carried out. A complete Warmstart, that is, loading MCPX or SAU, can be carried out only if a disk containing these files is ready.

3. If no Coldstart Tape is found, the system halts with the ERROR light and all Bank 2 lights lit.

ROM SCANNING ALGORITHM FOR LEVEL 6 ROMS

The level 6 SYSTEM/MTR was designed for use in the B 95 system. The ROM includes numerous improvements over versions produced previously for use with other B 90 series systems. The B 95 is a totally data comm based system, so the ROM interface with this has now been enhanced. The enhancements to the ROM also allow the use of peripheral types introduced for the B 95 system. The ROM is split into two distinct 4 K byte sections. One half is the code to perform the "Power On" confidence routines. The other half contains the code to search for and load a required bootstrap. The bootstrap micro-code, once loaded, executes the required function requested by the operator. The system part of the ROM has the ability to branch back into the MTR routines if, at any time, it detects a possible hardware fault.

The level 6 SYSTEM/MTR ROM requires a second ROM to control the data comm input and output. This ROM also contains all the messages displayed by the firmware based routines or the bootstrap code.

The level 6 SYSTEM/MTR ROM must be used in conjunction with CMSBOOT version 3.04.59 or greater. This means that any disk from which the ROM is to obtain its bootstrap code must have the correct level of bootstrap in the sectors allocated for bootstrap information.

Data Communications Requirements

The level 6 SYSTEM/MTR ROM has the capability to display its current status or requirements to the operator through a terminal device. There are certain conditions which must be met if a terminal device is to be used to communicate with the ROM.

The system must have a Data Communications Power Pak PCB installed in processor channel 5. This may be one half of a Dual Data Communications Power Pak PCB if one is available. The interconnection to the terminal must be asynchronous Two Wire Direct Interface (TDI) type. The terminal must be a TD 830 look-alike device which has been qualified for use with B 90 series systems.

Suggested terminal styles and minimum firmware levels:

Style	Firmware Level
TD 830	4.00
TD 830J	4.00
MT 983	3.08
MT 985	3.08
MT 993	3.08
ET 1100	5.00

NOTE

The use of terminals which have a firmware level lower than those suggested, is not totally prohibited. The suggested levels represent the levels of firmware with which the ROM was tested.

The internal switch settings of the terminal must be set to allow for TDI operation. In addition, the terminal must have enough video RAM available to allow two page operation. The address registers of the terminal require to be set to @01 (hex 30 31). The ROM will attempt to communicate with the user through a suitably configured terminal at one of five possible line speeds. The line speeds searched through are 38400, 19200, 9600, 4800 and 1200 baud, scanned in that order. Failure to locate a ready terminal after all the possible line speeds have been checked, causes the ROM to enable keyboard input only. If the initialize button is pressed, a scan will again be made to find a usable terminal.

Keyboard Input and Output

If a suitable terminal is not available, it is possible to initiate the loading of software from the MTR keypad. Any error conditions detected will be displayed on indicator banks located above the keypad keys.

ROM Start Up Cycle

After the system is powered on, the ROM performs routine confidence checks to verify that the hardware is fully functional. Once it has been ascertained that the system is functional, the firmware enters SYSTEM mode, allowing software to be loaded into memory. The following provides an overview of the ROM controlled start up cycle.

Number	Description of Action Performed
1	I/O Confidence Test All LEDs on the keyboard are set on, LEDs 7 to 0 on the FAULT CODE bank are then set off sequentially, followed by LEDs 7 to 0 on the START CODE bank. If a failure is detected, the whole test will be repeated continually until the system is powered off or initialized.
2	Processor Confidence Test After the processor confidence routine is performed, if no errors are detected, the system can now begin to display its progress to the operator. From this point onward, it has been presumed that a terminal was found to be available. After successfully passing the processor confidence test, the message "System Confidence Test" is displayed. This message appears on page 2 of the terminal and all further confidence test output is scrolled upwards.
3	Memory Confidence Test The message "Starting Memory Test .....Please wait " displayed on the terminal indicates that the memory test has begun. The message "Memory Test completed" indicates the end of the memory test. If an error is found on memory page zero, a message indicates this and an attempt is made to locate error free memory. On final completion of the memory test, the operator is prompted to insert an MTR disk.
4	Timer Confidence Test No indication is provided to inform the user that the internal timer test has started. If no failure is detected, the message "Internal timer test complete" is output and the test sequence continues.
5	Winchester Disk Controller Confidence Test Tests are made to all drives attached to each Winchester controller. If no failures are detected on a disk controller, the confidence routine moves on to the next stage.
6	Check Floppy Disk Drive for MTR Disk After successfully completing the preceding confidence tests, the ROM scans the floppy disk drives on the system in search of an F. E. bootstrap. If no such bootstrap is detected, the message "CONFIDENCE CHECKS SUCCESSFULLY COMPLETED" is displayed and the terminal returns to page 1. This indicates that the MTR carried out by system ROM has not detected any failures. If any failures are notified by the MTR routines, the user is prompted to insert an MTR disk and run the full disk based confidence routines.
7	Enter System Mode This is the point where a user is prompted to supply software loading instructions to the firmware. A menu screen is sent to the user's terminal if one is available and

(continued)

Number	Description of Action Performed
	the LEDs above keys 0, 4, 5 and 6 are set on indicating they are enabled. The option selection can be made either from the user's terminal or by the enabled keys on the systems keypad. The options available to the user are as follows: 4. Start normal operation.....(WARM START) 5. Save contents of memory for analysis ... (DUMP MEM) 6. Start independent utility program..... (STAND ALONE) 0. Copy software from removable media ....(COLD START) The numbers used on the menu screen correspond to the number of the key which initiates the function.
8	Return from Selected Option Once the operator has finished using the required function and terminates its use, the system performs the I/O confidence test, checks for an MTR disk and returns to System Mode. This may not be the case if a disk with F. E. bootstrap is found to be in the floppy disk drive. In this case the full ROM based MTR cycle will be executed once more.

COLDSTART BOOTSTRAP ALGORITHM

When the Coldstart Bootstrap (COLDBOOT) has been loaded successfully, COLDBOOT tries to load the utility called COLDSTART from the device from which COLDBOOT was loaded.

If the utility COLDSTART is not found, the ERROR light is lit and PK1 through PK8 are extinguished.

Pressing the RESET key after COLDSTART not found on hardware error has no effect – that is it continues to light ERROR and display channel address, and OCK options still remain.

To try again, user must mount COLDSTART medium and re-initialize machine.

If the error persists, seek technical assistance.

WARMSTART SCANNING ALGORITHM

The successful loading of CMS bootstrap (WARMBOOT) causes PK3 through PK6 to be illuminated. Also, OCK4 is enabled to display the WARMBOOT version number in a two part process (see Hardware Error Conditions for format). Note PK1 through PK8 are on bank 2.

When PK3 (execute MCPX), PK5 (memory dump to disk) or PK6 (execute SAU) is pressed, a two pass scan algorithm is invoked. This is similar to the first two scans described in "Loading Warmstart Bootstrap", but with the following differences.

To indicate that a channel or drive fault is encountered by the bootstrap, the bootstrap displays the channel address on PK1 through PK8 by illuminating the channel PK and extinguishing the rest. That is, a fault on channel 0 is indicated by PK1 being lit and PK2 through PK8 being extinguished. A fault on channel 1 is indicated by PK2 being lit and PK1 and PK3 through PK8 being extinguished and so on.

Similarly, when both passes have failed to find the desired file, PK1 through PK8 are extinguished and only the ERROR light is illuminated.

The time-out incurred by the bootstrap encountering a powered off SDI device is 70 seconds.

When a channel or drive fault is encountered, pressing the RESET key causes the bootstrap to continue its scan at the next lowest drive or channel.

MEMORY DUMP TO MAGNETIC TAPE

This only applies to systems with ROM level 5.

The system must be in the CMS Warmstart Bootstrap state, with PK3 to PK6 lit.

NOTE
The release level of the bootstrap must be 3.04 or higher.

If the system is in the initial state, press PK2 to load a Warmstart Bootstrap.

Insert a magnetic tape with a write ring into the drive and press the LOAD and ONLINE buttons on the tape streamer.

Press PK4.

The contents of RAM are written to the magnetic tape. During the dump, an indication of the memory address being dumped is displayed on bank 4. When the dump is complete, the system returns to the Bootstrap state (PK3 to PK6 lit). The magnetic tape is labelled "MEMDUMP/MEMORY".

Press the ONLINE and then the UNLOAD buttons on the tape streamer, remove the magnetic tape and remove the write ring from the tape. Clearly mark the tape with the date and time and submit it with details of the fault to your support personnel.

For possible errors see "System Load Errors".

NOTE
If the Warmstart bootstrap (WARMBOOT) was loaded from magnetic tape and this tape is to be used for the dump, the tape must be rewound to the load point. This is because the ROM leaves the magnetic tape past the load point to prevent accidental writing to the magnetic tape.

MEMORY DUMP TO DISK

Before memory dump to disk can be initiated, the following conditions must be met:

If the system is configured with a console/MTR keyboard, the CMS bootstrap WARMBOOT must have been loaded and entered. This is indicated by PK lights 3, 4, 5 and 6 only being lit.

To initiate the dumping of memory to disk, press PK5.

If the system is configured with an ELPRO keypad, the system must have been initialized and status lights 0, 4, 5 and 6 only being lit.

To initiate memory dump to disk, press PK5.

Memory dump to disk continues as follows:

1. The system searches for a file named MEMDUMP on a removable disk.

   1) Higher numbered channels are scanned before lower numbered channels; higher numbered drives are scanned before lower numbered drives.

   2) If a MEMDUMP file exists on a removable disk, the system will check that the second digit of byte 44 of the Disk File Header contains 0; otherwise it will not dump memory to the disk. This check ensures that either:
      a. memory has not been dumped to the disk since the MEMDUMP file was created by the utility GEN.DUMPFL, or
      b. since memory was last dumped to the disk, it has been submitted to PMB90 for analysis.

   3) The system will begin dumping memory from location #1000 to the first removable disk found to contain a valid, single-area MEMDUMP file.

   4) If the MEMDUMP file is not large enough to contain the memory to be dumped, the system will dump as much memory to the file as it can hold and then wait for user action, displaying #AA (on Bank 2 lights of a console/MTR keyboard, on status lights of an ELPRO keypad). The user must load another MEMDUMP removable disk (if additional disks have not already been loaded) and press the console/MTR RESET key or the ELPRO key 7. The system will then resume dumping memory to the removable disks from the location following the last area dumped.

   NOTES:
   – As many disks as are required can be loaded.
   – Cartridges and floppy disks are both considered removable disks for dumping purposes.
   – Once dumping to a removable disk has begun, dumping to a fixed disk is inhibited. Consequently, if it is desired to dump memory to a fixed disk, ensure that no removable disks containing MEMDUMP files have been loaded. If it is desired to dump to removable disks, ensure that sufficient MEMDUMP space exists on the removable disk (or disks).

2. If the system cannot find a valid MEMDUMP file on a removable disk, it searches for a MEMDUMP file on a fixed disk.

   1) Highter numbered channels are scanned before lower numbered channels; higher numbered drives are scanned before lower numbered drives.

   2) No check is made on the Disk File Headers of MEMDUMP files resident on fixed disks.

3) The system will begin dumping memory from location #1000. The MEMDUMP file must be large enough to contain all of remaining memory.

3. After all of memory has been successfully dumped to disk, the system waits, and displays on the console/MTR PK lights 3, 4, 5 and 6 = warmstart options, on the ELPRO status lights 0, 4, 5 and 6 = start options.

SYSTEM LOAD ERRORS

The following table of errors covers all symptoms found during start-up of any B 90 except the B 95, warmstart, memory dump to disk or cassette, or entry to Stand-Alone utilities.

Symptom	Possible Causes	Suggested Action
Sequential lighting and extinguishing of lights does not occur on entry to Initial State (for example, depressing Load Enable button).	MTR switch in wrong position.	Set MTR switch to "normal". Depress Load Enable button.
From Initial State, PK2 is ignored.	Keyboard locked in "shift" mode.	Depress shift key. Depress PK2.
Depression of PK2 causes numeric light to be lit.	PK1 was depressed by mistake (this clears memory).	Depress Load Enable. Depress PK2.
Depression of PK2 causes keyboard lights plus ERROR light to be lit:		
(a) PK1, PK8 (Bank 2) lit (b) One of PK1-PK8 (Bank 2) extinguished, all other (Bank 2) lit	No bootstrap found (disk not initialized correctly). Disk media or drive fault.	Press RESET to continue scan. Check on disk used. Reload with correct disk. Press Load Enable. Press PK2. See below for diagnosis. Note PK lights. Check that disk is not at fault by using disk in another drive. Power off faulty disk, replace with backup copy. Request technical assistance.
All keyboard indicators flash.	Memory parity.	Depress Load Enable, depress PK1 (to clear memory), then depress Load Enable again, and retry. Request technical assistance if not successful.
Print head does not initialize at start of warmstart.	Console printer faulty.	Open and close cover. Repeat warmstart. Request technical assistance.
Memory dump to disk: Depression of PK5 causes ERROR lit, PK1-PK8 (Bank 2) extinguished.	No MEMDUMP file found on disk.	Load a disk with a MEMDUMP file and depress PK2 then PK5 to retry the dump.
Memory dump to disk: ERROR light is lit during dump. One lit. Bank 2 (PK1-PK8) light lit.	MEMDUMP file on disk is too small to hold memory contents for this machine.	Load disk with larger MEMDUMP file; depress PK2 then PK5 to retry the dump.
Memory dump to disk: system warmstarts.	PK3 was depressed instead of PK5.	Memory dump cannot now be taken. Allow warmstart to complete.

(continued)

Symptom	Possible Causes	Suggested Action
Entry to S.A.U: Depression of PK6 causes one D light, plus ERROR, to be lit.	SAU could not be loaded from disk: disk error.	Note D and PK lights. Remove disk, replace with backup copy. Press Load Enable, PK2 and PK5. See below for diagnosis.
Entry to S.A.U: load stops with all keyboard lights lit.	Console printer not ready, or datacomm terminal not in receive mode.	Check forms transport closed. Check printer cover down. Check levers in correct position. Utility should continue. Depress RCV on datacomm terminal with lowest address.

Diagnosis of Disk Errors at System Load Time

If an error occurs during the previously described scanning of the channels by the ROM, the system will illuminate the error light and the number will be displayed on Bank 2 (PK1-PK8) lights. That means that the light on Bank 2 which is not lit indicates the channel on which the error occurred, where the channels have to be counted from 0 to 7 on the lights from left to right.

At this moment the machine will wait for keyboard input. The following keys are enabled.

OCK1: to display
- BANK 2 (PK1-PK8) – Primary status
- BANK 4 (PK17-PK24) – Secondary status of a removable disk, device status 1 of BSMDII or 211 fixed disk, or data transfer of magnetic tape.

OCK2: to display
- BANK 2 (PK1-PK8) – Device status 2 of BSMDII or 211 fixed disk, or transport status of magnetic tape.
- BANK 4 (PK17-PK24) – Device status 3 of BSMDII or 211 fixed disk or dynamic status of magnetic tape.

OCK3: to display
- BANK 4 (PK17-PK24) – contents of the ROM identity byte.

RESET – to continue the channel scan at the next lower-numbered channel/drive.

NOTE

All status displays are bit-reversed.

NOTE

This ROM scanning and error indication is not applicable on the B 95.

Caelus Device Primary Status

Bit	Interpretation
0	Drive Addressed (0 = Upper, 1 = Lower)
1	0 = Seek complete
2	0 = End of cylinder
3	0 = Search complete
4	0 = Secondary status condition
5	1 = Operational
6	0 = Seek complete
7	1 = Status OK

Caelus Disk Secondary Status

Bit	Interpretation
0	Ignored
1	1 = on cylinder
2	0 = Illegal seek
3	0 = Write inhibit
4	0 = Sector not found
5	0 = LRC error (parity)
6	0 = Illegal command sequence
7	0 = Device error

SDI Host Controller Status

Bit	Interpretation
0 1 2	Apart from MTR use, these bits should be set to 001, indicating that the Host Controller is ready to accept a Device Controller command or 1st segment data from Host.
3	1 = Device Controller Status available
4	1 = Buffer available
5	1 = Buffer not empty
6	1 = Interface timeout (hardware fault)
7	1 = Interface retry exceeded (hardware fault)

SDI Device Status

Bit	Byte 1	Byte 2	Byte 3
0	Drive No. Bit 0 (LS)	1 = Error	1 = Not Ready
1	Drive No. Bit 1	1 = Search Unsuccessful	1 = Disk Expiring
2	Drive No. Bit 2 (MS)	1 = Corrected Data	1 = Write Protected
3	1 = Transfer Delay	1 = Command not Accepted	1 = New Disk
4	1 = N Sectors Before Read	1 = Command Error	1 = Danger
5	1 = N Sectors Before Write	1 = Address Error/End of Drive	1 = Confidence Test Completed
6	Operation Complete	1 = Mandatory Int. to Host	1 = Temporarily Unavailable
7	Interrupt	1 = Address not Found	Unassigned

Magnetic Tape Device

Primary Status

Bit	Interpretation
0	1 = Operation complete
1	1 = Data transfer exception (reason in secondary status byte 1)
2	1 = Transport state condition (reason in secondary status byte 2)
3	1 = Tape mark found
*4	1 = Record bigger than expected
*5	1 = Record smaller than expected
6	1 = End of record
7	1 = Request Data Transfer (Buffer empty if write, buffer full if read)

Secondary Status

Bit	Byte 1	Byte 2	Byte 3
0	Tape corrected error	Transport address *	1 = Formatter busy *
1	Tape hard error	1 = Transport address *	1 = Data busy * 1 = high speed 0 = low speed
2	Parity error from formatter	1 = Write Inhibited tape	1 = high speed 0 = low speed
3	Controller detected error	1 = Drive online	Tape streaming *
4	Service too late	1 = Drive Ready **	not used
5	1 = Gap time out	1 = Tape past EOT	not used
6	not used	1 = Tape at load point *	not used
7	not used	1 = PE tape ID found at BOT *	1 = Rewinding *

* Does not result in request to host
** Required only on change in transport status

ERRORS UNDER MCP CONTROL

The following table of errors covers many symptoms found while using the B 90 under normal control of the MCPX, with suggested causes and actions to take.

Symptom	Possible Causes	Suggested Action
Unidentifiable problem	MCP file corrupted	Use backup copy of MCP
Little happening, with D2 on, D3 off most of time	System thrashing; too many jobs or too large jobs in mix.	If console not in use, set GT ON. If light pattern above PK17 to PK20 is: on, on, off, on, then thrashing is confirmed : D lights 4 to 7 indicate mix number of executing task. Try entering MX to clear condition : if not successful, take memory dump and do clear start : request technical assistance.
PK lights 17 to 24 flashing with "DF" pattern (on,on,off,on, on,on,on,on)	Hardware disk I/O error during MCP task.	Take memory dump: request technical assistance to analyze memory dump to find disk address for use in fixing the disk. To help find the fault, warmstart and run the KA utility on the system disk before executing any programs. (See KA, section 4). If shows any "AREAS ASSIGNED TWICE" the disk directory may be bad. DO NOT USE THIS DISK until recovery operations have been completed.
PK lights 17 to 24 flashing "AC" pattern (on,off,on,off,on, on,off,off) or "AD" pattern (on,off,on,off,on,on,off,on)	See below:	(Take recovery action as for "DF" messages above: DO NOT USE THE DISK until recovery action is completed.)
	hardware disk I/O error during MCP disk directory operation.	Request technical assistance to fix the disk.
	Logical error in disk directory due to previous fault.	Rerun using backup disk : analyze corrupted disk using S.A.U. PDX function and report.
Non-zero retry count on PO of system disk	Deteriorating disk performance.	Ignore if 10 or 20 during normal running in one day. Call field engineer if greater rate than usual rate for the site.
	Improper care of removable disks: possibly failure to re-initialize disks after long use.	All frequently used removable disks should be cleaned and re-initialized regularly.

(continued)

Symptom	Possible Causes	Suggested Action
Logging not taking place.	SYS-SUPERUTL not on systems disk.	Load SYS-SUPERUTL to systems disk.
	TL not on systems disk.	Load TL to systems disk.
	Logging not specified in SYSCONFIG file.	Rerun CONFIGURER and warmstart.
PD, RM function not available.	No SYS-SUPERUTL	Load SYS-SUPERUTL to systems disk.
	SYS-SUPERUTL already performing one of its functions (see discussion in section 4).	Check with MX if SYS-SUPERUTL is performing a function (mix-number 12 gives the name of the function): wait then re-enter command.
Cannot run data comm.	No NDLSYS file on system disk.	Load NDLSYS file (you may need to run an NDL compilation).
	NDLSYS file does not correspond to actual network on machine.	Load correct NDLSYS file; or recompile NDL program.
	NDL.INTERPX not present on system disk.	Make sure the correct interpreter (for example MICROPOL) is named NDL.INTERPX and located on the system disk.
	No MCS program.	Ensure that the correct MCS is executing as well as any user data comm program.
System not active: programs in mix but D-light pattern does not change; keyboard input (for example, MX) is possible.	Program swapped out waiting on memory.	Enter <mix> GO to resume program execution; if symptom persists, program is too large for available memory.
System not active : programs in mix but D-light pattern does not change; keyboard input (for example, MX) causes error bell to ring, or remote SPO mx is only scrolled to the bottom line.	Console file input in process.	Complete input using an OCK key.
	System printer is jammed.	Open and close printer cover to initialize print head. Repeat if not successful first time. If still not successful, perform a warmstart.
	Peripheral (possibly line printer) hang.	Enter CL command. If not successful, make peripheral ready or not ready; switch it off or

Symptom	(continued) Possible Causes	Suggested Action
		on; insert and unload spare cassette in cassette drive. If this fails, take memory dump, report fault and warmstart. Note: never load or unload disk media, or cassette media which are in use, until system has returned to initial state.
	MCP software error.	Take memory dump, report fault and warmstart.
System initializes (PK1 and PK2 lit)	Corrupted MCP code file.	Take memory dump, report error, rerun on backup disk.
	Unexpected hardware error.	Take memory dump, report error, re-warmstart.
	Unknown MCP error.	Take memory dump, report error, attempt to rerun. If still not successful, request technical assistance.
PK lights 17 to 24 flashing (pattern other than above)	MCP has detected an error condition from which it cannot recover.	Refer to section on MCP diagnostic messages for further details.
All keyboard lights flash.	Memory parity.	Initialize system, and rerun. If still faulty, request technical assistance.
Program load taking long time : keyboard disabled and D lights 4 to 7 flashing.	Program loaded thrashing due to lack of available memory; too many or too large programs in mix.	Try keyboard input. If this fails, initialize system and retry.
Hissing sound from disk drive.	Disk drive heads are damaged.	Stop work. Remove the disk from the damaged drive. THIS DISK IS PROBABLY DAMAGED, DO NOT LOAD ANY OTHER DISK INTO THE DRIVE. Request technical assistance.
Non-functional disk drive.	Hardware fault on drive.	Load a good disk on drive (UNLESS IF DRIVE HEADS HAVE CRASHED), run SAU LS utility. If this produces errors, the drive is faulty: call field engineer.
I/O errors from disk.	Non-function disk drive.	See above.
	Crashed disk head.	See above.
	Physically damaged disk media.	Replace with new disk media.

(continued)

Symptom	Possible Causes	Suggested Action
	Disk media has bad areas.	Run CHECK.DISK to report on parity errors. Run KA to give disk directory analysis: if this shows AREA MISSING or AREA ASSIGNED TWICE, run SQ <diskname> VERIFY to recover the area which is missing or which has been assigned twice. Use backup disk for system files. Re-cover any other files by attempting to COPY them one-by-one from the corrupt disk to a good disk.

ERRORS DURING WARMSTART OF A CONSOLELESS SYSTEM

To warmstart a consoleless system (for example, B 93, B 94, B 95 and B 96), the following files must reside on the system disk:

MCPX
BILINTERPX
SYSLANGUAGE
SYSCONFIG
NDLSYS
NDL.INTERPX
MCSOBJ
MCSWARM } These files may have other names
MCSIN

If one or more of these files is missing or disturbed then the system will not start. In that case an error code is displayed on the keyboard lights. Table 8-1 gives an explanation of these errors. In the light pattern a 0 indicates the light is off, an X indicates the light is on.

Table 8-1. Consoleless System Error Codes

Light Pattern	Condition	Suggested Action
0000 000X	NO SYSCONFIG	The system disk must contain a SYSCONFIG file. Use SAU COPY to copy a good SYSCONFIG file to system disk, then restart the system.
0000 00X0	INVALID MCS ZIP TEXT	The zip-text in the SYSCONFIG file is invalid. Use SAU COPY to copy a good version of SYSCONFIG to the system disk.
0000 00XX	PROGRAM NOT FOUND	MCS named in SYSCONFIG file is not found – either zip text in SYSCONFIG incorrect or MCS not on disk. Use SAU COPY to copy a good MCS to disk or use SAU CH to change the name of the MCS to the name requested by the SYSCONFIG file.
0000 0X00	INTERPRETER NOT FOUND	BILINTERPX or NDL.INTERPX not on disk or MCS code file corrupt. Use SAU COPY to copy these three files to disk.
0000 0X0X	NO USER DISK	Loading of the MCS or NDLSYS has failed due to lack of disk space. Use SAU RM to remove unneeded files from disk and start again.
0000 0XX0	MCS ALREADY PRESENT	An attempt was made to start a second MCS. If the current MCS supports remote SPO, the system is unaffected, else use SAU CH to change the name of the remote spo capable MCS to the name of the MCS first started and restart the system.
0000 0XXX	DISK ERROR	There is a disk parity error on one of the files to be loaded. Save all the files from the system disk using SAU COPY then re-initialize the disk and load all files back to

Table 8-1. Consoleless System Error Codes
(continued)

Light Pattern	Condition	Suggested Action
		system disk. Then warmstart the system. In case the system has only tape as a backup system, the system has to be coldstarted using the COLDSTART utility with the initialize option.
0000 X000	CODE FILE ERROR	The MCS zipped is corrupted. Use SAU COPY to copy a good MCS to system disk.
0000 X00X	DC LOAD ERROR	Load of datacomm failed because the NDLSYS file is not suitable for this system. Use SAU COPY to copy a suitable NDLSYS, NDL.INTERPX combination to system disk.
0000 X0X0	SPIM WENT DS/DP	The NDLSYS, NDL.INTERPX combination may not support SPIM. Use SAU COPY to copy a suitable NDLSYS, NDL.INTERPX combination to disk or copy an NDLSYS, NDL.INTERPX combination which supports ETS (Enhanced Terminal Spo), in which case SPIM is not needed.
0000 X0XX	MCS WENT DS/DP	The MCS went DS/DP. On system disk a dumpfile is created called DMFILxx, where xx is the mix number of the MCS. Normally it will be DMFIL01. Restart the system and ask for technical assistance to analyse the dumpfile. Keep a copy of the spo-log. If the MCS goes DS/DP again during warmstart, use SAU COPY to copy a good version of the MCS to disk.
0000 XX00	NO SYSLANGUAGE	The light pattern will be flashing. It indicates that there is no SYSLANGUAGE file on disk, or that the MCP could not read it. Use SAU COPY to copy a good version of SYSLANGUAGE to the system disk.

If any other error occurs during warmstart, request technical assistance.

B 90 DEPENDENT UTILITIES

The following pages describe those system utilities which run under MCP control but which are relevant only to the B 90 CMS software.

BUILDTRANS (Build Translation Table)

CMS provides translation tables for all versions of the keyboard, screen, CRT, line printer and serial printer currently supported by the MCP. The user has the capability to specify a non-standard translation table for any of these devices, by setting up SYSCONFIG (see CONFIGURER and SYSTRANS for naming conventions) appropriately. At warmstart the MCP uses SYSCONFIG to select the desired translator table(s) from the file SYSTRANS. See SYSTRANS FORMAT for details of naming and format.

The translation of devices is achieved by:

1. Modification of existing translation files.
2. Creation, and subsequent modification, of new translation files consisting

    1) entirely of newly defined tables.
    2) of translation tables acquired from existing files.
    3) of a combination of 1) and 2).

Limitations

Buildtrans will allow the creation of table.names of any format. However, when the table is to be used by the LT command, the table.name should have the format:
<device>.<variant>.<country/language>.V<version.no>

See SYSTRANS Format later in this section for details.

There is a limit of 200 tables and 400 directory entries for a new translation file.

For an old file there is a limit of twice the number of tables present in the file, or 200, (which ever is the greater), and twice the number of directory entries in the file, or 400 (which ever is the greater).

Input limitations:

For a screen device, this limit will be 80 characters. For other devices, a maximum of 120 characters can be entered.

Only MODE 0 and MODE 1 translation tables can be used on the B 90.

MODE 1 Translation tables for input translation must be used with a keyboard device which has a mode key or equivalent.

The only supported devices on a B 90 for translation are:

B 91 and B 92 Keyboards
CRTs
B 91 and B 92 serial printers
Line printers

This is true only if the hardware device is supported for the B 90 in question.

Operating Instructions

The BUILDTRANS utility enables the user to perform the required operations to construct/modify translation tables for devices.
There are three types or modes of translation tables.

Mode 0

This is the simplest translation table, where there is one entry for each code in the character set. The table size is 256 bytes.

Mode 1

In this mode each internal code has n number of output codes. Each internal code has one entry, n bytes in length, where n = length of each entry (the length is fixed for all entries). This mode is used for devices such as current CMS Arabic line printers, where each internal code is translated to either 1 or n characters. Unused print codes for each entry are set to hexadecimal FF.

Mode 2 (not implemented on B 90)

This is a completely general table which allows code Ci, where i is greater than or equal to 0 and less than or equal to 255 ($0 \leq i \leq 255$), to have j output forms, where j is greater than or equal to 0 and less than or equal to 255 ($0 \leq j \leq 255$). If j = 0 then no translation will take place. This mode allows complete contextual analysis of the language being translated.

Shift Codes

Each translation table requires shift codes. This allows the translation table to be used with a 7 bit device, (that is, the device can only recognize 128 codes, compared with 256 for an 8 bit device). The format of the table using shift codes is two 128 entries making a 256 entry table.

The value of @0000@ hexadecimal should be entered for devices which use an 8 bit interface. For devices which use a 7 bit interface, the 1 byte shift out code followed by the 1 byte shift in code, for the device, should be entered as a 4 digit hexadecimal number.

Workfile Creation

The utility operates on a workfile which can be created in one of three ways. If an existing translation file is to form the basis of the new file, a GET of that file can be done. If no existing file is sufficiently suitable, an empty workfile can be created by doing a MAKE. In addition, it may be possible to RECOVER a previously created workfile.

When a workfile is available, operations can be performed on it.

For example :

1. A table from an existing translation file can be INCLUDEd in the workfile, providing that a table of the same name does not already exist in it.
2. A new user table can be built using the BUILD function.
3. A user table can be DELETEd.
4. A user table can be altered using EDIT.
5. Tables and/or directory entries can be LISTed.
6. When the workfile is complete, a new translation file can be created from the workfile using the SAVE option.
7. If the workfile is no longer required, it may be REMOVEd.

A subsequent GET, MAKE or BYE is only permitted if the workfile has been SAVEd or REMOVEd, or no modification has been performed on it.

Initiation and Working Mode

The initiation of the BUILDTRANS utility and many of the functions used are similar to the operation of CMSCANDE.

If a data comm System is present, the program can be executed from a TD 800-type terminal device. However, if data comm is not present, or access is denied by the MCS, the utility attempts to use console I/O. If this fails, then Displays and Accepts will be used. This same procedure is used if data comm was being used but the MCS is terminated during the execution.

If a data comm failure is detected during program execution, a message is sent to the SPO, giving the error status and a choice of options :

ABORT removes the work files, closes the input file (if present) and goes to End Of Job.
OK returns control to the user and ignores the offending message.
SPO directs Output/Input to the SPO via a Display/Accept
GO attempts to transmit the message again.

Functions

The functions used by BUILDTRANS utility are similar to those of CMSCANDE. The available functions are:

GET, MAKE, INCLUDE, LIST, BUILD, EDIT, DELETE, LET, SAVE, REMOVE, RECOVER and BYE.

GET

(For example: GET SYSTRANS)

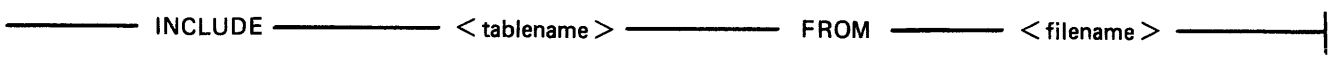
GET causes a workfile to be generated from the file supplied. It is only permitted after a SAVE, REMOVE or a previous GET/MAKE when no subsequent alteration(s) have been performed.

MAKE

(For example: MAKE MYSTRANSFILE)

MAKE causes an empty workfile to be generated. The same restrictions as for the GET function apply.

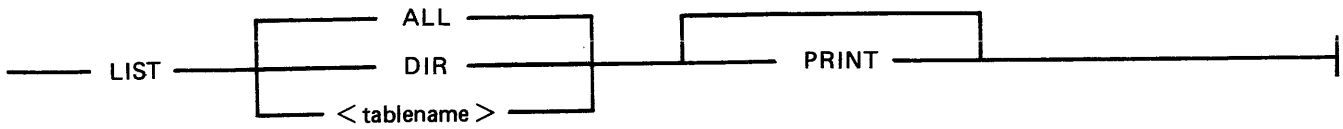
INCLUDE



(For example: INCLUDE LP.B924930.MYTAB.V1 FROM OTHERFILE)

INCLUDE copies an existing table from a named translation file to the current workfile, but only if a table of the same name does not already exist in the workfile. INCLUDE is only permitted if a workfile is already present.

LIST



(For example: LIST LP.B924930.MYTAB.V1 PRINT)

LIST outputs information from :

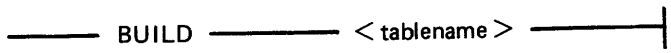
1. the directory of the workfile, or
2. a translation table, or tables, contained in the workfile, or
3. a combination of 1 and 2.

The output will be directed to the device from which the function was requested. However, by specifying PRINT, output will be directed to a hard copy device.

If data comm is being used, and the PRINT option was NOT selected, then the function can be terminated prematurely by entering "BRK". The utility then returns to the state existing prior to receipt of the LIST request.

If output is to a screen device, then BUILDTRANS gives the user an option to end the list or to continue after each screenful. In addition, if the entire file is being output, then this option applies after each complete table has been output.

BUILD



(For example: BUILD LP.924930.MTTAB.V2)

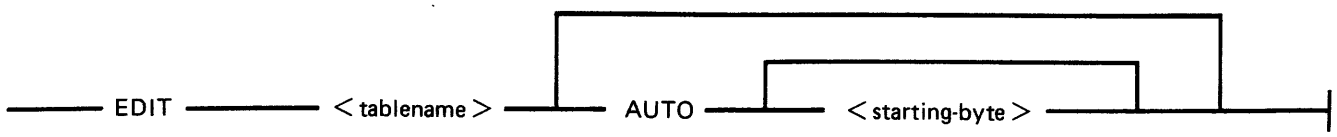
BUILD allows a new user table to be defined, providing that a table of that name does not already exist in the current workfile.

Details about the translation mode and, if using mode 1, the length of the translation elements, are requested.

An implementation limit of 10 bytes per byte to be translated will be imposed. The utility outputs the hex value to be translated, and the user enters the required value(s). A null response causes hex FF to be inserted in the appropriate position(s) in the table for modes 0 and 1, and a length of zero for mode 2.

Entering END causes the rest of the table to be initialized to the appropriate values depending on the mode; that is mode 0 and 1 will have hex FF written to the remaining positions. Mode 2 causes a length of zero to be inserted in the index table for each remaining item.

EDIT



(For example: EDIT LP.B924930.MYTAB.V1 AUTO 20)

EDIT performs alterations on any user table in the workfile. By specifying AUTO, BUILDTRANS outputs the hex value to be translated, starting at the value supplied, or at zero if no value is specified. BUILDTRANS then expects the new table value to be entered. On accepting this value, the next value to be translated is output, or EDIT terminates if the last value output was FF. A null reply causes the default value to be inserted.

NEXT causes the current entry to be retained and the next value to be translated will be output. If the AUTO option is not selected, then two hex numbers must be entered, separated by at least one space, after BUILDTRANS has issued a prompt. The first number is taken as the byte to be translated, and must be in the range 0 to FF. The second number is taken as the table value and must be of the correct length. A utility limit of up to 20 hexadecimal characters for this second value is imposed.

Entering END exits from this mode and the rest of the table is left untouched.

DELETE

(For example: DELETE LP.B924930.MYTAB.V2)

DELETE removes the specified user table, if it exists, from the workfile.

NOTE

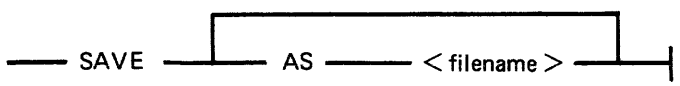
It is not possible to delete a table-name pointing to to a system table.

LET

(For example: LET LP.B924930.MYTAB.V3 EQUAL LP.B924930.MYTAB.V1)

LET allows a new directory name to be generated which points to an already existing table. The only attributes which can be changed are the shift codes and the machine restriction.

SAVE



(For example : SAVE AS MY TRANSFILE

SAVE causes a new translation table to be produced from the current workfile. The default file name is that associated with the current workfile from an earlier MAKE or GET. The workfile can be saved under a different name using the "AS <filename>" option.

REMOVE

(For example: REMOVE)

REMOVE removes the current workfile, if there is one, and returns the utility to the state which existed prior to a GET or MAKE. It is also applicable if a workfile is found, by BUILDTRANS, at BOJ.

RECOVER

(For example: RECOVER)

RECOVER is only permitted at BOJ if BUILDTRANS discovers a file which appears to be an old workfile. It attempts to restore the file to the state which existed prior to the abnormal termination.

BYE

(For example: BYE)

BYE causes BUILDTRANS to go to EOJ.

BUILDTRANS Message Format

Message	Possible Cause	Suggested Action
<filename> IS NOT A VALID TRANSLATION FILE	The specified file does not have the filetype of a translation file.	Check input for correct filename.
-BREAK COMMAND RECEIVED	Normal output after entering a BRK command.	None.
-<option> IS NOT A VALID OPTION	Self-explanatory.	Correct input and re-enter.
-OPTION HAS ALREADY BEEN SPECIFIED	Self-explanatory.	None.
THE WORKFILE FOR BUILDTRANS CAN BE RECOVERED, IF REQUIRED	The utility has found a workfile on disk that can be recovered.	Use RECOVER or REMOVE command.

(continued)

Message	Possible Cause	Suggested Action
-PLEASE ISSUE A RECOVER OR REMOVE REQUEST	Self-explanatory.	Use RECOVER or REMOVE command.
-NO ENTRIES TO LIST	The workfile is empty.	None.
-WORKFILE IS CURRENTLY BEING USED BY ANOTHER PROGRAM	Self-explanatory.	Wait until the other program stops.
-LAST OPERATION ATTEMPTED WAS <command>	Self-explanatory.	None.
A TRANSLATION FILE HAS ALREADY BEEN OBTAINED	An attempt was made to GET or MAKE another workfile without saving or removing the current one.	Use SAVE or REMOVE first, then GET or MAKE the new file.
-PLEASE ISSUE A SAVE OR REMOVE REQUEST	Self-explanatory.	Use SAVE or REMOVE command.
-WORKFILE HAS BEEN ALTERED	Self-explanatory.	None.
-NO FILE NAME SPECIFIED	Self-explanatory.	Correct input and re-enter.
REQUEST IN PROCESS ...	The utility is working on your request.	Wait.
WORKFILE HAS BEEN SUCCESSFULLY RECOVERED	Self-explanatory.	None.
-HARDWARE ERROR DETECTED WHILE READING FILE <filename>	The utility has encountered a parity error while reading the specified file.	Check the integrity of your disk.
-<number> IS NOT A VALID NUMBER	Self-explanatory.	Correct input and re-enter.
-NUMBER EXPECTED	Self-explanatory.	Correct input and re-enter.
-<number> IS NOT A VALID HEXADECIMAL NUMBER	Self-explanatory.	Correct input and re-enter.
-<string> IS NOT VALID INPUT	Self-explanatory.	Correct input and re-enter.
-ONLY ONE DISK NAME SHOULD BE SPECIFIED	Self-explanatory.	Correct input and re-enter.
-ONLY ONE FILE NAME SHOULD BE SPECIFIED	Self-explanatory.	Correct input and re-enter.
UNABLE TO RECOVER WORKFILE	Self-explanatory.	None.

(continued)

Message	Possible Cause	Suggested Action
–SHIFT CODES MUST BE 4 CHARACTERS	Self-explanatory.	Correct input and re-enter.
UNABLE TO ACCESS <filename>	Self-explanatory.	See next message for the reason.
–A TRANSLATION FILE IS REQUIRED	Self-explanatory.	Use a GET or a MAKE command.
–NO TABLE NAME SPECIFIED	Self-explanatory.	Correct input and re-enter.
–TABLE NAME <tablename> EXCEEDS 20 CHARACTERS	Self-explanatory.	Correct input and re-enter.
–TABLE <tablename> IS ALREADY PRESENT IN THIS TRANSLATION FILE	An attempt was made to build a table with a name which already exists in the translation file.	Use another tablename or modify the existing table.
–TABLE <tablename> IS NOT PRESENT IN TRANSLATION FILE <filename>	An attempt was made to INCLUDE a non-existent translation table.	Check input for correct table-name and re-enter.
–NUMBER OF ENTRIES LISTED IS <number>	Self-explanatory.	None.
PLEASE ENTER THE REQUIRED TRANSLATION MODE	Self-explanatory.	Enter 0, 1 or 2.
PLEASE ENTER THE REQUIRED ANALYSIS MODE	Self-explanatory.	Enter a number in the range 0 – 255.
PLEASE ENTER THE REQUIRED HEX SHIFT CODES	Self-explanatory.	Enter two bytes of hexadecimal characters.
PLEASE ENTER THE REQUIRED MACHINE RESTRICTION	Self-explanatory.	Enter up to three different values in the range 1-3.
PLEASE ENTER THE MAXIMUM SIZE OF TRANSLATION VALUES	Self-explanatory.	Enter a value in the range 1 – 10.
– <tablename> IS A SYSTEM TRANSLATION TABLE	An attempt was made to EDIT or DELETE a translation table of type system.	None.
FILE <filename> HAS NOW BEEN SAVED	Self-explanatory.	None.

(continued)

Message	Possible Cause	Suggested Action
-TABLE <tablename> HAS MULTIPLE DIRECTORY ENTRIES	An attempt has been made to EDIT a table for which there are multiple directory entries. The EDIT will be allowed but the user must realise that the table is addressed by other directory entries.	None. This is only a warning.
-NUMBER OF TABLES CANNOT EXCEED <number>	A large number of tables is generated without requesting a save	Use a SAVE and then a GET to continue the work on the specified translation file.
-NUMBER OF DIRECTORY ENTRIES CANNOT EXCEED <number>	See above.	See above.
-SIZE OF TABLE ENTRY CANNOT EXCEED 10 BYTES	A table of type mode 1 or 2 has been discovered, the translation value cannot exceed 10 bytes.	Correct input and re-enter.
-TRANSLATION VALUES ARE <number> BYTES	Self-explanatory.	None.
-NUMBER OF DIRECTORIES SAVED IS <number>, NUMBER OF TABLES IS <number>	Self-explanatory.	None.
-NO TRANSLATION VALUES GIVEN	Self-explanatory.	Correct input and re-enter.

SYSTRANS Format

SYSTRANS contains translate tables for all versions of the keyboard, screen, line printer and serial printer currently supported by the MCP. The user has the capability to specify non-standard translation for any of these devices by setting up Sysconfig appropriately and the MCP selects the desired table(s) at warmstart. If there is no Sysconfig entry for a device, the MCP selects from SYSTRANS the system table supplied for that device. If SYSTRANS is absent at warmstart, all translation remains as it was at PO time. Therefore, when the MCP has been set up initially; that is, the first warmstart has been completed, SYSTRANS is no longer required unless the operator wishes to alter translation. The user who wishes all translation to be based on the keyboard version need do no more than at present in order to warmstart – the MCP defaults to the correct translation automatically.

Each system table has a unique name of the format:

<device>.<variant>.<country/language>.V<version no.>

<device> is KB, SP, LP or SC.
<variant> is used to distinguish different devices with the same device kind; for example, Odec/4 and 3050 line printers.

The full list of variants is as follows:

Device	Variants
KB	KB
SP	B 91, B 92
LP	B 9249-4, B 9249-30, B 9252
SC	SS1, CRT

Keyboard versions currently supported are as follows:

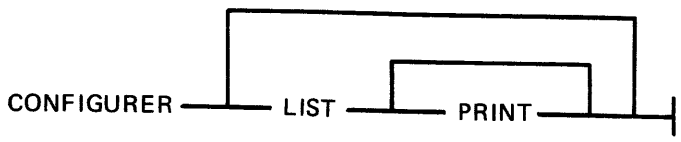
Version No.	Country/Language	Abbreviations
0	Arabic(Consoleless)	ARBC
1	United States	USA
2	United Kingdom	UK
3	France	FRNCE
4	Italy	ITALY
5	Germany/Switzerland	GRMNY
6	Brazil/Portugal	BRZIL
7	Spain	SPAIN
8	Denmark/Norway	DNMRK
9	Yugoslavia	YGSLA
10	Sweden/Finland	SWDN
12	Japan(Kawase)	JAPAN
13	Russia	RSSIA
14	Turkey	TRKY
15	Greece	GRCE
16	Iceland	ICLND
18	Sweden/Finland	SWDN
20	Arabic	ARBC
21	Hebrew	HBRW
22	Hangul	HNGL
24	Japan(Katakana)	KTKNA

Examples

1. Version 1 serial printer on B 91 : SP.B91.USA.V1
2. Version 2 serial printer on B 92 : SP.B92.UK.V2
3. Version 3 self scan 1 : SC.SS1.FRNCE.V3
4. Version 4 CRT : SC.CRT.ITALY.V4
5. Version 5 B 9252 line printer : LP.B9252.GRMNY.V5
6. Version 6 3050 line printer : LP.B924930.BRZIL.V6
7. Version 7 ODEC/4 line printer : LP.B92494.SPAIN.V7
8. Version 8 keyboard : KB.KB.DNMRK.V8

CONFIGURER (Configure B 90 System)

Railroad diagram:



CONFIGURER operates on the file SYSCONFIG to be displayed on the screen.

LIST option:	This allows the contents of SYSCONFIG to be displayed on the screen.
PRINT option:	This allows the contents of SYSCONFIG to be displayed on the printer.

CONFIGURER now operates in one of two possible modes, depending on which one of the two conditions applies when CONFIGURER is executed. The two conditions are:

1. SYSCONFIG is present on the system disk and can be opened by CONFIGURER.
2. SYSCONFIG is not present on the system disk or it cannot be opened by CONFIGURER.

If SYSCONFIG is present on the system disk and can be opened by CONFIGURER, the user is allowed to:

1. Update fields of information that are present in this OLD SYSCONFIG.
2. Insert new fields of information that are not present in the OLD SYSCONFIG.

Therefore, when all the updates and insertions have been completed via prompts from CONFIGURER, the old SYSCONFIG file is removed and a new SYSCONFIG file is created by CONFIGURER from the information supplied, and then "JOB COMPLETED" is displayed.

NOTE

Although the CREATING of a NEW SYSCONFIG can be thought of as being achieved by UPDATING and INSERTING fields in the OLD SYSCONFIG, in fact, each time CONFIGURER is being executed a completely NEW SYSCONFIG file is CREATED. Therefore, any UPDATE or INSERTION of a field in SYSCONFIG only exists when all such UPDATES and INSERTIONS have been carried out and CONFIGURER has displayed the message: "END CONFIGURER".

If SYSCONFIG is not present on the system disk, cannot be opened by CONFIGURER or is not compatible with this version of CONFIGURER, the user creates a new SYSCONFIG from scratch in the following way:

If a file is to be present in the new SYSCONFIG:

1. A message, prompting the user to enter information for this field, is displayed.
2. The user enters the information to be placed in SYSCONFIG.

Prompts marked with a "£" only appear when CONFIGURER has opened and old SYSCONFIG and the relevant entry is present in SYSCONFIG.

Any response to prompts marked with "U" (as far as the utilities are concerned) become effective as soon as the relevant utility is executed. These responses will only take effect if CONFIGURER has

come to an orderly termination. Responses to prompts which are not marked with a "U" only become effective at the next WARMSTART.

Operating Instructions

1.0	£	DO YOU WANT TO CHANGE ANY LOGGING OPTIONS? ENTER Y OR N This asks whether or not items, in the SYSCONFIG file, concerned with the logging of system messages are to be updated. For an explanation of logging in 3.05, see section 4 under LOGGING IN THE 3.05 RELEASE. Valid responses are: "Y" (yes) or "N" (no). If the response is "N", go to 2.0.
1.1	£	SPO LOGGING IS SWITCHED ON or SPO LOGGING IS SWITCHED OFF
1.1.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" and SPO logging is switched on go to 1.2. If "N" and SPO logging is switched off go to 1.3
1.1.2	£	IS LOGGING OF SPO MESSAGES REQUIRED? ENTER Y OR N This asks whether or not logging of system messages is required at the next warmstart ("Y" is advisable). If the response is "N" go to 1.3
1.2	£	LOGGING OF SYSTEM MESSAGES IS CYCLIC or LOGGING OF SYSTEM MESSAGES IS NON-CYCLIC depending on the current form that system messages are to be logged in.
1.2.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 1.3
1.2.2	£	DO YOU WANT LOGGING OF SYSTEM MESSAGES TO BE CYCLIC? ENTER Y OR N This asks whether system messages are to be logged in CYCLIC or NON-CYCLIC form. Valid responses are: "Y" for CYCLIC or "N" for NON-CYCLIC.
1.3	£	CURRENT SIZE OF PRIMARY LOG FILES IS <integer> SECTORS where <integer> = the size, in sectors, of the four primary log files.
1.3.3	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" and SPO logging is switched on, go to 1.4. If "N" and SPO logging is switched off go to 1.5.
1.3.2	£	ENTER SIZE OF PRIMARY LOG FILES IN SECTORS – MINIMUM IS 32, MAXIMUM IS 16383 This requests the size, in sectors, of the four primary log files. A valid response is an <integer> in the range 32 to 16383.
1.4	£	CURRENT SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IS <integer> SECTORS
1.4.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 1.5.
1.4.2		ENTER THE SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IN SECTORS – MINIMUM IS 32, MAXIMUM IS 262140 A valid response is an <integer> in the range 32 to 262140.

1.5	£	CURRENT SIZE OF SECONDARY SYSTEM MAINTENANCE ENTRY FILE(S) IS <integer> SECTORS
1.5.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 1.6.
1.5.2		ENTER SIZE OF SECONDARY SYSTEM MAINTENANCE ENTRY FILE(S) IN SECTORS – MINIMUM IS 32, MAXIMUM IS 262140 A valid response is an <integer> in the range 32 to 262140.
1.6	£	CURRENT BACKUP DISK FOR LOG FILES IS <disk-id> or CURRENT BACKUP DISK FOR LOG FILES IS SYSTEM DISK depending on whether backup log files are to reside on the disk named <disk-id> or on the system disk.
1.6.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 2.0.
1.6.2		ENTER ID OF BACKUP DISK FOR LOG FILES This requests the identifier of the disk that backup log files are to reside on. Valid responses are: 1) a <disk-id> 2) <blank>, in which case backup log files will reside on the system disk.
2.0	£	CURRENT TEXT TO BE ZIPPED AT WARMSTART IS x (where "x" is the text to be zipped at the next warmstart) or NO TEXT WILL BE ZIPPED AT WARMSTART
2.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 3.0.
2.2		ENTER TEXT TO BE ZIPPED AT WARMSTART This requests the text to be zipped at the next warmstart. No checking is performed on the text entered. However, only the first 255 characters of the text entered are zipped.
3.0	£	CURRENT POWER OFF MESSAGE IS x (where "x" is the message to be displayed when a session is terminating) or THERE IS NO POWER OFF MESSAGE
3.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 4.0.
3.2	£	ENTER POWER OFF MESSAGE This requests the text of the message to be displayed when a session is terminating. The text must not be more than 80 characters in length.
4.0	£	CURRENT LANGUAGE DICTIONARY FILENAME IS <file-name> where <file-name> is the name of the current local language message file.
4.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 5.0.
4.2	U	ENTER FILE NAME OF REQUIRED MESSAGE FILE This requests the <file-id> of the required local language message file. Valid responses are: 1) a <file-id> 2) <blank>, in which case the message file will be "SYSLANGUAGE".

5.0	£	DO YOU WANT TO CHANGE ANY PRINTER BACKUP OPTIONS? ENTER Y OR N This asks whether or not items in the SYSCONFIG file concerned with printer backup files are to be updated. If response is "N" go to 6.0.
5.1	£	CURRENT PRINTER BACKUP FILE SIZE IS <integer> RECORD(S) where <integer> is the maximum size, in records, of a printer backup file.
5.1.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 5.2.
5.1.2	U	ENTER PRINTER BACKUP FILE SIZE IN RECORDS – MINIMUM IS 1, MAXIMUM is 999999. This requests the maximum size, in records, of printer backup files. A valid response is an <integer> in the range 1 to 999999.
5.2	£	PRINTER FILES ARE SENT TO BACKUP or PRINTER FILES ARE NOT SENT TO BACKUP
5.2.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 5.3.
5.2.2	U	DO YOU WANT PRINTER FILES TO GO TO BACKUP? ENTER Y OR N This requests whether or not printer files are to go to backup.
5.3	£	CURRENT PRINTER BACKUP DESIGNATE DISK IS <disk-name> or CURRENT PRINTER BACKUP DESIGNATE DISK IS SYSTEM DISK depending on whether printer backup files are to reside on the disk named <disk-name> or on the system disk.
5.3.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 6.0.
5.3.2	U	ENTER NAME OF PRINTER BACKUP DESIGNATE DISK This requests the disk that the printer backup files are to reside on. Valid responses are: 1) a <disk-name> 2) <blank>, in which case printer backup files reside on the system disk.
6.0	£	DO YOU WANT TO CHANGE ANY SPO OPTIONS? ENTER Y OR N This asks whether or not items in the SYSCONFIG file concerned with the local SPO and the remote SPO are to be updated. If the response is "N" go to 7.0.
6.1		DOES THE SYSTEM HAVE A LOCAL SPO? ENTER Y OR N This asks whether or not a local SPO will be present at the next warmstart. If the response is "N" go to 6.2.
6.1.1	£	LOCAL SPO IS SCA or LOCAL SPO IS SPA depending on whether the local SPO is a screen (SCA) or a serial printer (SPA).
6.1.2	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 6.2.

6.1.3		ENTER LOCAL SPO OPTION, EITHER SCA OR SPA This asks whether the local SPO is a screen (SCA) or a serial printer (SPA). NOTE If the user's choice is not applicable to the machine's configuration, the final choice of the type of local SPO will be determined by the MCP at warmstart.
6.2	£	THE SYSTEM HAS A REMOTE SPO or THE SYSTEM DOES NOT HAVE A REMOTE SPO depending on whether or not a remote SPO will be present at the next warmstart.
6.2.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" and the system has a remote SPO go to 6.3. If "N" and the system does not have a remote SPO go to 7.0.
6.2.2		DOES THE SYSTEM HAVE A REMOTE SPO? ENTER Y OR N
6.3	£	CURRENT REMOTE SPO MCS IS x where "x" is the <file-id> and initiating message of the remote SPO's MCS.
6.3.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 7.0.
6.3.2	£	ENTER FILENAME AND INITIATING MESSAGE OF REMOTE SPO MCS This requests the MCS of the remote SPO. Valid responses are: 1) a <file-id> followed by the initiating message. No checking is performed on the initiating message. The length of the <file-id> plus initiating message must not be greater than 40 characters. 2) <blank>, in which case the MCS and initiating message will be "MCSOBJ DC CC".
7.0	£	DO YOU WANT TO CHANGE ANY TRANSLATION FILES? ENTER Y OR N This asks whether or not the current set of translation tables is to be updated. See BUILDTRANS utility for details about translation. If the response is "N" go to 8.0.
7.1	£	CURRENT NUMBER OF TRANSLATION TABLES IS <integer>
7.1.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" and <integer> is greater than zero go to 7.2. If "N" and <integer> equal to zero go to 8.0.
7.1.2		ENTER NUMBER OF TRANSLATION TABLES – MAXIMUM IS 255 A valid response is an <integer> not greater than 255.
7.2		TABLE NUMBER <integer> where <integer> is the number of the translation table that is currently being updated or inserted. If the set of translation tables is to be updated, this is displayed for all translation tables in the new SYSCONFIG file.
7.2.1	£	CURRENT DEVICE KIND AND NUMBER IS abn "ab" is the CMS DEVICE MNEMONIC and "n" is the DEVICE NUMBER of a translation table.
7.2.2	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" go to 7.3.

7.2.3		ENTER DEVICE KIND AND NUMBER This requests a CMS DEVICE KIND and a DEVICE NUMBER for a translation table. A valid response is abn, where: "ab" is one of the CMS DEVICE MNEMONICs "SP", "LP", "SC" or "KB" and "n" is a DEVICE NUMBER so that "A" ≤ n ≤ "Z".
7.3	£	CURRENT TRANSLATOR NAME IS <identifier>
7.3.1	£	DO YOU WANT TO CHANGE IT? Y OR N If the response is "N" and there are more tables to be completed, go to 7.2. If "N" and all tables have been completed go to 8.0.
7.3.2		ENTER NAME OF TRANSLATOR A valid response is an <identifier> of not more than 20 characters. If there are more tables to be completed, go to 7.2.
8.0		END CONFIGURER This is the normal job terminating message. The utility has now released the file under the name "SYSCONFIG" onto the system disk.

Output Messages

Message	Possible Cause	Suggested Action
INVALID SELECTION, PLEASE RETRY	Self explanatory. Request is repeated.	Use a valid response, see request.
<faulty input> IS NOT A VALID FILESIZE	The value for the filesize is illegal. Request is repeated.	use a valid response, see request.
INVALID CHARACTER IN <identifier>	<identifier> has invalid character(s) in it. Request is repeated.	Valid characters are A-Z, 0-9, . (dot) – (dash).
<faulty input> IS NOT A VALID DISK NAME	Probably the disk name contains more than seven characters. Request is repeated.	Check input then re-enter a valid disk name.
<faulty input> IS NOT A VALID FILE NAME	Probably the file name contains more than 12 characters. Request is repeated.	Check input then re-enter a valid file name.
TEXT LENGTH EXCEEDS UTILITY LIMIT	Text contains more than 80 characters. Request is repeated.	Re-enter text cut down to 80 characters or less.
<faulty input> IS NOT A VALID INPUT FOR THE NUMBER OF TRANSLATION TABLES	The value for the number of translation tables is illegal. Request is repeated.	Enter an integer in the range 0-255.
HARDWARE ERROR DETECTED WHILE WRITING TO FILE <file-name>	A parity error was encountered while writing to the specified file. The utility terminates and no new SYSCONFIG file is created.	

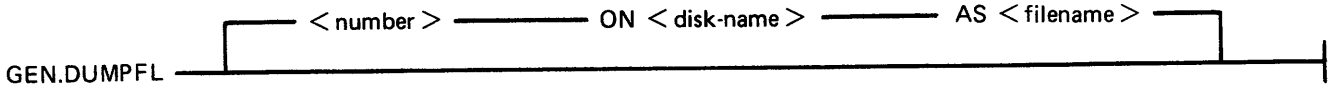
(continued)

Message	Possible Cause	Suggested Action
HARDWARE ERROR DETECTED WHILE READING FILE <file-name>	A parity error was encountered while reading from the specified file. The utility terminates and no new SYSCONFIG file is created.	

GEN.DUMPFL (Create Empty B 90 Memory Dump File)

The function of the GEN.DUMPFL utility is to create, on a disk specified by the user, a single-area file to which the contents of memory can be dumped. The name and size of the file can be specified by the user. As a check against bad sectors, the utility writes a repetitive pattern of ASCII "A" through "L" to each record of the created file.

Railroad diagram:



where:

size	specifies the file size in K bytes. If specified, it must be an integer in the range 64 to 11492; if not specified, the default value 1024 is assumed (1024 K bytes being sufficient to contain 8 pages of memory).
disk-name	if not specified, the system disk is assumed.
file-name	if not specified, a file named MEMDUMP is created. Note that memory can only be dumped (by the CMS bootstrap WARMBOOT) to disk files named MEMDUMP. So, if any other file-name is specified, the file-name must be changed to MEMDUMP before memory can be dumped to it successfully.

If it is intended to dump memory to a fixed disk, the size of the file created on that disk by GEN-.DUMPFL must be large enough to hold all of memory.

If it is intended to dump memory to removable disks, the size of the file created on a removable disk by GEN.DUMPFL need not be large enough to hold all of memory as the Dump Memory to Disk function of CMS bootstrap WARMBOOT can dump to a sequence of removable disks containing MEMDUMP files.

When one single-area file has been created on a disk, the user is informed by the following message:

"DUMPFILE <file-name> CREATED ON <disk-name>
EARLIER DUMPFILES OF SAME NAME ARE REMOVED"

If the size of the file created on a removable disk is smaller than the file size requested, the user is informed by the following message:

"MORE DISK SPACE IS REQUIRED
LOAD ANOTHER – NOT THE SAME – DISK
ENTER ITS NAME"

The utility will wait for the user to ready another removable disk and enter its name. It will then create a file with the name originally specified by the user. The size of this file will be the difference between the size originally requested and the size of the first file created, or the size of the largest area available on the disk, whichever is smaller. Requests for more disk space will continue to be made until the sum of the size of the files created is equal to the size originally requested.

NOTES:

1. An area of at least 256 sectors must be available on a disk before GEN.DUMPFL will attempt to create a dump file. If no such area exists, the following message will be displayed:

"INSUFFICIENT DISK SPACE TO CREATE MEMORY DUMP FILE"

The user must then remove files from the disk or load another disk.

2. A sequence of removable disks may include cartridge and/or floppy disks. However, it is recommended that the types are not mixed. A sequence of removable disks may include Caelus, SDI or Winchester disks, although it is recommended that the types are not mixed.

3. When dumping memory to a sequence of removable disks, disks need not be loaded in the same order as they were submitted to GEN.DUMPFL.

When the total file size created, on a single disk or on a sequence of removable disks, is equal to the size requested by the user, the following message is displayed before GEN.DUMPFL goes to EOJ:

"ALL SPACE REQUESTED HAS BEEN RESERVED"

Examples:

To create a file named MEMDUMP on the system disk with a size of 1024 K bytes of memory, enter:

GEN.DUMPFL

To create a file named DUMP.FILE on disk ABC with a size of 80 K bytes of memory, enter:

GEN.DUMPFL 80 ON ABC AS DUMP.FILE

Output Messages

Message	Possible Cause	Suggested Action
SIZE TOO SMALL	Size requested less than 64	Re-input.
SIZE TOO LARGE	Size requested greater than 11492	Re-input.
DISK NAME TOO LARGE	Disk name greater than seven characters.	Re-input.
FILE NAME TOO LONG	File name greater than 12 characters.	Re-input.
DISK NOT FOUND	Incorrect disk name entered.	Re-input.
INVALID DISK NAME	Incorrect disk name entered.	Re-input.

INSTALL (Disk File Copy From System to Fixed Disk)

The INSTALL utility provides a simple method of copying all necessary system files from a mini or cartridge bootstrap system disk to any disk, except 1 Mb and 3 Mb disks. This is achieved by having the INSTALL utility resident on the Bootstrap disk. When INSTALL is executed, it copies the contents of the Bootstrap system disk, except the utility itself, to the named disk, including those files which cannot normally be copied because of "OTHERUSE" restrictions on the COPY utility FREE and SHARED options.

Railroad Diagram:

<disk name> is the name of the disk to which files are to be copied.

The utility does not support the star-file feature *<file-name>. When complete, the operator can power off and warmstart the "installed" fixed disk.

The following conditions must be true for the INSTALL utility to function:

1. There is a suitable mix (if not, INSTALL will not be loaded).
2. The executing system is a B 90.
3. The INSTALL utility is resident on the system disk.
4. The system disk is a Burroughs Mini disk.
5. The named output disk is not a Burroughs Mini disk.

NOTE

A suitable mix occurs when no user programs, utilities or intrinsics are running.

If any of the conditions 2-5 do not apply, an error message is displayed and the utility terminates. (See Output Messages.)

Providing conditions 1-5 are satisfied, and the specified output disk is on line, the utility copies the contents of the Bootstrap disk to the named disk. The utility does not copy itself.

As each file is copied, the following message is displayed:

<file-name> COPIED

If a file cannot be copied for any reason, an error message is displayed (see Output Messages). INSTALL will not copy keyfiles.

NOTE

If a keyfile is found on the Bootstrap disk, a warning message is displayed.

If a file with the same file-name exists on the output disk, the duplicate file will be removed from the output disk if the filetype is the same as the file being copied. If the filetypes are different, the user is prompted by the MCP to change the name or to remove the duplicate file on the output disk.

Output Messages

Message	Possible Causes	Suggested Action
<file-name> COPIED TO <file-name>	Successful copy.	None.
<file-name> REMOVED	Self-explanatory.	None.
PLEASE POWER OFF THE SYSTEM DISK AND REWARMSTART FROM <disk-id>	All files have been successfully copied.	None.
<file-name> NOT COPIED	Self-explanatory.	See second message given for explanation.
– FILE IS A KEYFILE	Self-explanatory.	Remove keyfile from Bootstrap disk.
– CANNOT ALLOCATE THIS FILE ON DISK <output disk-id>	Self-explanatory.	None.
HARDWARE ERROR DETECTED WHILE READING FILE	Self-explanatory.	Use another disk.
HARDWARE ERROR DETECTED WHILE WRITING TO DISK <output disk-id>	Self-explanatory.	Use another disk.
THE FOLLOWING <integer> FILES WERE NOT COPIED – <file-name list>	Self-explanatory.	See second message given for explanation.
PLEASE USE "COPY" TO COPY THESE FILES, FROM ANOTHER DISK IF NECESSARY	Self-explanatory.	See second message given for explanation.
BEFORE POWERING OFF DISK <system disk-id> AND WARMSTARTING FROM DISK <disk-id>	One or more files could not be copied by the utility.	See second message given for explanation.
– THIS SYSTEM IS NOT A B90	INSTALL can only be run on a B 90 system.	None.
THIS PROGRAM MUST BE EXECUTED FROM THE SYSTEM DISK	INSTALL must reside on the system disk.	Copy the INSTALL utility onto the system disk and re-run.
– SYSTEM DISK <disk-id> NOT A MINI OR CARTRIDGE DISK	Self-explanatory.	Use the correct type of disk and re-run the utility.
– SYSTEM DISK <disk-id> CANNOT BE ACCESSED	Self-explanatory.	System disk may not be on line.

(continued)

Message	Possible Causes	Suggested Action
– DISK <output disk-id> IS A MINI DISK	The output disk is an invalid mini disk for example, 1 Mb or 3 Mb mini.	Remove the disk and use one of the following types: 211 fixed disk 201I fixed disk cartridge
– DISK <output disk-id> IS CURRENTLY LOCKED BY ANOTHER PROGRAM	The output disk is currently being used by another program.	Wait until other program has finished and re-submit.
UNABLE TO CONTINUE WITH INSTALL	Self-explanatory.	See second error message given for explanation.
– DISK <output disk-id> IS FULL, PLEASE REFORMAT FOR LARGER NUMBER OF FILES	Self-explanatory.	Reformat disk using DSKUTL.
– HARDWARE ERROR DETECTED ON DIRECTORY OF DISK <disk-id>	Self-explanatory.	Use another disk.
– DISK <output disk-id> IS CURRENTLY LOCKED BY ANOTHER PROGRAM	Self-explanatory.	Wait until other program has finished using disk and then retry.
– DISK <output disk-id> IS NOT ON LINE	Self-explanatory.	Insert and/or make ready output disk

PATCHMAKER

(Patching B 90 Machine-Code Object Program Files)

This utility gets the patches from either the SPO via displays and accepts or from a disk file. The patches are then made to the relevant code file. Stringent conditions are enforced to make the patch, including the necessity to apply each patch in the correct order. Each line of patch data contains an extra byte which is used as a checksum for that line. If a line of data is incorrectly entered, the operator is prompted to re-enter that line – this does not invalidate previously entered lines of data. An overall checksum is also included to verify the entire patch. All previous patches must be applied before the next patch.

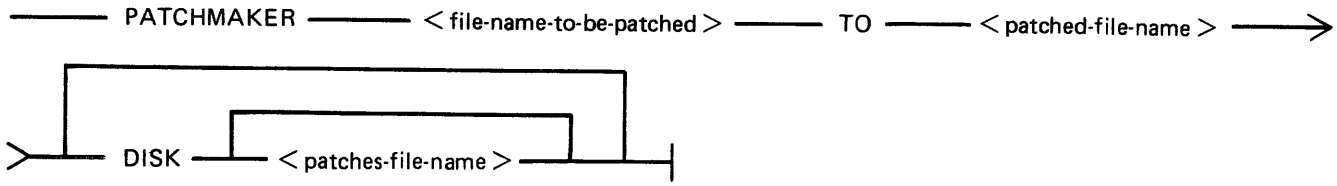
In order to achieve continuity of system code-file names, the following naming convention will be used:

For B 90 MCP	MCPXn.nn.nn
For B 90 BILINTERP	BILXn.nn.nn
For B 90 COBOLINT	COBXn.nn.nn
For B 90 SORTINTRINSIC	SINTXn.nn.nn
For B 90 NDLINTERP	NDLXn.nn.nn
For B 90 MICROPOL	MICXn.nn.nn
For B 90 SAU	SAUn.nn.nn

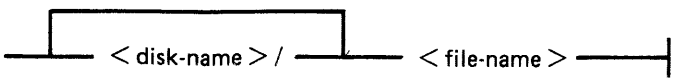
where n.nn.nn is the software mark, release and patch level (for example, MCPX3.04.09).

It is essential that an unused copy of all micro-coded software items is retained for patching. It is not possible to patch a B 90 MCP that has been used in normal B 90 operations. This is because certain MCP tables included in the code file are modified during operation. This modification would cause the check digit calculations in PATCHMAKER to fail. Use either the Stand-Alone Utility (SAU) COPY function, MCP COPY or LD utilities to create unused copies of all system software for patching purposes, and also to create backup copies of patched software. All files which PATCHMAKER requires must reside on disk.

PATCHMAKER is executed via an initiating message which contains all the information concerning the mode of entry of the patches (displays and accepts or patch file) and the names and locations of the files involved. The initiating message for PATCHMAKER is as follows:



Where <file-name-to-be-patched>, <patched-file-name> and <patches-file-name> are defined as:



<file-name-to-be-patched>	This is the name of the file that is to have the patches applied to it.
<patched-file-name>	This is the name which will be given to the patched file, if the patching has been successful.
DISK	If the keyword DISK is specified, PATCHMAKER will look for a file from which to extract the patches. If no <patches-file-name> is specified, PATCHMAKER will expect to find a file called "PATCHES" on the system disk. In either case, the file must have been created by PATCHMAKER (see note following) or the appropriate internal Burroughs program.

NOTE

PATCHMAKER has no option to create a patch file. The utility will create a patch file if, after patches have been entered via accepts, a fatal error occurred or the operator terminated execution after a non-fatal error. The patch file created under these circumstances should never be applied to a code file, as the result may be incorrect.

If the keyword DISK does not appear in the initiating message, the entry of patches will be by displays and accepts via the SPO. The utility will display:

ENTER PATCHES NOW

The patches must now be entered by accepts, via the SPO when prompted, from the hardcopy provided. The characters must be entered exactly as supplied, although spaces are not significant and may be entered as found convenient. The utility will ask for resubmission of lines which are incorrect.

When the utility has correctly applied the patches, the following message will be displayed before PATCHMAKER terminates:

PATCHES TO <file-name> HAVE BEEN SUCCESSFULLY APPLIED AND SAVED AS FILE <file-name>.

Patching Procedure

A sample B 90 CMS System Software Flash is shown in the following paragraphs.

Sample Flash

B 90 CMS System Flash No. 3.04-xx.
SUBJECT: B90 MCP VERSION 3.04.04
PROBLEM: The FUNCTION F can result in.....

: :
: :
: :

2032801

: :
: :
: :
TEMPORARY FIX:

Follow the instructions for PATCHMAKER found in section 8 of the Series B 90/B 900/CP 9500, B 1800/B 1900 Computer Management Systems (CMS) System Software Operation Guide, Form No 2032801.

The sub-title of the relevant procedure is "PATCHING IMPLEMENTATION"

A MCPX3.04.04

B MCPX3.04.05

C 3030 3030 3030 304D 4350 D9
5833 2E30 332E 3037 2000 E4
0000 00B5 3D6F C12C 0132 32
3820 38C4
0332 3837 2033 3020 1C7F F6
3120 0A5A 001C 0216 3B0F E2
197A

D .... C47A * (ASTERISK)
NOTE: The character * must be entered. The word ASTERISK is included for clarification only.

E .... MCPX

WARNING

This is not a live patch. It is shown only to illustrate the new patching procedure.

The following procedure is designed to make patching easier for the user to understand and implement. The format for each patch, issued in a CMS Flash, will conform to the sample Flash shown previously.

System Flashes which involve patches to system software will consist of a number of "labelled inserts" (A through E) where:

A .... Is the file name of the file to be patched.

B .... Is the file name to be given to the patched file.

C .... Is the actual patch.

D .... Is the patch checksum with terminating character.

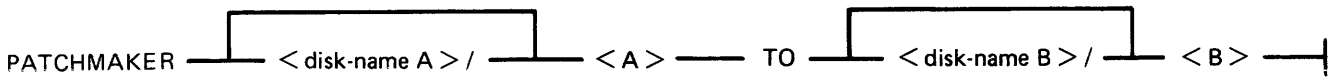
E .... Is the executing name of the file just patched.

The instructions contained in the following procedure contain "labelled blanks" (A through E). When applying a patch using this procedure, the "labelled blanks" are replaced with the text of the corresponding "labelled inserts" of the Flash.

Operating Instructions

Before executing the program PATCHMAKER, perform steps 1 and 2.

1. Ensure that the program PATCHMAKER is resident on the system disk.
2. Copy an unexecuted version of the file < A > to either the system disk or user disk. This is the file to be patched. SAU COPY, CMS COPY or CMS LD may be used to copy the file.
3. Execute PATCHMAKER with the following initiating message



Where:
<disk-nameA> is the name of the disk where file < A > resides.
<disk-nameB> is the name of the disk on which file < B > is to be saved.

The system displays:
"ENTER PATCHES NOW"
"<mix number>/PATCHMAKER ACPT"

4. Enter "AX <mix number> space" < C > for each line of the flash as supplied.
   If entering the patch through a console SPO (SELF-SCAN, CRT or Printer), terminate each line by pressing OCK1.

   If entering the patch through a data comm SPO, terminate each line with the transmit (XMT) key.

   The spaces contained in < C > can be entered. This makes the data easier to read from the flash and to enter via the keyboard. These spaces are ignored by PATCHMAKER.

   "ERROR IN LAST LINE ENTERED – RESUBMIT"
   "<mix number>/PATCHMAKER ACPT"
   Check the content of the last line entered and re-enter the data correctly.
5. When all the lines of < C > have been entered correctly, enter "AX <mix number> space" < D >.
   Ensure that the last character entered on this line is the character * (asterisk or star). This character informs PATCHMAKER that this is the last entry and is the overall checksum for the patch.

   For non-English keyboards, this is the key which produces the ASCII code @2A@.

   The utility will then display:

   PATCHES TO <disk-nameA>/< A > HAVE BEEN SUCCESSFULLY APPLIED, AND SAVED AS FILE <disk-nameB>/< B >

   and PATCHMAKER will go to End Of Job.

The procedure for installing the patched software is dependent on the backup medium used on the system. If the backup medium used on the system is disk, follow steps D1 through D6. Otherwise, follow steps T1 through T7 for systems which use tape as the backup medium.

D1. Logically PO the disk containing < B > and if it is a removeable disk, write disable on it.

D2. Load Stand Alone Utilities (SAU).

D3. Using SAU RM, remove < A > and < E > from each disk in your library (including your system disk). Copy the file < B > to all of the disks which contained < A > or < E >.

D4. Retain a copy of < B >, which must never be executed in order that future patches can be applied if necessary. The patch implementation is now complete for your library files.

D5. On the system disk, using SAU CH, enter
CH <system-disk-name>/< B > TO <system-disk-name>/<E>

D6. Enter "WS"

The system can now be warmstarted and will be fully operational with the newly patched software.

For systems which use tape as backup medium, the procedure is similar except that the CMS tape utilities (SYCOPY, LD, COPY) are used.

T1. Follow your normal library updating procedures to replace < A > with < B >.

T2. Copy < B > to the system disk.

T3. Logically PO the system disk.

T4. Load Stand Alone Utilities (SAU).

T5. Using SAU RM, enter
RM <system-disk-name>/< E >

T6. Using SAU CH, enter
CH <system-disk-name>/< B > TO <system-disk-name>/< E >

T7. Enter "WS".

The system can now be warmstarted and will be fully operational with the newly patched software.

Patchmaker Error Messages

The following error messages are produced for a bad initiating message. Check the initiating message, and re-enter corrected message.

1. INCOMPLETE INITIATING MESSAGE
2. UNEXPECTED PARAMETER <offending symbol> IN INITIATING MESSAGE.
3. UNEXPECTED CHARACTER <character> IN INITIATING MESSAGE.
4. KEYWORD - TO - MISSING FROM INITIATING MESSAGE.
5. DISK NAME <disk-name> IN INITIATING MESSAGE TOO LONG.
6. FILE NAME <file-name> IN INITIATING MESSAGE TOO LONG.
7. ENTER PATCHES NOW.
   Prompts the user to begin entering the patches.

The following error messages are produced if an error is detected when entering patches via the SPO.

8. ERROR IN LAST LINE ENTERED – RESUBMIT

   The line just entered, checksum has failed. Check the line for errors and re-enter.

9. ERROR IN PATCHES ENTERED – RUN ABORTED

   The overall checksum for the patches entered has failed. Check the input to PATCHMAKER, then restart patch procedure.

10. PATCHES HAVE BEEN ENTERED CORRECTLY.
    Displayed subsequent to correct SPO input of patches.

11. PATCHES ARE FOR FILE <file-name> NOT <file-name>.

    The file name entered, in the initiating message for the file to be patched, is different to the file name embedded in the patches. Check the input to PATCHMAKER, then restart patch procedure.

The following error messages inform the operator of an error condition and ask the operator to take corrective action. The following prompt is displayed in addition to the error and correction messages.

ENTER C TO CONTINUE, S TO STOP

If the operator is able to correct the error, "C" should be AXed. This will cause PATCHMAKER to repeat the checks and continue.

If for some reason the operator does not wish to continue, "S" should be AXed. If the patches were entered via accepts and displays, the file "PATCHES" is saved and the message:

FILE PATCHES SAVED

is displayed. The utility will stop.

If a response other than "C" or "S" is given to the prompt above, the following error message is displayed:

RESPONSE MUST BE C TO CONTINUE, S TO STOP

At this point "C" or "S" must be AX'ed to PATCHMAKER.

12. FILE <file-name> FOR PATCHING IS ABSENT.
    PLEASE MAKE <file-name> AVAILABLE.

    The file named for patching is not present on the disk specified. To continue, copy the required file to <file-name>. Then AX "C" to PATCHMAKER.

    To stop the utility, AX "S" to PATCHMAKER.

13. INITIAL CHECKSUM DISCREPANCY FOR FILE <file-name>.
    PLEASE MAKE CORRECT FILE AVAILABLE AS <file-name>.

    The file on the system with <file-name> is either corrupt or the wrong file.

To continue, copy a good version of the correct file to <file-name>, then AX "C" to PATCHMAKER.

To stop the utility, AX "S" to PATCHMAKER.

14. <file-name> IS NOT A FIRMWARE FILE.
PLEASE MAKE CORRECT FILE AVAILABLE AS <file-name>.

The file on the system with <file-name> is not the correct type of file to be patched. To continue, copy the correct file to <file-name>. Then AX "C" to PATCHMAKER.

To stop the utility, AX "S" to PATCHMAKER.

15. DATE OF <file-name> CONFLICTS WITH PATCH DATE
PLEASE MAKE CORRECT FILE AVAILABLE AS <file-name>

16. DISK <disk-name> IS NOT AVAILABLE.
PLEASE MAKE DISK <disk-name> AVAILABLE.

The disk <disk name> is not on the system.
To continue, load the disk called <disk name> or ready it using RY if it is not ready. Then AX "C" to PATCHMAKER.

To stop the utility, AX "S" to PATCHMAKER.

The following error messages inform the operator of errors encountered during patching. PATCHMAKER will terminate.

If the patches were entered via displays and accepts, the patches will be saved in a file on the system disk called "PATCHES". This allows PATCHMAKER to be re-run with the option DISK, meaning the operator does not have to input the patches again. The following message will be displayed to inform the operator that this has been done:

17. FILE PATCHES SAVED

18. ERROR ON READING RECORD <record-number> OF FILE <file-name>

A read error has been detected when reading from the file <file-name> which was being patched. Re-run PATCHMAKER with a good version of the file to be patched.

The following messages are caused by errors encountered during patching. They may be caused by tampering with the file "PATCHES".

19. PATCHING ERROR – PATCH TOO LARGE FOR <file-name>

Request technical assistance.

20. PATCHING ERROR – CANNOT FIND RECORD IN <file-name> FOR PATCH.

Request technical assistance.

21. PATCHING ERROR – TERMINATING CHECKSUM FOR <file-name>
INCORRECT AS <file-name>

This is displayed after a successful PATCHMAKER run.

PMB90 (Analyze B 90 Memory Dumps)

This utility is an interactive program which produces a formatted print of the contents of a memory dump tape or disk file produced by the memory dump feature of the B 90 bootstrap ROM. The tape must be labelled "MEMDUMP/MEMORY" or the disk file must be named MEMDUMP on the system disk unless otherwise specified to PMB90. Neither MEMDUMP file-name nor disk-name may start with a number.

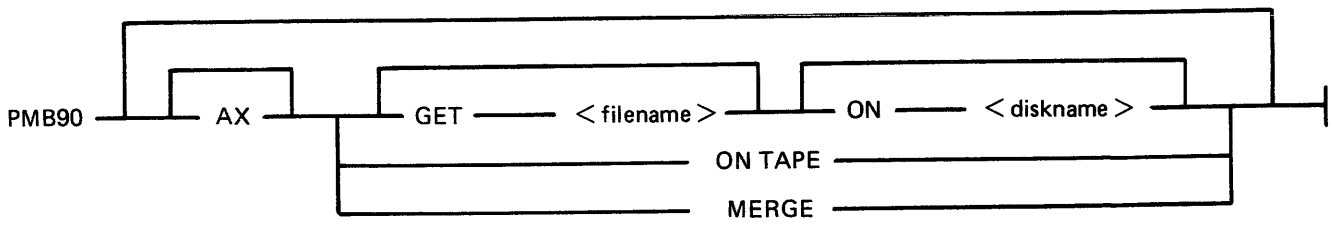
The utility requires the following files on the system disk. These are unchanged from earlier releases:

PMB90	– object code file
PMBHELP	– data file of prompt messages
PMBERROR	– data file or error messages
PMBM.xxxxx	– data file for information on MCP xxxxx
PMBO.xxxxx	– data file for information on MCP xxxxx

The x values vary with each release: the files provided with each release must be used with that release, otherwise incorrect analysis may be made. The utility checks version numbers of its data files and issues a warning if they are not the same as the PMB90 version. All data files include version numbers.

Starting The Utility

The utility can be executed with a number of options in the initial message. The format is as follows:



The meaning of these options is as described in the following paragraphs.

AX

Memory dump analyser will issue an accept when input is required from the operator. AX is ignored if PMB90 is initiated via datacomm.

GET <filename>

This option informs PMB90 that it should search for a memory dump file on the system disk (or on the disk specified in the ON option) with the name specified in <filename>. If the filename option is not used, PMB90 will search for a file called MEMDUMP. will search for a file called MEMDUMP.

If PMB90 is initiated through datacomm and no GET option is used, the utility will display a dump file choice menu.

ON <diskname>

This option informs PMB90 on which disk it has to search for a memory dump file.

NOTE

If GET and ON option are both specified, then the GET option should precede the ON option.

ON TAPE

This option informs PMB90 that the memory dump file has to be found on a magnetic tape labelled "MEMDUMP/MEMORY".

MERGE

This option informs PMB90 that its memory dump file is to be assembled from two or more files held on different disks. This facility is required for situations in which the memory dump file of the machine being dumped is too large to fit onto one removable disk.

NOTE

The specification of MERGE in the initiating message excludes specification of any other option.

PMB90 can also be started from a TD 830 capable datacomm terminal. Once started, the utility displays a screen on which the user may select the filename to be analysed, and the device on which has to search for this file. If the file is found and successfully opened, the utility displays the base menu screen. From this screen, the user may choose one of the options described in the following paragraphs with the exception of the HELP option. When choosing one of these options the utility displays another screen where the user may choose one of the specific parts from the memory dump file to be analysed.

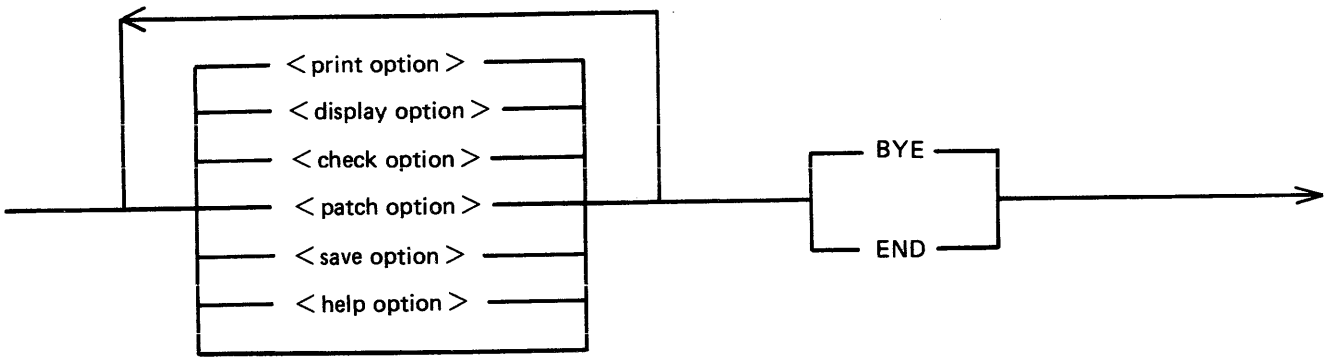
Using The Utility

Where the console is to be used for communication, the user is informed if no console is available, and is given the opportunity to make a console file available. Non-printer console files allow up to 224 characters input.

The analysis takes place in an interactive manner. All available options and instructions on how to use them are provided in response to the input "HELP". Further details on a particular option are provided in response to the input "HELP option". A knowledge of the MCP is required in order to diagnose the reason for any particular memory dump.

The complete dump options are given here in railroad diagram format, with further details in the following paragraphs.

Syntax Of Input Specifications



Print

This option specifies that output from the following list of <print-option>s is to be printed on the console.

Display

This option specifies that output from the following list of <print-option>s is to be displayed on the SELF-SCAN screen. In this case the SET.SCREEN parameter should have been used in the initial message.

Check

The two <check-option>s are MEMORY.LINKS and ALL.MEMORY. The MEMORY. LINKS option gives a print of all memory links from the initial pointer (PTRX) until the end of the chain or an inconsistency is reached.

Save

This option enables a copy of the dump (patched if required) to be made on the specified disk with the specified file-name.

Patch

This option enables an invalid area of memory to be patched in the dump, to enable PMB90 to continue its analysis.

Help (Non-Interactive PMB90 Only)

If no <help token> is specified, the list of available <options> is displayed. Otherwise, for each specified <help token>, the available sub-options are listed (if any) with a short semantic description.

NOTE

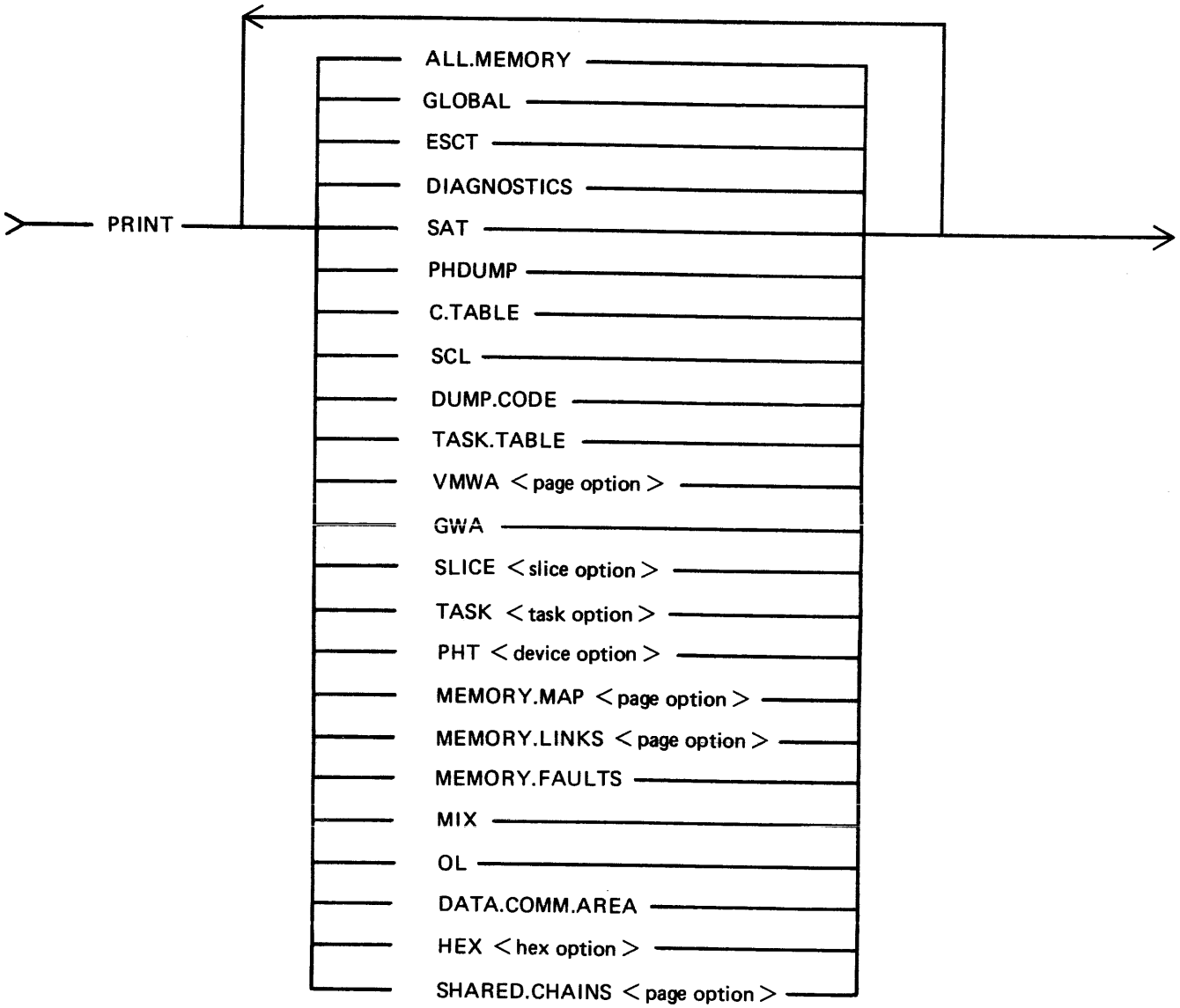
All <identifiers> input in the same line as "HELP" will be taken as <help tokens>.

Also "HELP" is not valid input when PMB90 is being run as a datacomm applications program. The user is guided by the menu displays.

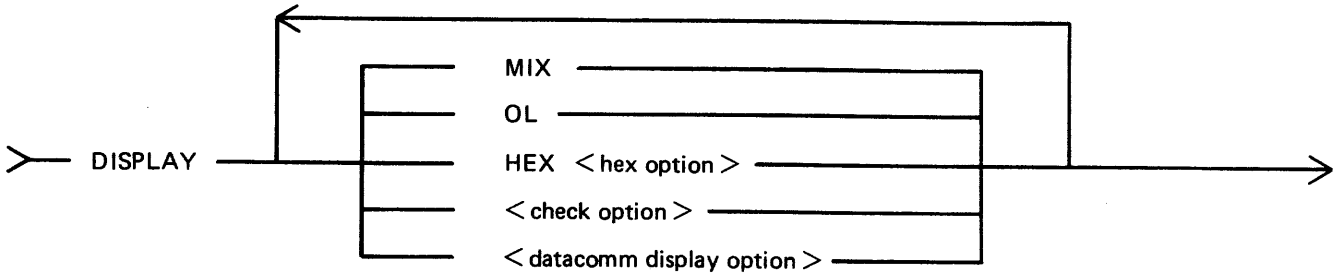
End, Bye

These alternative options cause PMB90 to go to normal EOJ.

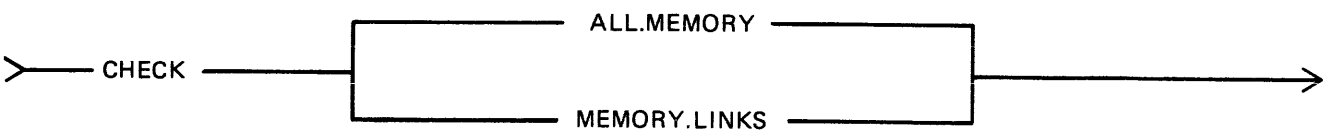
<print option>



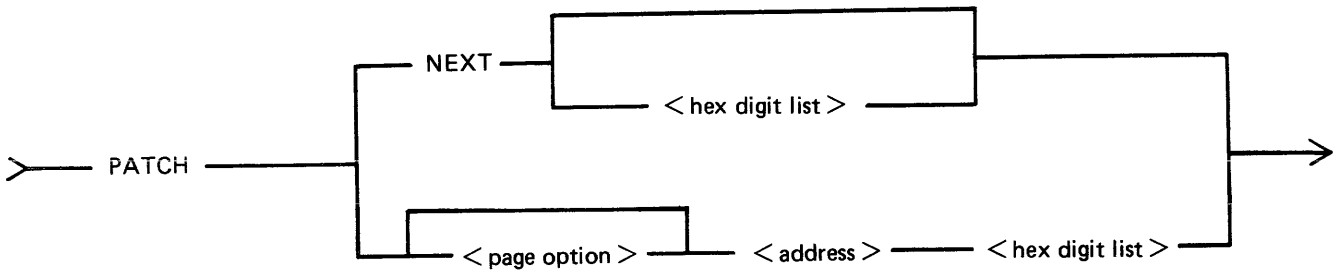
<display option>



<check option>



<patch option>



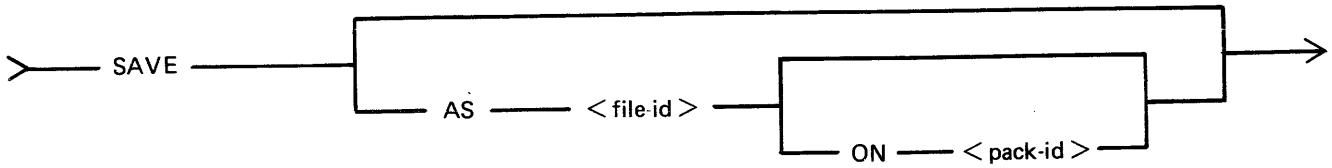
<address> = is a hexadecimal number of one to four digits.
<hex digit list> = is n hexadecimal digits, where n = 2.

<datacomm display option>

DIAGCBUF	WAITSPIM	VMG	VERSIONINFO
SPO.DATA	SCL	PSEUDO.REG.DUMP	SAT
PHDUMP	OPTIONS	INTERGLBL	GLBLM
DEBUG.GLOBAL	DEBUG.EXTRA	CURRENT.DICTIONARY	MIX
CT.INFO	REMOTE.SPO.GLOBAL	ROMDUMP	OL
LOGGING	NUMERICLIMITS	APPEND.STORE	

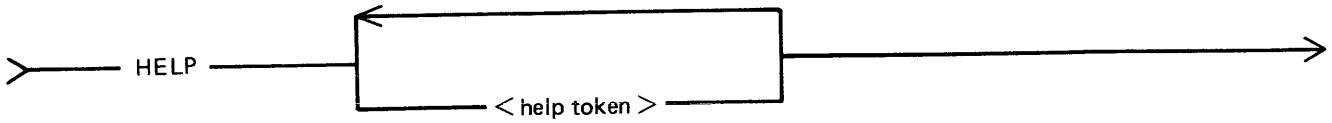
VMWA <page option>
PHT <device option>
HEX <hex option>

<save option>



<file-id> = valid CMS file identifier
<pack-id> = a valid CMS pack identifier

<help option>

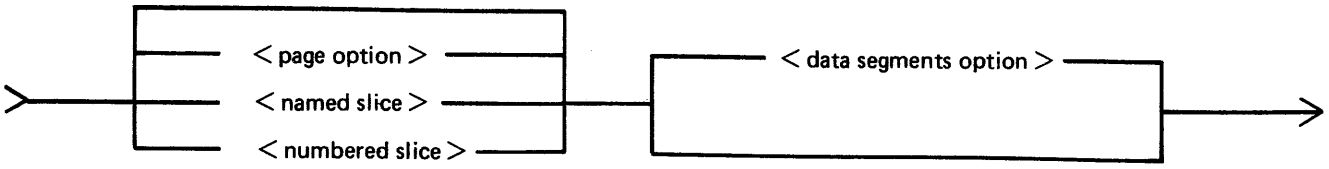


where <help token> =

ALL.MEMORY	GLOBAL	ESCT
DIAGNOSTICS	SAT	PHDUMP
C.TABLE	SCL	DUMP.CODE
TASK.TABLE	VMWA	GWA
SLICE	TASK	PHT
MEMORY.MAP	MEMORY.LINKS	MEMORY.FAULTS
MIX	OL	HEX
DATA.COMM.AREA	HELP	PRINT

DISPLAY	CHECK	SAVE
PATCH	END	BYE
PRINT.OPTION	DEVICE.OPTION	SLICE.OPTION
TASK.OPTION	WA.OPTION	DATA.SEGS.OPT
PAGE.OPTION	DISPLAY.OPTION	SHARED.CHAINS

<slice option>

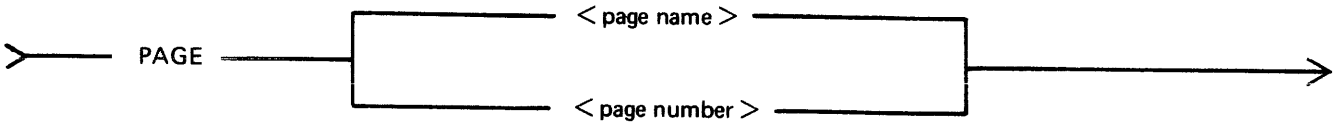


Where <named slice> =

DISKDDR	LPDDR	PANDDR	DSKSDI	MCHSN
SYSDISPSN	CASSDDR	SENDDR	ENDCRSN	MTPRSN
TICKRSN	KBDDR	DCCH	ADCDDR	SDCDDR
INITIALISE	RTCDDR	DCMESS	ICMDDDR	OPENCLOSE
CONBUFSN	SUSN	SPO	REMSPOSN	SCREENSN
SCLBUFSN				

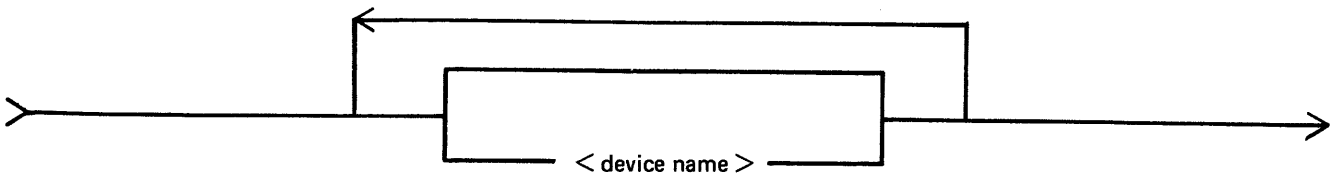
<numbered slice> = an integer in the range 0 – 123 inclusive.

<page option>



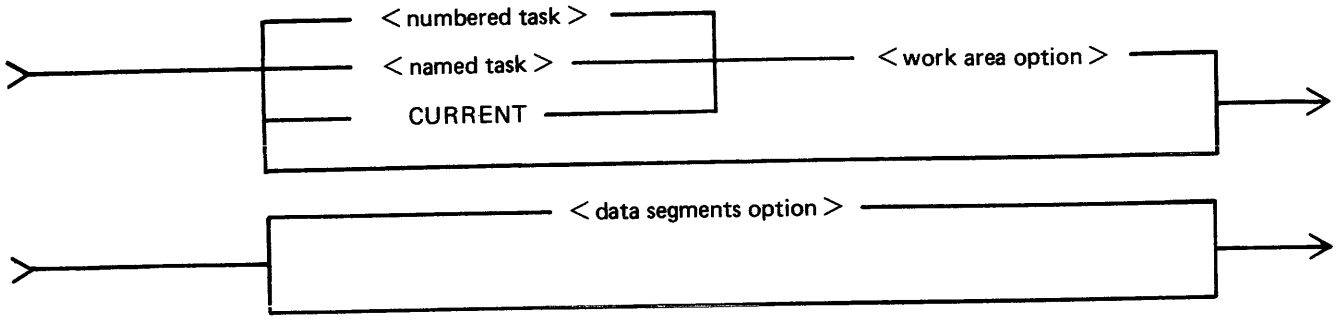
<page name> = ZERO/ONE/TWO/THREE/FOUR/FIVE/SIX/SEVEN
<page number> = a non-negative integer
<data segments option> = DATA.SEGMENTS/NO.SEGMENTS

<device option>



Where:
<device name> = SP LP CT DK DF MT RT SC DCP KB SDC ADC DM CX DI

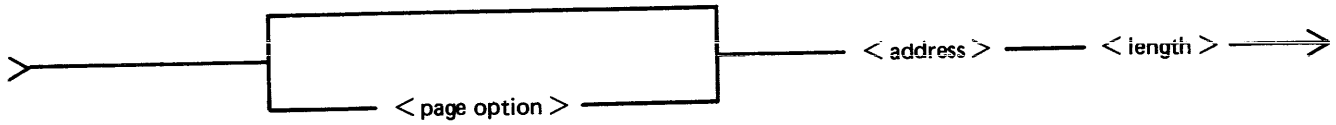
<task option>



Where:

<numbered task>	= an integer in the range 0-31 inclusive
<named task>	NDL MCS RSF BAILIFF SCL LOADER
<work area option>	= COBOL MPLII SORT BIL NDL RPG MCS

<hex option>



Where
<length> = a hexadecimal number of 1 to 4 digits.

ALL.MEMORY

This option creates analysis of fixed MCP data areas, configuration table and task table if present, all peripheral handler tables, memory links and analysis of all locked slices in memory plus all overlayable code and data segments present in memory at the time of the dump.

GLOBAL

This option creates an analysis of the fixed MCP data area. It includes the last code executed.

C.TABLE

This option gives a print of the configuration table, if it was in memory at the time when the dump was taken.

ESCT

This option gives a print of the mix table (execution scan priority table).

DIAGNOSTICS

This option gives a print of the MCP's diagnostic buffer area.

TASK.TABLE

This option gives further analysis of the mix, if the overlayable task table was present in memory at the time when the dump was taken.

MEMORY.MAP

This option provides an analysis of the layout of memory and may be output for any specified page.

MEMORY.LINKS

This option analyzes the layout of the overlayable area of memory and may be output for any specified page.

VMWA

This option gives a print of the virtual memory work area only.

GWA

This option gives a print of the global work area only.

SCL

This option prints the keyboard buffer only.

SAT

This option gives a print of the slice address table only.

MIX

This option gives a selective analysis of parts of the dump relating to the tasks running at the time of the dump.

OL

This option gives a selective analysis of peripheral configuration information.

DATA.COMM.AREA

This option prints the areas of memory relating to data communications.

PHT

This option gives a print of selected peripheral handler tables. If the <device-option>s are absent, then all peripherals attached to the system at the time of the memory dump are analyzed. Allowable values for <device-option> are:

CX	Channel expander
LP	Line printers
SP	Serial (console) printer
CT	Cassettes
DK	Cartridge disks
DF	Fixed disks
DM	BSM, BSMII disks
KB	Keyboard
SC	Screen
ADC	Asynchronous data comm controllers
SDC	Synchronous data comm controllers
DI	ICMDs
RT	Time-of-day clock

DCPP	Data communication power pak
MT	Magnetic tape

SLICE

This option provides selective printing of locked slices of memory or of data segments. It does not allow for associated segments to be output. The <slice-option> may be either a "slice-number" in the range 123, or one of the following names:

DISKDDR	RTCDDR
LPDDR	DCMESS
PANDDR	TICKRSN
KBDDR	SYSDISPSN
CASSDDR	DSKSDI
SENDDR	ENDCRSN
SCREENSN	MCHSN
SUSN	MTPRSN
INITIALIZE	ADSDDR
SDCDDR	OPENCLOSE
DCCH	SPO
CONBUFSN	SCLBUFSN
ICMDDDR	REMSPOSN

TASK

This option prints the contents of a task control block (TCB).

The <task-name> may be one of:

NDL
RSF
BAILIFF
SCL
LOADER

The <wa-option> may be one of:

MPLII
BIL
COBOL
RPG
SORT
NDL

HEX

This option provides a print (or display) in hexadecimal and byte format of selected pairs of memory. The <start address> is a four-hex-digit number and the <length> is also specified as a four-hex-digit number.

PHDUMP

This option causes the peripheral handling dump area to be output.

DUMP.CODE

This option will cause 1024 bytes of the code that was being executed at the time the machine was halted to be dumped to the printer (in hex format). This enables the user to check for code corruption.

MEMORY.FAULTS

This option will analyse memory links, all peripheral handler tables and all slices in memory, but will only output error messages for any inconsistencies encountered.

SHARED.CHAINS <page option>

This option will cause an analysis of any shared slices found on the page specified to be sent to the printer. If <page option> is omitted, all pages in the memory dump will be analysed.

<datacomm display option>

The following names of absolute maps can be displayed on the user's datacomm terminal:

DIAGCBUF	APPEND.SORE
WAITSPIM	VMG
VERSIONINFO	SPO.DATA
SCL(BUFFER)	SAT(M)
PSEUDO.REG.DUMP	PHDUMP
OPTIONS	INTERGLBL
GLBLM	DEBUG.GLOBAL
DEBUG.EXTRA	CURRENT.DICTIONARY
CT.INFO	REMOTE.SPO.GLOBAL
ROMDUMP	LOGGING
NUMERICLIMITS	

These maps cannot be displayed when PMB90 is running via console I/O or ACCEPT/DISPLAY. Also valid for display on the datacomm terminal are:

PHT	<device option>
VMWA	<page option>
MIX	
HEX	
OL	

HEX, MIX and OL are the only options that are also valid for display via console I/O and ACCEPT/DISPLAY.

NOTE

When submitting memory dumps for analysis, it is helpful if some preliminary analysis has already been performed. The following option is recommended:

```
PMB90
PRINT MIX OL MEMORY.MAP MEMORY.LINKS GLOBAL PHT TASK CURRENT
```

Always provide the MEMDUMP file on magnetic media even if this preliminary analysis has been performed.

Initiating PMB90

In a non-datacomm environment the initiating message is as defined previously in "Starting The Utility" (see railroad diagram).

Example:

PMB90 or
PMB90 AX

For interactive mode, the initiating message entered from the application page is:

<cc>RN PMB90

where <cc> is the control character used on the system. A GEMCOS/TMCS interface and a TD 830 like terminal is assumed.

If an attempted execution in non-RN mode takes place, that is, the user is attempting to attach to an already running PMB90, the user will be warned by PMB90 to use RN and he will be detached. Similarly, if the GEMCOS/TMCS interface is not set, then a warning will be displayed at the user's terminal by PMB90 and he will be detached.

Using The MERGE Option

A new command "MERGE" is used in the initiating message.

Example:

PMB90 MERGE or
PMB90 MERGE AX or
RN PMB90 MERGE

NOTE: Specification of any one of the commands:
GET <file-id>
ON <pack-id>
ON TAPE
in the initiating message along with MERGE, will be detected as an error by PMB90 and cause termination of the run.

Following a valid initiating message, PMB90 will prompt:

ENTER PACK ID AND FILE ID FOR MERGED FILE IN FORM
<pack-id>/<file-id>
A NULL REPLY WILL RESULT IN MERGED FILE CALLED 0000000/MERGEDUMP

NOTE: Null reply is one of the following:
1. A null accept if "AX" was specified in the initiating message or
2. <RETURN> if console input is being used ("AX" not specified in the initiating message, or
3. <RETURN> if PMB90 is running as a datacomm applications program.

PMB90 will attempt to create a merged dump file with the specified name but will not succeed if:
1. The file already exists (remove it, change its name or specify different name for merge).
2. The disk is not on line.

If the disk cannot be opened, PMB90 will again prompt for a merge file name with final ultimatum.

If the disk cannot be opened, PMB90 will again prompt for a merge file name with an ultimatum.

If the disk is opened successfully, PMB90 will prompt the user with:

```
LOAD AND READY DISK WITH MEMDUMP PART 1
ENTER PACK ID OF DISK
```

PMB90 will look for the specified disk and attempt to open a file MEMDUMP on it as part 1 of the split dump file. This will be copied over to the file specified earlier and PMB90 will then prompt:

```
LOAD AND READY DISK WITH MEMDUMP PART 2
ENTER PACK ID OF DISK
```

PMB90 again searches for a file MEMDUMP on the specified disk and this operation is repeated until all the split files of the dump have been processed.

PMB90 will then prompt:

```
MERGED DUMP FILE CREATED
INPUT DUMP OPTIONS REQUIRED
```

The user can now continue with the dump analysis in the usual way.

On completion of the PMB90 run, the merged file will be left on the disk and any subsequent analyses may be done using an initiating message of the form:

```
PMB90 AX GET MERGEDUMP
```

Early Merge Termination

The facility exists to stop the merge process and instruct PMB90 to commence analysis of the memory dump file assembled so far, once the first split file has been processed. After PMB90 has prompted:

```
LOAD AND READY DISK WITH MEMDUMP PART n [n ≥ 2]
ENTER MFID OF DISK
```

The user may enter MERGE.OVER to terminate the MERGE process.

Once MERGE is terminated, the PMB90 will then prompt:

```
MERGED DUMP FILE CREATED
INPUT DUMP OPTIONS REQUIRED
```

The user can now commence normal dump analysis.

NOTE:
If the user enters MERGE.OVER when PMB90 expects PART 1 of the dump file, this causes the following message to be displayed:

```
NO MEMDUMP FILE YET READ IN SO PMB90 CANNOT ANALYSE
```

Further prompts will request the user to:

LOAD AND READY DISK WITH MEMDUMP PART 1
as previously described.

NOTE

The user is advised that this facility should be regarded as for emergency use only, as the behaviour of PMB90 when analysing a truncated dump file is unpredictable. The only circumstances in which it might be reasonable to use this facility are if part of a dump was lost or unreadable due to corruption.

Alternatively a user may wish to be able to assemble part of a dump, for example, where he intends to analyse something known to reside in page 0 of memory and which is therefore near to the beginning of the dump.

NOTE

Under no circumstances will PMB90 allow a user to skip a split file, that is, follow a PARTn with PART n + 2.

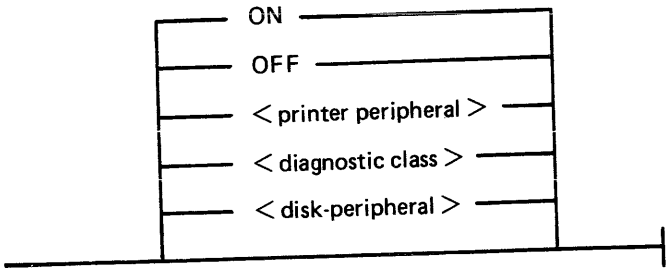
B 90 DEPENDENT INTRINSICS

The following pages describe intrinsics which are included in the B 90 MCP only.

GT (General Trace)

The general trace command is an MCP intrinsic which displays various diagnostic information either on the system console or on the line printer. The general format is:

Format:



To turn the trace printing on, with printing on the console (SPA) enter:

```
GT ON
```

To direct the diagnostic printing to a line printer, specify the printer peripheral. For example:

```
GT LPA
```

Note that the trace must also be turned ON in this case. The trace will be interleaved with any program printout.

To turn the trace off, enter:

```
GT OFF
```

To display the cumulative number of retries performed on a disk unit since the last warmstart, specify the desired peripheral. For example:

```
GT DMA
```

The general B 90 machine code trace is implemented as follows:

Each trace point is identified by a diagnostic class and a diagnostic value. The class is one of 16, identified by a hex digit (0-F). This identifies the system function being performed, as follows:

0	Open/close file handling routines
1	Indexed file handling
2	Accept/display routines
3-7	Intrinsic functions (for example, SORTINTRINS)
8	Automatic volume recognition (AVR) routines
9	BAILIFF (task handling MCP routine)
A	Disk space allocate/deallocate routines
B	Interpreters (BILINTERPX, COBOLINTX, NDL.INTERPX)
C	MCP communicate handler (MCH)
D	Virtual memory (VM) routines
E	Task control routines (for example, BAILIFF)
F	I/O master interrupt handler (MIP)

The diagnostic value is a hex number (0-F) giving a measure of depth of trace required (0 is least significant, F is critical).

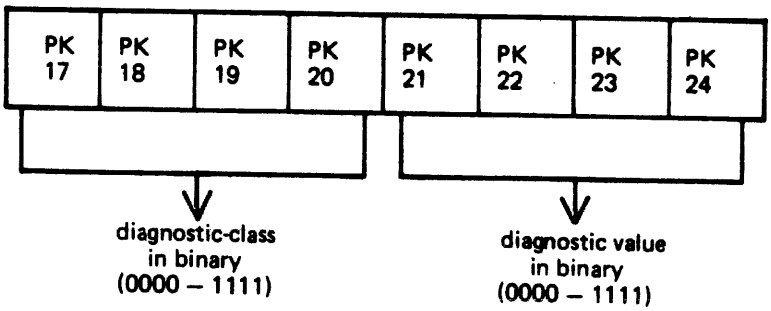
The GT command allows the storage of a switch-value for each diagnostic class. If the trace has been initiated via the GT ON command, then each time a trace point is encountered in the machine code, a diagnostic printout will occur if the switch-value for that class of trace is lower than or equal to the diagnostic value of that trace point. For example, if a particular trace point had a diagnostic value of C, then a trace point would occur if the switch-value for the appropriate class was in the range 0-C inclusive. If the switch-value set by the GT command was D, E or F, no trace print would occur.

The default values set at warmstart are:

trace option OFF
diagnostic print directed to SPA
all switch-values set to F.

Therefore the only trace prints that will normally occur are the critical ones (diagnostic value F).

Note that if the trace option is ON, the PK lights display the current trace point, whether or not the printing takes place, as follows:



For example, a trace point with class D (binary 1101) and severity 7 (binary 0111) will cause PK lights 17, 18, 20, 22, 23 and 24 to be lit.

The switch-value entered in the GT command must contain two hex digits:

first digit – switch-value for register diagnostics
second digit – switch-value for memory diagnostics

If no options are specified for the GT command, then the current switch-values for each of the 16 diagnostic classes are displayed.

Examples:

To find the retry count for DMA and DKA:

```
GT DMA
   DMA          0     RETRIES
GT DKA
GT DKA INVALID – DEVICE NOT ON SYSTEM
ENTER "OL" FOR MORE DETAILS
```

To set the switch-values to CC for interpreters, then interrogate all switch-values, then turn on printing to LPA:

```
GT B CC
GT
DIAGNOSTIC LEVELS SET ARE FFFFFFFF FFFFFFFF
                          FFFFFFCC FFFFFFFF

GT LP
GT ON
```

Note that only the disk peripheral option and the GT command with no further options result in an immediate response to the operator.

Format of Diagnostic Printout

The format of a register diagnostic message is given here, where each X represents a single hexadecimal digit. The hex string is printed on one line.

register:	AD	BO	B1FL	J	K	L
	xx	xx	xxxx	xxxx	xxxx	xxxx

register	M1	M2	WR	X	Y	MXA	MXB	UMRX	AD,ESCT
	xxxx	xxxx	xxxx	xxxxxxxx	xxxxxxxx	xxxx	xxxx	xxxx	xxxx

The diagnostic class and value are given by the AD register (first 2 digits, repeated in first pair of last four digits). For example, a DF diagnostic with M1 = 1111 would look like:

DF	xx	xxxx	xxxx	xxxx	xxxx	xxxx									
				1111	xxxx	xxxx	xxxxxxxx	xxxxxxxx	xxxx	xxxx	xxxx	DF	xx

where x indicates any hexadecimal digit.

When GT is switched on, system fatal errors (which would normally result in a set of PK lights 17 to 24 flashing) are reported as a diagnostic printed message followed by initialization to the Initial state. A memory dump should normally by taken to find more information. When GT is switched off, the pattern of PK lights 17 to 24 that are set flashing on a system fatal error correspond to the value of the AD register.

ENHANCEMENTS TO LOGGING

The SPO feature on B 90 systems allows more than one SPO device to be used on the system. This creates a problem in the routing of SCL responses. To ensure that SCL responses go to the device which solicited the output, the system uses the fields ODT ID and ODT TAG. These entries are now included with any system message log entry recorded within the log files. They are used to identify which type of device and, in the case of multiple terminal SPOs (MTS), which station issued or received the logged SCL message.

The ODT ID field is the field which identifies the SPO device type. The user may now determine whether the SCL was to or from a console type SPO, a program SPO (SPIM) or an MTS terminal SPO.

ODT ID	Issuing or Receiving Device Type	ODT TAG
@FF@	Initial (zipped from SYSCONFIG)	@FFFF@
@00@	Local (Console or Self Scan)	@0000@
@01@	MTS Terminal	Logical Station Number
@70@	SPIM Terminal	@0000@

The ODT TAG field provides the user with information to allow the identification of which MTS terminal issued or received the SCL. The ODT TAG should be regarded as a 16 bit value and is split into two separate sections. The first section, bits 0 to 4, has been reserved for future expansion. Bits 4 to 15, the second section, provide the Logical Station Number (LSN) of the SCL device which issued or received the message. The value of the LSN is in hexadecimal notation, thus an ODT TAG value of hex 0028 would refer to station 40 within the network.

Variations To PL And IR Caused By MTS

When SCL is input, or if a data comm task is executed from a terminal which is the controlling SPO, any SCL output produced will be directed only to that SPO. This is altered if the terminal is not the controlling SPO. In the case of a terminal which is not a controlling SPO, the SCL is routed both to the SPO page of the terminal and to the controlling SPO, if present. A similar type of system is used when any attempt is made to view the contents of the message log files.

When the PL utility is used to print out the message log files, the resulting output depends on which SPO initiated the utility. If the utility was started from the controlling SPO, the entire content of the message log within the given parameters is printed. If the utility was initiated from a non controlling SPO device, a selective printout is output giving only the SCL related to that particular SPO.

The use of the IR, LB and LF functions of SYS-SUPERUTL has been changed to work in a manner similar to PL. The messages displayed on a non-controlling SPO are only those associated with that particular terminal. A terminal which is the controlling SPO would have all required SCL messages recalled and displayed.

SECTION 9
B 900/CP 9500 DEPENDENT SYSTEM SOFTWARE

INTRODUCTION

This section discusses those functions of CMS that are unique to the B 900/CP 9500 systems.

The discussion is intended for two types of user.

1. Systems operators who wish to know how things work.
2. Software support personnel who need more technical information.

To accommodate these two audiences, this section is divided into subsections. Operational descriptions of procedures are given first. Functional descriptions follow.

In addition, at the end of this section there is a subsection titled Utilities and Intrinsics. These utilities and intrinsics are either unique to this system or have some feature that is unique to these systems. For a discussion of other utilities and intrinsics, refer to the earlier sections of this manual.

The descriptions of these features are written in such a way that both system operators and support personnel can find the information they need.

CONTROL PANEL

The first step in learning how to operate the computer system is to become familiar with the controls on the central processor. These are shown in figure 9-1.

The hexadecimal keypad at the left is usually covered. It is used to run maintenance test routines (MTRs), to take ROM-generated dumps (explained later in this section) and to run COLDSTART functions when no local SPO exists.

At the upper right of the control panel is a series of switches. Some design levels have more switches than others. These switches are used to perform coldstart functions, take a memory dump and pass control to backup processors.

At the left is a set of lighted displays that flashes on and off when the system is operating. The numbers shown are significant. Most indicate normal operation or are for maintenance purposes. The Functional Description subsection discusses these numbers in detail. The time is displayed in the bottom set of lights.

Data Communications Transmit and Receive Indicators

These four pairs of lights are used by the B 900/CP 9500 to indicate when it is transmitting to, or receiving from, a terminal on a particular data communications line.

Data Communications Group Select Switch

This seven-position thumbwheel switch is used to select a four-line group of data communications lines for monitoring on the four pairs of transmit and receive indicators.

OPERATOR ATTENDED/UNATTENDED

Systems are described as either:
Operator attended, or
Operator unattended.

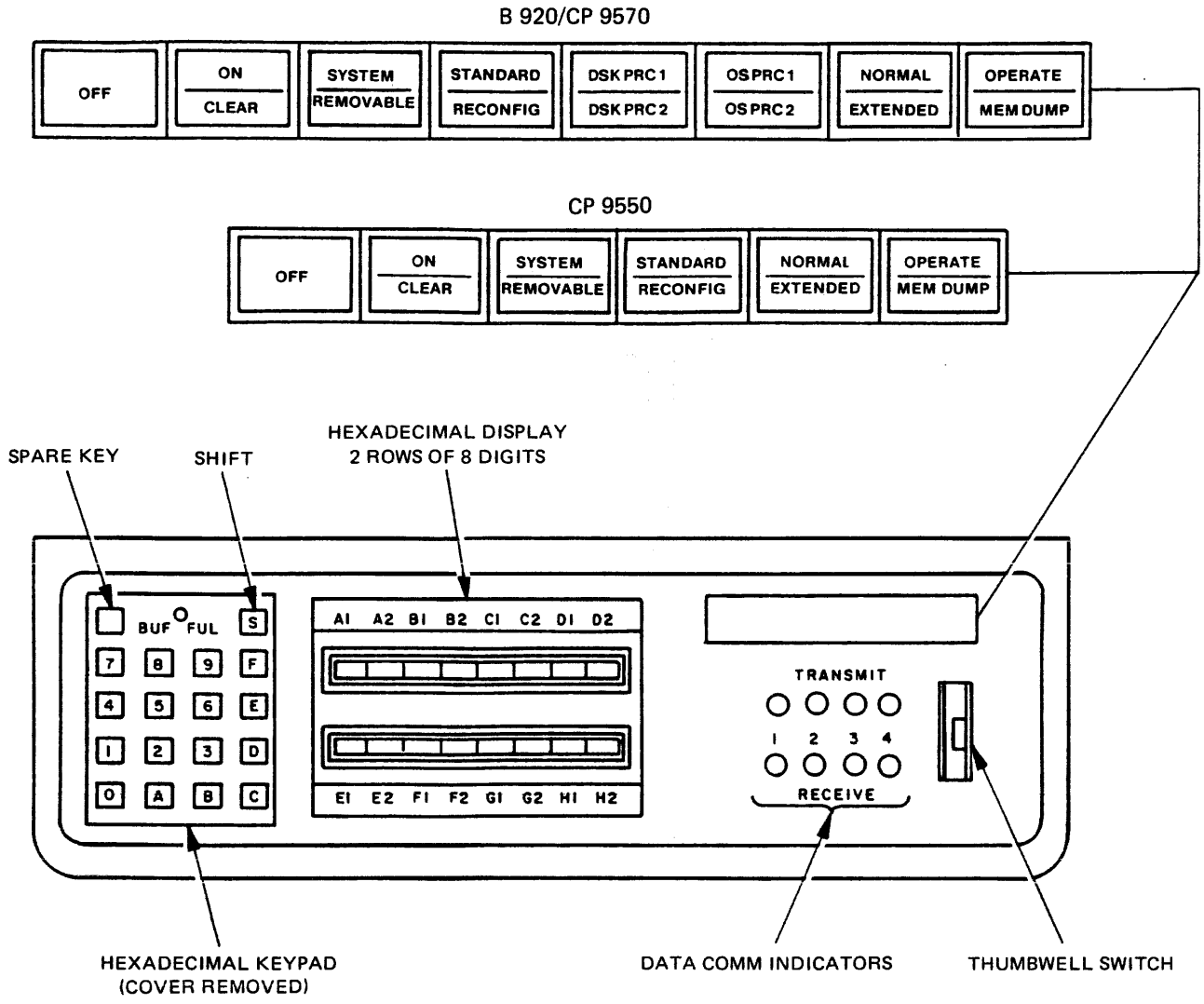


Figure 9-1. Control Panel

In operator attended mode, a Supervisory Printer Output (SPO) is part of the system. This SPO looks very much like a typewriter attached to a television screen. An operator types information on the keyboard to communicate with the processor. The numbers and words typed on the keyboard appear on the screen. The operator pushes a button on the keyboard to transmit the information on the screen to the system, and responses from the computer appear on the screen.

A SPO is shown in figure 9-2.

In operator unattended mode, there is no SPO. The operator starts the system by pushing the appropriate buttons on the control panel. The entering of commands in this mode is handled by a terminal on the data communications system which has been especially designated for this work. This terminal is called a Remote SPO.

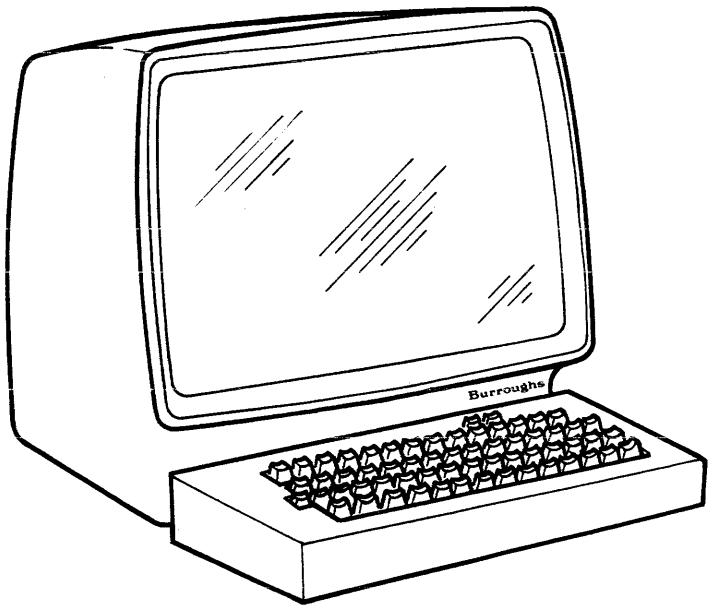


Figure 9-2. SPO

For the Remote SPO to function, a program called SPO Interface Module (SPIM) must be present on the system. The operations of the SPIM program are beyond the scope of this manual. However, this manual does explain WARMSTART/COLDSTART in unattended mode.

With the 3.05 release, the functions of SPIM can be handled by the Enhanced Terminal SPO (ETS) feature (explained later in this section).

The rest of this section focuses on the operation of systems where a SPO is present.

SYSTEM START-UP

In starting the system, the operator should understand that there are two different kinds of on and off. There is a physical on and off where the actual equipment is involved and there is logical on and off where the internal processing system is made ready to begin work or to shut down.

In a typical day, a system operator turns the physical system on (power-on); brings up the software; does whatever work is scheduled; logically powers-down and then physically shuts off the equipment (figure 9-3).

For more specific information on the powering on and off procedure, refer to the B 900/CP 9500 Operators Manual, form 1166360.

Physical Power-On

Each piece of equipment has an ON and OFF switch.

The most important thing to remember when turning on equipment, is to do so in the appropriate order:

1. Turn on the processor
2. Turn on the SPO (if the system has one)
3. Turn on the fixed disk unit (if present)

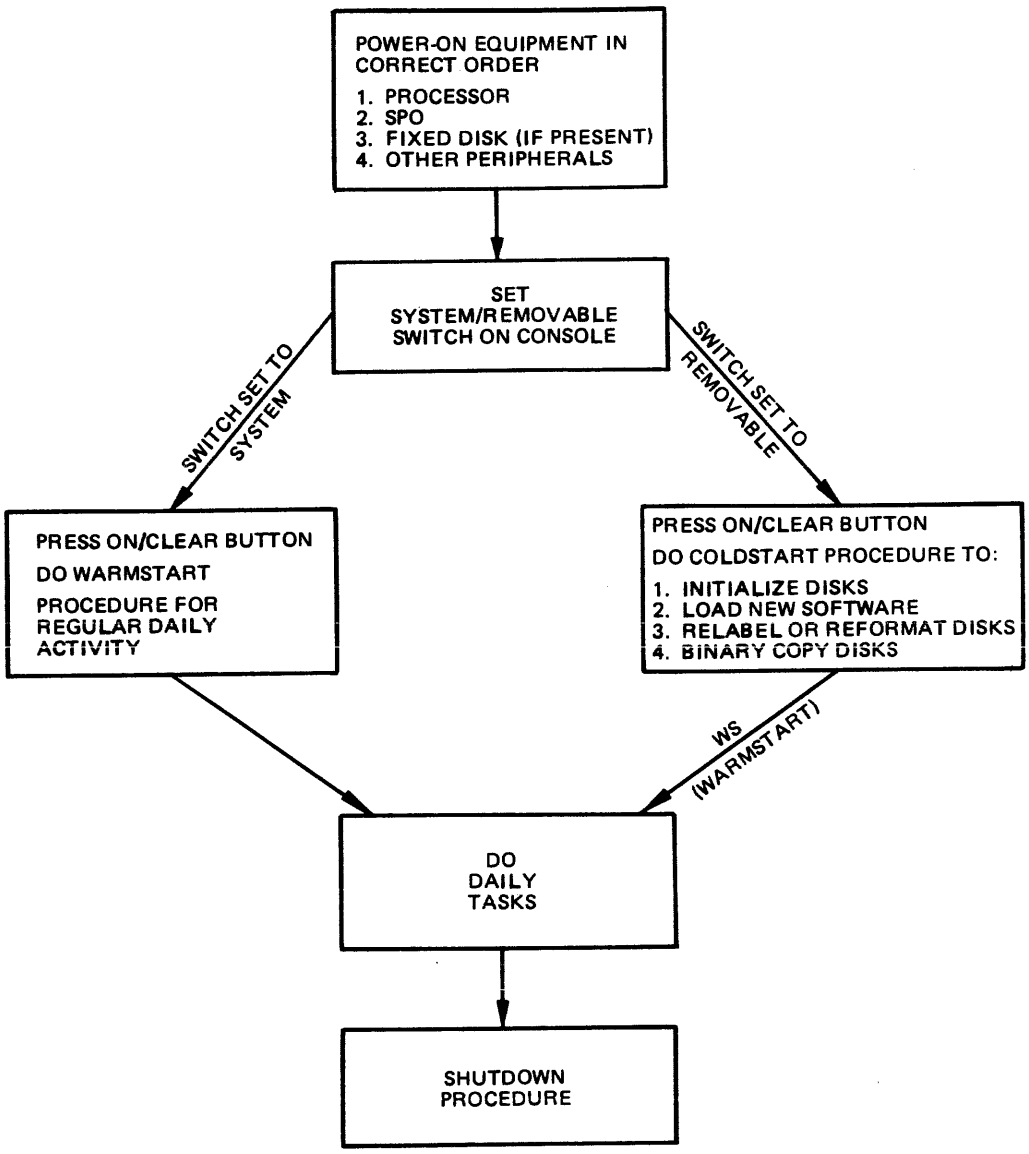


Figure 9-3. Block Diagram of Operating Steps in Daily Running of B 900/CP 9500

4. Turn on the other peripherals, for example, removable disk units (including PACK and the PACK controller) and printers.

Equipment is turned off in the reverse order after the logical power off.

WARMSTART

In normal day-to-day system operation, warmstart is the way the system is made ready to do work (logically ready). The warmstart instructions assume the hardware has already been turned on in the correct order.

System With SPO

In a system that has a SPO, the warmstart procedure is as follows:

1. Set the SYSTEM/REMOVABLE switch on the processor to SYSTEM.
2. Press the ON/CLEAR button.
   Some time passes.

Pressing ON/CLEAR causes the system to display a message on the SPO which gives the version of the Burroughs Master Control Program, lists the equipment that is on-line and requests the correct date and time.

A sample message is:

```
BURROUGHS CMS MCP 3.05.00
DFC SYS1/0 FILES OPEN
DF =SYS DISK
DFD SYS2/0 FILES OPEN
ENTER DATE AND TIME USING DT COMMAND
COMMENCING LOG FILE CONSOLIDATION
```

3. The operator should type in the command DT, the correct date using numbers separated by slashes and the time with no punctuation.

   Example
   DT 11/14/84 0920

   Leading zeros are needed. For example, February must be entered as 02; 9 a.m. as 0900.

The system does not accept any other format for entering the date and time. After this information has been typed in correctly, press the XMT button to transmit.

At this point the computer does internal work. A short period of time passes and a series of numbers flashes on and off on the display panel. When the internal processing is complete, a message appears on the display screen similar to this:

```
14 Nov 84 84315* 1020
TRANSFER COMPLETED
COMMENCING LOG FILE REALLOCATION
LOGGING IS INITIATED ON 11/14/84 at 10:22:32 (MCP VERSION 03.05.00)
```

* Julian date

When the MCP version message appears, the system is warmstarted.

Warmstart Variations

1. If the system has no SPO, or if the NORMAL/EXTENDED switch is set to EXTENDED, the system uses the default values of 11/11/11 for the date and 24:00:00 for the time. The warmstart procedure is entered automatically if the SYSTEM/CONFIGURATION switch is set to SYSTEM and the ON/CLEAR button is pressed.

2. If there is no fixed disk unit or if the fixed disk unit is not to be powered on, a disk with the required system files (the system disk) must be inserted in the removable disk drive designated as the loader channel. The rest of the procedure remains the same.

Coping With Problems

If something has gone wrong with the warmstart procedure, the simplest method of correction is to turn off the equipment and start again.

Some common things that may cause the warmstart to fail are:

Disk unit not turned on
Removable disk is disabled
Something was typed incorrectly
You forgot to transmit, or you hit the wrong button when attempting to transmit.
Required system files are missing or corrupt.

If restarting does not correct the problem, consult the lighted numbers on the display panel. The numbers constitute an error message. The meanings of these messages and suggested ways of correcting problems are given under Functional Description later in this section.

COLDSTART

The COLDSTART procedure is used by both system operators and field engineers.

There are several reasons for an operator to do a COLDSTART. For example:

1. New disks need to be initialized (made ready for use).
2. The computer is new and the system files must be loaded for the first time.
3. Burroughs has made improvements to the software and a new level of system software must be entered.
4. The operator wishes to relabel, reformat a disk or do a binary copy.

There are other uses for COLDSTART, including:

1. Initializing maintenance test routine (MTR) disks (the FE command).
2. Installing additional fixed disk units to the system.

The last two uses are explained in the COLDSTART Functional Description subsection as they are more likely to be used by field engineers than by the system operator.

There are three types of COLDSTART operation:

1. Operator attended (SPO is present).
2. Operator attended (no SPO is present, system options are entered through keypad).
3. Operator unattended (there is no SPO and system options are selected by switches).

Operator attended mode is explained first, followed by a brief explanation of unattended operation.

COLDSTART Initiation

Procedure

The first steps of the COLDSTART procedure are much the same for any COLDSTART activity. The hardware should be turned on.

1. Insert the B900RL1 disk in the loader device (write disabled).

2. Set the SYSTEM/REMOVABLE switch to REMOVABLE and put the other switches in their primary positions (STANDARD, NORMAL, OPERATE, DISK PROCR1 and OSPROC1).

3. Press ON/CLEAR button on processor. Wait a few minutes.

COLDSTART Attended Mode – SPO Present

Carry out steps 1 to 3 of the initiation procedure then continue as follows.

The system displays a message on the SPO screen giving the version number of the MCP operating the system and requesting the date.

Example:

```
BOJ COLDSTART 03.05.00
ENTER DATA <MM/DD/YY>
(BOJ stands for Beginning-Of-Job)
```

4. Enter the date in the style requested. Leading zeros must be included.

   Example:

   ```
   01/04/84
   ```

   The system responds by telling the operator what disk drives are on-line. Example:

   ```
   DMA:  B900RL1
   DFD:  0000000
   END OL
   ```

5. After the system responds with the following message, the operator can enter one of the functions from table 9-1.

   ```
   ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS>
   NOTE:  TO PRINT A BRIEF EXPLANATION IN RESPONSE TO ANY PROMPT, ENTER
   "?EX"
   ```

Table 9-1. COLDSTART Functions

BC	Binary Copy: duplication of removable disks.
?DS	Discontinue: terminates the following functions: BC, FE, IN, LD, RL, RF, RP.
?EX	Explain: provides an explanation when COLDSTART expects a response.
FE	Field Engineering: test function used by Burroughs engineers.
IN	Initialize: initializes a disk.
LD	Load: loads files from release disk to system disk.
OL	On-Line: lists status of drives on system.
RF	Reformat: clears information from disk but does not check for bad sectors.
RL	Relabel: relabels a disk.
RP	Replace: replaces system files on the system disk with the files from B900RL1. No checks are performed.
WS	Warmstart: warmstarts the system.

BC (Binary Copy)

BC allows rapid duplication of removable disk media that are the same type and size. The function copies all sectors from one CMS removable disk, called the source disk, to a second CMS removable disk, called the target disk. The function preserves the BOOTSTRAP portion of track zero on the target disk, as well as some labelling information. The three phases of the binary copy function are disk selection, copy and verification.

NOTE

The function does not update the pack-id of the data file part of indexed pairs. Therefore, if an indexed pair is copied from disk A to disk B, the keyfile still contains the data file pack-id as "A". To avoid this situation, the target should be given the same pack-id as the source disk.

Procedure

The NORMAL/EXTENDED switch should be set to NORMAL. If a local SPO is connected to the system, binary copy will be prompted from the SPO keyboard, and its messages will be displayed on the SPO screen. If no local SPO exists, the function will be prompted from the keypad and will display its status in the indicator display blanks next to the keypad.

SPO – Attended Operation

1. Enter BC when the message ENTER FUNCTION appears.

2. Enter the drive-id of the target disk.
   Display – ENTER DRIVE-ID OF TARGET DISK <3 characters such as "DKA">
   ENTER – <drive-id> or ?EX for explanation

3. Enter pack-id to be inserted into the label of the target disk. The disk-name consists of from one to seven characters (A-Z, 0-9, . –)
   Display – ENTER DISK-NAME TO BE ASSIGNED TO THE TARGET DISK
   Enter – <disk-name> or ?EX for explanation

4. Enter drive-id of the source disk.
   Display – ENTER DRIVE-ID OF SOURCE DISK <3 CHARACTERS SUCH AS "DKA">
   Enter – <drive-id>

For an explanation of the error messages which may occur during the disk selection phase, refer to table 9-2.

COLDSTART determines that the source and the target disks are the same type and the copy begins. If any block of 32 sectors on the source disk cannot be read, the block will be replaced with binary zeros on the target disk. If the function is unable to write to the target disk, the function will abort. Refer to table 9-3 for messages displayed during the copy phase.

After the copy is completed, all sectors on the target disk are read to report all read parity errors encountered. This phase guarantees the integrity of the copied data. Refer to table 9-4 for messages displayed during the verification phase.

NOTE

The verification phase checks the target disk for parity errors only. No "compare" between the source disk data and the copied target disk data is performed.

Table 9-2. Error Messages of Disk Selection Phase of Binary Copy

Message	Cause
Invalid input – invalid disk type specifier detected	A removable disk type cartridge (K), mini (M), or pack (P), was not specified for target or source disk.
Invalid input – invalid disk drive specifier detected	Specified disk drive not in range from A to P.
DXU: Not ready	Specified target or source disk was not ready.
Invalid input – one or more non-blank characters are expected	The pack-id transmitted had a blank character imbedded in the input.
Invalid input – too many characters in disk name	Pack-id transmitted exceeded the maximum length of seven characters.
Invalid input – invalid character in disk-name	Pack-id transmitted contained character other than (A-Z, 0-9, .–).
Invalid input – source and target may not be the same disk	Target and source disk provided referenced the same disk.
Warning – source and target disks have the same name	For information only – does not abort binary copy. The disk name supplied for the target disk was the same as the source disk name.
Error – source and target disks are not the same type	Target and source disks were not the same type and size.
End BC	Function complete.

Table 9-3. Error Messages Displayed During Copy Phase of Binary Copy

Message	Cause
Binary copy from <source mnemonic> to <target mnemonic>	Information only – indicates the disk to disk copy has begun.
Track zero on <disk mnemonic> has a bad sector. Binary Copy aborted	One of the first 32 sectors on the target disk has a bad sector. The copy is aborted.
Output error on target disk – try another disk. Binary Copy aborted.	The target disk could not be written because of a disk error. Another copy could be attempted with a different disk.
Transient input error in <start address> to <end address>	When reading a block of 32 sectors, an error occurred that could not be isolated to the exact sector when reading each sector or source individually.
Input errors occurred on these sectors:	This message heads a list of hexadecimal sector addresses which had errors and were copied as binary zeros.
<number> source disk input errors encountered	This is the number of sector addresses listed, as in preceding message.

Table 9-4. Error Messages of Verification Phase of Binary Copy

Message	Cause
Checking target disk for input errors	Information only – indicates the end of the copy phase and the beginning of the verification phase.
Transient input error from <start address> to <end address>	When reading a block of 32 sectors, an error occurred that could not be isolated to the exact sector when reading each sector of the target individually.
Input errors occurred in these sectors:	This message heads a list of hexadecimal sector addresses on the target disk which contained errors.
<number> target disk input errors encountered	The number of sector addresses listed, as in preceding message.
Read check on target disk complete.	Information only – indicates the verification phase is complete.
<source mnemonic> copied to <target mnemonic> as <target pack-id>	Information only – indicates the binary copy function is complete.

?DS

This command can be entered to "discontinue"any of the COLDSTART operations. To use it, enter ?DS in place of any response that is expected to contain three or more characters.

FUNCTION DS-ED
appears on the SPO.

FE

This is a command generally used by field engineers (FE) to initialize disks which are used for maintenance test routines. (For an explanation of its use, refer to the COLDSTART Functional Description subsection.)

IN (Initializing Disks)

When new disks are to be used for the first time, they must be initialized. Disks are also initialized when they have been used for a time and the information they contain is no longer needed. Essentially, this is re-initialization, but the procedure is still termed "initialize".

Initialize does five main things.

1. Makes the disk a B 900/CP 9500 disk.

2. Sets up a disk directory. A directory is the first thing physically encountered on a disk and contains a record of every file on the disk, its name, size and location. When a disk is initialized, the operator decides how many files the disk may have.

3. Gives the disk a name, also frequently called a label.

4. Checks the disk for bad sectors (sections). The number of bad sectors is reported to the operator. These portions of the disk are subsequently not used. Over long periods of time, sectors of a disk may become damaged. For example, if 50 or more sectors of a disk cartridge are no longer usable, it is wise to discard the disk.

5. Erases all information previously on a disk.

Procedure

Follow steps 1 through 5 under COLDSTART Initiation, then proceed with step 6 as follows:

6. When the message ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS> appears, put the disk to be initialized (often called a scratch disk) in an empty drive. Enter:
   IN
   at the SPO and transmit.
   The system responds with
   ENTER DRIVE-ID OF DISK <3 CHARACTERS, SUCH AS "DKA">

7. Enter the three-character drive name of the drive where you placed the disk which is to be initialized.
   Example:
   DMA
   The system response is
   VALUES FOR DISK-NAME, OWNER'S ID, SERIAL NUMBER AND MAX-NUMBER-OF-FILES-ON-DISK MUST NOW BE ASSIGNED. DO YOU WISH TO ASSIGN THESE VALUES YOURSELF OR HAVE THE SYSTEM DO IT FOR YOU?
   ENTER "SELF" OR "SYSTEM"

8. If you enter SYSTEM, all further information needed by initialize is derived from the default values. Refer to the Default COLDSTART Assignment subsection for a list of default options. The system automatically completes the initialization and displays the message:
   END IN

9. If you enter SELF, the system responds with:
   ENTER DISK-NAME

10. Enter the name of the disk being initialized. The name is selected by the operator and may be up to seven alphanumeric characters long, beginning with a letter.
    The system responds with:
    ENTER SERIAL-NUMBER

11. Enter a serial number up to six digits long. There must be some response to this request, so a six-digit number chosen at random may be entered if it is not necessary to keep records of disk serial numbers.
    The system responds with:
    ENTER OWNER'S-ID

12. The owner's-id may be 1 to 14 characters long. It is usual to enter either a company name, a department name, or the name of an individual.
    The system responds with:
    ENTER MAX-NUMBER-OF-FILES-ON-DISK

13. Enter the number of files that is to be the maximum permitted on this disk. The possible range is from 1 to 2,804. For general use, a number between 250 and 300 inclusive is a good number to select for mini disks and cartridges. Between 1000 and 2804 is preferred for fixed disks and disk packs. The number you choose depends on how large you think your files will be. If you think you will be entering a large number of very short files, you may wish to make this number larger.
    When this information is received, the system proceeds to complete initialization.
    The message:
    END IN
    ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS>
    appears.

To initialize another disk, repeat the IN procedure.

Initializing Fixed Disks

The fixed disk system is initialized if the system is new, a new level of software is to be loaded onto the system, or when additional fixed disk units are being added to the system.

The procedure is as already described, using DF= for the drive-id.

LD (Load)

New Software Received

The procedure for installing new software is handled through COLDSTART. The tasks are done in the following order:

1. Initialize or reformat the system disk(s).
2. Load the new release using LD.
3. Warmstart

Procedure

To initialize the disks on the system, follow the steps given for initialization (IN). Use the new B900RL1 disk to do the COLDSTART.

The initialization of the disks is finished when the message END IN or END RF appears, followed by ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS>. Proceed as follows:

1. Enter
   LD
   and transmit. The system responds to LD with the message:
   ENTER DISK-NAME OF DISK YOU WISH TO LOAD FILES TO

2. The disk name may be one to seven alphanumeric characters long, beginning with a letter. SYSPACK is the disk name for the fixed disk.
   The system then begins loading the system files. Each file is listed as it is loaded. It is possible that the information being loaded is on more than one disk. If so, they are labelled in sequence: B900RL1, B900RL2, B900RL3. When all the information on one disk has been loaded, the system issues the following message:

   THERE IS NO DISK ON-LINE NAMED B900RL2
   AT THIS POINT, YOU MAY EITHER BRING THE MISSING RELEASE DISK ON-LINE AND CONTINUE PROCESSING, OR YOU MAY EXIT THE LOAD FUNCTION
   ENTER "CONTINUE" OR "EXIT"

3. If there is more than one release disk, remove the first one, place the next one in the drive, type CONTINUE and transmit.
   If there is only one disk or if you have entered the last disk in a series, type EXIT and transmit.
   If all of the system files have not been loaded, LD will issue a warning message.
   The remaining system files may have to be loaded individually.
   The successful completion of LD issues the message:
   END LD

NOTE

This function is intended to be used to transfer files from unmodified B 900 release media onto a system disk. Any other use could give unpredictable results.

OL (On Line)

This command lists the disks that are on-line. It functions in the same way as the OL command does during regular system operation.

Example:

```
OL
```

is entered. The system responds with:

```
DMB:  ACCOUNT
DFC:  THISDSK
DKG:  PACK123

END OL
```

This means:
A Burroughs super mini disk named ACCOUNT is ready on drive B.
A fixed disk named THISDSK is ready on drive C.
A cartridge disk named PACK123 is ready on cartridge drive G.

The system requests the next function:

```
ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS>
```

RF (Reformat)

The Reformat (RF) COLDSTART command clears a disk (fixed or removable).

RF is faster than IN. RF erases the disk directory, which is why it is so rapid. Old files are still present, but cannot be accessed and will be overwritten by new information.) This disk must have been initialized previously.

However, the RF command does not check for bad sectors. If this command is used, the number of bad sectors reported will be the same as the number which was reported the last time the disk was initialized. Consequently, it is recommended that the CHECK.DISK command is run against any disks which are reformatted.

CHECK.DISK will list all bad sectors. These bad sectors can then be removed using the XD (Delete Bad Disk Sectors) utility.

If this is not done and any bad sectors are present, information may be written to these bad sectors and subsequently lost.

Procedure

Follow steps one to five under COLDSTART Initiation, then proceed with step 6 as follows:

6. When the message ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS> appears, put the disk to be reformatted in an empty drive. Enter:
   RF
   at the SPO and transmit.
   The system responds with
   ENTER DRIVE-ID OF DISK <3 CHARACTERS, SUCH AS "DKA">

7. Enter the three-character drive name of the drive where you placed the disk which is to be reformatted.
   Example:
   DMA
   The system response is:
   VALUES FOR DISK-NAME, OWNER'S-ID, SERIAL NUMBER AND MAX-NUMBER-OF-FILES-ON-DISK MUST NOW BE ASSIGNED. DO YOU WISH TO ASSIGN THESE VALUES YOURSELF OR HAVE THE SYSTEM DO IT FOR YOU?
   ENTER "SELF" OR "SYSTEM"

8. If you enter SYSTEM, all further information needed by reformat is derived from the default values. Refer to the Default COLDSTART Assignment subsection for a list of default options. The system automatically completes the reformat, and displays the message:
   END RF

9. If you enter SELF, the system responds with:
   ENTER DISK-NAME

10. Enter the name of the disk being reformatted. The name is selected by the operator, and may be up to seven alphanumeric characters long, beginning with a letter.
    The system responds with:
    ENTER SERIAL-NUMBER

11. Enter a serial number up to six digits long. There must be some response to this request, so a six-digit number chosen at random may be entered if no records are kept of disk serial numbers.
    The system responds with:
    ENTER OWNER'S-ID

12. The owner's-id may be 1 to 14 characters long. It is usual to enter either a company name, a department name, or the name of an individual.
    The system responds with:
    ENTER MAX-NUMBER-OF-FILES-ON-DISK

13. Enter the number of files to be permitted on the disk, in the range 1 to 2804. Generally 250-300 is selected for mini disks, and 1000-2804 is chosen for fixed disks.
    When this information is received, the system proceeds to complete the reformat.
    The message:
    END RF
    ENTER FUNCTION <BC, FE, IN, LD, OL, RF, FL, RP, WS>
    appears.

NOTE

When reformatting a fixed disk the system does not ask for serial number and owner's-id. Number of files will be requested for each fixed disk assemblage. The drive-id for a fixed disk is "DF=".

RL (ReLabel)

This function is used to relabel (retitle) removable disks, fixed disks and pseudo disks.

Procedure

Follow steps 1 to 5 under COLDSTART Initiation, then proceed with step 6 as follows:

6. When the message ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS> appears, put the disk to be relabelled in an empty drive. Enter:
   RL
   The system responds with:
   ENTER DISK-NAME OF DISK TO BE RELABELLED

7. Enter the disk's current name, as it appears when OL is issued. The system responds with:
   ENTER NEW DISK-NAME TO BE ASSIGNED TO THIS DISK

8. Enter the new name to be associated with the disk. A disk name is up to seven alphanumeric characters long, beginning with a letter. The system updates the specified disk with with the new name and displays the message:
   END RL
   ENTER FUNCTION <BC, FE, IN, LD, OL, RF RL, RP, WS>

RP (Replace)

The Replace (RP) command can be used to load new software into the system.

It has the advantage over LD in that an RP can be done without removing other files from the system. The disadvantage is that version string checks are not done.

The following system files can be replaced:

SYSBOOTSTRAP
SYSDSCP
SYSMCP1
SYSMCP2
SYSWARMSTART
SYSHOSTLOAD
SYSICP
SYSMPLII
SYSTRANSLATE
SYS-SUPERUTL
SYSCOBOL
SYSCONFIG
SYSLANGUAGE
COPY
SYSCOLDSTART
SYSBLAISE
SYSBNACP
SYSSNACP
SYSINITBOOT

1. To use RP, follow the basic COLDSTART procedures and enter RP when the ENTER FUNCTION <BC, FE, IN, LD, OL, RF, RL, RP, WS> message appears. The new B900RL1 disk must be used to COLDSTART. The system responds with
   ENTER DISK-NAME OF DISK ON WHICH TO REPLACE FILES

2. The disk name may be one to seven alphanumeric characters long, beginning with a letter. SYSPACK is the disk name for the fixed disk.

The Replace function uses the system files listed above from the release disk to replace the copies of these files on the specified disk. If an error occurs, the Replace terminates and any replacement files which have been loaded are deleted. This returns the disk on which the Replace was being performed to its original state.

The function lists all files which have been replaced, and notifies the user when the Replace function finishes successfully, by the message:

END RP

WS (Warmstart)

This command allows the operator to warmstart the system directly from the COLDSTART operation. In general, an operator performs all desired COLDSTART functions (such as initializing a number of disks), then enters WS to prepare the system to begin regular daily work.

This function begins by alerting the operator to set the SYSTEM/REMOVABLE switch to SYSTEM by issuing the message:

SET SYSTEM/REM SWITCH TO SYSTEM

Once the switch is set, the message:

EOJ COLDSTART

appears and the warmstart procedure begins and requests the date and time. Enter DT followed by the date and time.

Example:

DT 01/06/83 1010

The disks that are on-line are then displayed.

Warmstart is finished when the MCP version appears on the SPO.

COLDSTART Attended Mode – No SPO Present

The COLDSTART routine for systems without a SPO may be performed through the hexadecimal display and the keypad to allow execution of individual COLDSTART functions.

Tables 9-5 to 9-11 list the sequences of displays and inputs that follow the status of each of these functions. A description of each function's capabilities has already been given.

Follow steps 1 to 3 of the normal procedure for COLDSTART Initiation. The system is ready for a function to be entered when an 80 appears in Bank D.

Table 9-5 shows the sequence of indicator displays and keypad inputs which follow the status of the binary copy in keypad attended operation.

Table 9-5. Keypad Attended Operation – Binary Copy

Display: Key in:	80 in Bank D 50	Enter function. Selects binary copy function.
Display:	40 in Bank A 50 in Bank D 10 in bank E 00 in Bank F	Binary copy function selected. Enter disk number of target disk.
Key in:	<00-0F>	00 specifies drive A; 01 specifies drive B, and so on.
Display: Key in:	52 in Bank D <00-0F>	Enter disk number of source disk. 00 specifies drive A, 01 specifies drive B, and so on.
Display:	53 in Bank D	Updating track zero.
Display:	54 in Bank D	Copying source disk to target disk.
Display:	55 in Bank D	Verifying target disk parity.

NOTE

During the copy and verify display, Banks G and H will show the two High Order bytes of the disk sector address of the sector currently being read.

The target disk will always be assigned the same disk name as the source disk in keypad attended mode, as there is no way to enter the pack-id through the key pad.

Refer to COLDSTART Messages in COLDSTART Functional Description for error codes displayed in Bank A and corresponding messages.

Table 9-6 shows the sequence of indicator displays and keypad inputs which cause an FE INITIALIZE during keypad attended operation.

Table 9-6. Keypad Attended Operation – FE Initialize

Display: Key in:	80 in Bank D 70	Enter function. Selects FE Initialize function.
Display:	40 in Bank A 70 in Bank D 10 in Bank E 00 in Bank F	FE Initialize function. Enter disk number of disk to be FE initialized.

Table 9-6. Keypad Attended Operation – FE Initialize
(continued)

Key in:	<00-0F>	00 specifies physical drive A, 01 – drive B, 02 = drive C, and so on.
Display:	40 in Bank A nn in Bank C 71 in Bank D 00 in Bank E nn in Bank F	nn is the number of disks being FE initialized. Bank D increments 71-7B as initialization steps are performed. Banks G and H will increment showing disk addresses being write/read tested.

The function is complete when an 80 returns to Bank D. This is a request for a new function.

Table 9-7 shows the sequence of indicator displays and keypad inputs which cause an Initialize during keypad attended operation.

Table 9-7. Keypad Attended Operation – INITIALIZE

Display: Key in:	80 in Bank D 10	Enter function. Selects Initialize function.
Display:	40 in Bank A 10 in Bank D 10 in Bank E 00 in Bank F	Initialize function. Enter the drive number of the disk to be initialized.
Key in:	<00-0F>	00 specifies drive A, 01 specifies drive B and so on. Use FF for fixed disk assemblage.
Display:	40 in Bank A nn in Bank C 12 in Bank D 00 in Bank E	nn is the number of the disk being initialized. Bank D increments 11-1C as initialization steps are performed. Banks G and H will increment, showing the disk address being write/read tested.

The function is complete when an 80 returns to Bank D. This is a request for a new function.

Table 9-8 shows the sequence of indicator displays and keypad inputs which cause a Load during keypad attended operation.

Table 9-8. Keypad Attended Operation – LOAD

Display: Key in:	80 in Bank D 20	Enter function. Selects Load function.
Display:	40 in Bank A 20 in Bank D 10 in Bank E 00 in Bank F	Load function. Enter drive number of disk to be loaded to.
Key in:	<00-0F>	00 specifies drive A, 01 specifies drive B and so on. Use FF for system psuedo-pack.

Table 9-8. Keypad Attended Operation – LOAD
(continued)

Display:	40 in Bank A 20 in Bank D 00 in Bank E nn in Bank F	nn is the number of the disk being loaded to. Bank G will increment from 0 as each file is loaded. When all files have been loaded from the release disk:
Display:	80 in Bank A 21 in Bank D 10 in Bank E 00 in Bank F	Enter 00 to exit or bring the next release disk on line and enter FF to continue.
Key in:	<00 or FF>	

Table 9-9 shows the sequence of indicator displays and keypad inputs which cause a Reformat during keypad attended operation.

Table 9-9. Keypad Attended Operation – REFORMAT

Display: Key in:	80 in Bank D 30	Enter function. Selects Reformat function.
Display:	40 in Bank A 30 in Bank D 10 in Bank E 00 in Bank F	Reformat function. Enter drive number of the disk to be reformatted.
Key in:	<00-0F>	00 specifies drive A, 01 specifies drive B, and so on. Use FF for fixed-disk assemblage.
Display:	40 in Bank A nn in Bank C 30 in Bank D 00 in Bank E nn in Bank F	nn is the number of the disk being reformatted. Bank D increments 31-3C as reformatting steps are performed.

The function is complete when an 80 returns to Bank D. This is a request for a new function.

Table 9-10 shows the sequence of indicator displays and keypad inputs which cause a Replace during keypad attended operation.

Table 9-10. Keypad Attended Operation – REPLACE

Display: Key in:	80 in Bank D 90	Enter function. Selects Replace function.
Display:	40 in Bank A 90 in Bank D 10 in Bank E 00 in Bank F	Replace function. Enter drive number of the disk on which Replace will be performed.
Key in:	<00-0F>	00 specifies drive A, 01 specifies drive B and so on. Use FF for the system pseudo-pack.

Table 9-10. Keypad Attended Operation – REPLACE
(continued)

Display:	40 in Bank A 90 in Bank D 00 in Bank E nn in Bank F	nn is the number of the disk on which the Replace is being performed. Bank G will increment from 0 as each file is replaced. When all files have been replaced from the release disk:
Display:	80 in Bank A 91 in Bank D 10 in Bank E 00 in Bank F	Enter 00 to exit or bring the next release disk on-line and enter FF to continue.
Key in:	<00-0F>	

Table 9-11 shows the sequence of indicator displays and keypad inputs which causes the system to WARMSTART during keypad attended operation.

Table 9-11. Keypad Attended Operation – WARMSTART

Display: Key in:	80 in Bank D 0F	Enter a function. Selects warmstart function.
Display:	40 in Bank A 0F in Bank D 00 in Bank E 0F in Bank F	Warmstart function. This display will flash on and off until the SYSTEM/REMOVABLE switch is set to SYSTEM. At this time, the system will warmstart.

COLDSTART – Operator Unattended Mode

The system automatically enters COLDSTART unattended mode when there is no SPO on the system, or when the COLDSTART operation detects that the NORMAL/EXTENDED switch is set to EXTENDED. In unattended mode, any one of a series of functions may be executed. However, individual functions may not be selected. For example, one series of functions is Initialize, Load, Warmstart. In COLDSTART unattended mode, the user may not run the Initialize function alone because all three functions will be executed automatically. The individual functions are the same as those implemented in COLDSTART attended mode. There are two series of functions which the operator may select:

1. INITIALIZE, LOAD and WARMSTART
2. REPLACE and WARMSTART

The system decides which set of functions to perform by checking the switch settings on the control panel. If the system detects that the SYSTEM/REMOVABLE switch is in the "SYSTEM" position, WARMSTART is initiated. If "REMOVABLE" is detected, COLDSTART will then examine the NORMAL/EXTENDED switch. If the "NORMAL" position is set, the system will expect input from the SPO or keypad. The "EXTENDED" position forces examination of the STANDARD/RECONFIG switch to decide which set of functions is to be executed. When the functions are completed, COLDSTART signals the user to set the control panel switches in their primary positions (SYSTEM, STANDARD, DSKPRC1, OSPRC1, NORMAL, OPERATE) so that WARMSTART can proceed. See figure 9-4 for a description of switch settings and their corresponding functions.

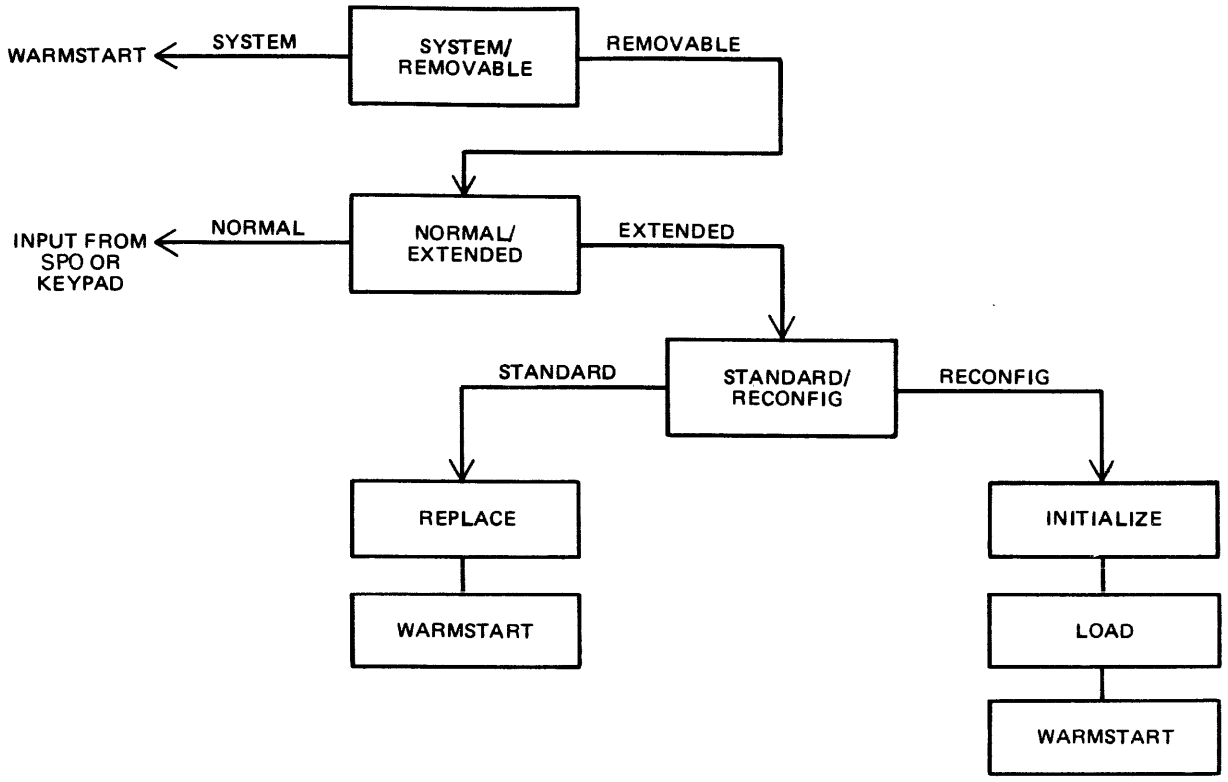


Figure 9-4. COLDSTART Switch Settings

INITIALIZE, LOAD, WARMSTART

In unattended mode the INITIALIZE/LOAD/WARMSTART series of functions is initiated when the user puts a B900RL1 disk in the loader channel and sets/presses the following system switches:

1. Set SYSTEM/REMOVABLE switch to REMOVABLE.
2. Set NORMAL/EXTENDED switch to EXTENDED.
3. Set STANDARD/RECONFIG switch to RECONFIG.
4. Press ON/CLEAR.

The INITIALIZE/LOAD functions act on the fastest disk device available to the system (usually fixed disk). If the user wishes to perform these functions on any device other than the fixed disk, the fixed disk must be physically powered down or it will be initialized.

After the release disk has been inserted and the control panel switches set, the system begins the Initialize function. All the necessary information, such as the number of files allowed, serial number, pack name, and owners-id are derived from default values. The user will be kept "informed" of the COLDSTART function in progress and its status via display banks A and D. Bank A will contain a code reflecting COLDSTART status and bank D will contain a code indicating the current COLDSTART function in progress.

After the Initialize function is complete, control is automatically passed to the Load function; when the Load function is complete, control is automatically passed to WARMSTART.

Bank D will contain @10@-@18@ during the Initialize function, @20@ during the Load function and @50@ during the WARMSTART function. After COLDSTART has loaded the system files from a release disk, COLDSTART will notify the operator by flashing the hexadecimal display on and off. This indicates user intervention is needed. Now the user has the option of inserting the next release disk to continue the Load function, or of setting the SYSTEM/REMOVABLE switch to SYSTEM so WARMSTART can take control.

In operator unattended mode, no check is made for the presence of all required system files. If any are absent, WARMSTART will not operate.

REPLACE, WARMSTART

In unattended mode, the REPLACE/WARMSTART series of functions is initiated when the user puts a B900RL1 disk in the loader channel and sets/presses the following switches:

1. Set the SYSTEM/REMOVABLE switch to REMOVABLE.
2. Set the NORMAL/EXTENDED switch to EXTENDED.
3. Set the STANDARD/RECONFIG switch to STANDARD.
4. Press the ON/CLEAR switch.

As in operator attended mode, the Replace function is used to replace certain files on a previously created disk. Replace does not perform version string checks. COLDSTART will execute the REPLACE/WARMSTART functions on the fastest speed device available to the system. If any load error occurs, the Replace function terminates and whatever replacement files have been loaded are removed. This returns the disk to its original state. COLDSTART will display any errors encountered during the Patch function on the hex display. When the Replace function completes, COLDSTART will flash the hex display on and off to indicate that WARMSTART is ready to take control. The operator should:

1. Remove the B900RL1 release disk.
2. Set the NORMAL/EXTENDED switch to NORMAL.
3. Set the STANDARD/RECONFIG switch to STANDARD.
4. Set the SYSTEM/REMOVABLE switch to SYSTEM.

After step 4 is complete, WARMSTART is initiated automatically.

POWER OFF

The system must be powered off in two steps:

1. Logical power off
2. Physical power off.

The system is logically powered off by entering:
PO SYSTEM (if there are no jobs in the mix)

If there are jobs running and PO SYSTEM has been entered, PO FINAL may be entered to DS the jobs and bring the system down.

The system responds with:
SYSTEM HAS BEEN POWERED DOWN
or whatever sign-off message has been entered in the SYSCONFIG file (using the CONFIGURER utility).

The system can now be physically powered off. Power off should be done in the following order:

1. All peripherals (such as line printer, cartridge drives, tape drives)
2. Fixed disk unit (if present)
3. Disk pack drives and disk pack controller (if present). Do not remove cartridges until red light comes on.
4. SPO
5. Processor.

TAKING SYSTEM DUMPS

During the processing of daily work, system and/or memory failures may occur. When they do, a dump of the contents of memory should be taken so the problem can be documented and subsequently corrected. A "dump" of memory writes the contents of memory to a file.

There are three methods of taking a system dump:

1. System memory dumps occur automatically when:
   1) A clearstart occurs
   2) A task processor (inside the physical unit) goes not-ready.
   3) The OPERATE/MEMDUMP switch is set to MEMDUMP. This automatic dump feature may be used when the system appears to be hung.
2. The operator can take a system memory dump using the the command GT MD.
3. Dumping instructions stored in Read Only Memory (ROM) can be used to take a system dump. This procedure is explained later in this subsection.

Operating System Dumps Created Automatically

A file designed to hold a dump is created automatically each time the system is warmstarted. The file is called SYSDMFILEnn (nn represents numbers between 00 and 05).

When the system warmstarts, it checks to see if SYSMDFILE00 already exists. If file SYSDMFILE00 is not present, this file is created and reserved for a future dump. If a dump is not taken, this empty file will be removed when the system is powered off. If a file names SYSDMFILE00 already exists, the system creates SYSDMFILE01 as the file which will hold a future dump. Up to six dump files may be on the system at any one time (SYSDMFILE00 through SYSDMFILE05).

If the system is warmstarted and six files are already on the system, a message

SYSDMFILE NOT CREATED MAX NO EXCEEDED

appears on the SPO.

Warmstart continues with no problems.

However, it is not recommended that the system is left in a condition where there is no space for a dump file creation. If the existing files are not needed, they can be removed with the RM command. If they are being kept for future reference, they should be copied to another medium and then removed from the system. Once space has been made for a dump file, the system must be re-warmstarted because the system can only write into a dumpfile which was created at the last WARMSTART.

NOTE

If the operator shuts down the system in any order except the correct one, empty dump files may remain on the system instead of being removed automatically. These empty files will eventually be noticed at WARMSTART and all available dump files will be reported as filled. An LR of the dump files is a quick method of determining whether the files are empty or whether they contain useful information.

If an automatic system dump is taken, messages indicating the file was opened and closed will appear on the SPO screen.

If the system hangs or clearstarts and the operator does not see these messages, assume the system dump file was not created and set the OPERATE/MEMDUMP switch to MEMDUMP.

During the automatic dump, indicator banks A through D are used to display the current state of the dump process. Table 9-12 describes the displays and possible errors.

Table 9-12. Automatic Dump Feature Messages

A Bank Display	Description
@01@	The dump routine is locating the SYSMCP files and the SYSDMFILEnn. C Bank – @00@ D Bank – @00@
@02@	RAM memory is being dumped. C Bank – bus address D Bank – page address
@03@	An operation is being performed with the data in "backing storage" (the read/write portion of SYSMCP). C Bank – @00@ D Bank – @01@ – read of backing store from SYSMCP @02@ – write of backing store to SYSDMFILEnn
@E1@	An error was encountered in locating a SYSMCP file or SYSDMFILEnn. B Bank – @01@ – a read error occurred in the first record of the dump file. @02@ – unable to locate a dump file on disk – suspect the creation of the dump file failed at the last warmstart. @03@ – the disk containing SYSMCP is not ready @04@ – the disk containing SYSDMFILE is not ready. C Bank – @00@ D Bank – @00@
@E2@	An error was encountered while RAM memory was being dumped. B Bank – @01@ – a read error occurred in the first record of the dump file. @02@ – unable to locate a dump file on disk – suspect the creation of the dump file failed at last warmstart. @03@ – the disk containing SYSMCP is not ready @04@ – the disk containing SYSDMFILE is not ready C Bank – bus address D Bank – page address.
@E3@	An error was encountered while "backing store" was being dumped. (Backing store is the read/write portion of SYSMCP.) B Bank – @01@ – a read error occurred in the first record of the dump file. @02@ – unable to locate a dump file on disk – suspect the creation of the dump file failed at the last warmstart. @03@ – the disk containing SYSMCP is not ready @04@ – the disk containing SYSDMFILE is not ready C Bank – @00@ D Bank @01@ – read of backing store from SYSMCP @02@ – write of backing store to SYSDMFILEnn.

Using GT MD

The operator uses the GT MD command to create a dump when he/she feels there is something wrong with the way the system is running. This might occur if some job appears to be running for a particularly long time and may be in a loop, or if the system does not appear to be responding correctly.

GT MD is explained in the Utilities and Intrinsics subsection.

Taking A ROM-Created Dump

If setting the OPERATE/MEMDUMP switch to MEMDUMP does not generate a dump, a ROM dump may have to be taken. A limitation specific to the CP 9580 DL4 hardware does not support the ROM dump capability to a 5.25 in disk loader device.

The procedure for taking a ROM memory dump is:

1. Set the OPERATE/MEMDUMP switch to MEMDUMP.

2. Press the ON/CLEAR switch.

3. Insert a disk into the loader channel of the system (designated by the system configuration). The system will not acknowledge the disk.

4. Remove the cover from the hex keypad. Using the hex keypad, press and hold the "S" key while separately pressing the "E" followed by the "D" key.

5. After pressing the "D" key, make a note of the contents of bank "D". If bank "D" contains a @00@, then proceed to the paragraph entitled "Multiple Disk Option". Otherwise continue to step 6.

6. Press and hold the "S" key and then the "F" key. This step initiates the memory dump process. The hex display during execution of the dump will be:
   A and B Banks = Memory Address
   D1 = Bus Address
   D2 = Page Address
   E and F Banks = Disk Sector Address
   If the display stops changing, the code in the D bank is used as an index into the ROM Dump Error Messages.

7. The ROM dump is complete when the hexadecimal display is blank. Note that this may take 15 to 45 minutes.

8. If the ROM dump terminates with a @12@ in bank D, the ROM dump has run out of disk space. This particular system does not have the ability to continue the ROM dump on another disk. However, the disk you have should not be discarded as it contains useful information.

9. The ROM dump file should now be converted using the ROMCONVERT utility.

10. To return to regular operation, put the OPERATE/MEM DUMP switch back to OPERATE and push ON/CLEAR. This is followed by the standard warmstart procedure.

Multiple Disk Operation

The ROM dump may be continued on another disk if bank D contained @00@ when the ROM dump was initiated. The continuation procedure is as follows:

1. Enter a one-digit bus address and a one-digit page address. The bus and page address that should be entered is "10".

2. Press and hold the "S" key and then the "F" key. This step initiates the memory dump process.

3. The ROM dump is complete when all hexadecimal displays are blank. (Go to step 12.)

4. If the ROM dump terminates with a @12@ in bank D, the ROM dump has run out of disk space. To continue the ROM dump on another disk, follow steps 5 and 6.

5. Note the values displayed in banks G and H. This information is the starting point to be entered for continuation of the dump.

6. Press and hold the "S" key, then press the "B" key.

7. Remove the full disk and replace it with a new one.

8. Press and hold the "S" key, then press the "D" key.

9. Enter the values as noted in step 5. (Leading zeros should not be included.)

10. Press and hold the "S" key, then press the "F" key.

11. The ROM dump is complete when the hexadecimal display is completely blank.

12. The ROM dump file should now be converted using the ROMCONVERT utility.

NOTE

A ROM dump overwrites any information on the disk to which it is written but does not overwrite the directory. As a result, if an LR is done of the dump disk, it will appear to have old information on it and not the dump, while the dump is actually present. To check the validity of the disk, the ROMANALYZER command can be entered. If dump file information appears on the printer, the ROM dump was successful.

Analyzing Dumps

When dumps are produced, they are in hexadecimal format. To put the dumps in usable form, Burroughs supplies three utility programs:

1. SYSANALYZER converts the hexadecimal dump information which results from a system dump into readable English so it is easier to examine.

2. ROMCONVERT converts a ROM dump into a format that can be analyzed by the SYSANALYZER program.

3. ROMANALYZER is a program which looks directly at all or part of a ROM dump and is a substitute for the SYSANALYZER program when a ROM dump cannot be converted.

One additional utility, DUMPFILE.UTL, pertains to the dumping procedure. This command can be used to divide a dump file into a number of smaller files or to reconstruct a file that has previously been divided.

The use of these utilities is explained in detail under Utilities and Intrinsics later in this section.

TAKING PROGRAM DUMPS

Sometimes problems arise with individual programs. They may stop processing entirely (without going to end-of-job) or they may go on and on, apparently in a loop.

If this happens, the operator may wish to dump the program using the DP command shown in section 3.

The dumped program can then be analyzed using the ANALYZER command explained in Utilities and Intrinsics later in this section.

FUNCTIONAL DESCRIPTION OF SYSTEM STARTUP

This subsection contains a functional description of procedures that the system follows every time the system is warmstarted or coldstarted. COLDSTART is a stand-alone utility whose primary function is to create and maintain a system disk and user disks. COLDSTART is not normally run as part of system operation. It is intended to be used when disk(s) and/or system software require maintenance. WARMSTART is a stand-alone utility whose primary function is to determine the system hardware and software configuration and pass control to the MCP (that is the Operating System). The MCP has the responsibility of co-ordinating all system resources and interface to the user. The MCP must be up and running before any COBOL, RPG, MPLII programs or system intrinsics can be executed on the system.

Startup software consists of the following programs:

1. SYSTEM STARTUP READ-ONLY MEMORY (ROM)
2. CMS TRACK ZERO BOOTSTRAP
3. SYSTEM DEPENDENT BOOTSTRAP (SYSBOOTSTRAP)
4. SYSCOLDSTART
5. SYSWARMSTART

As soon as the system enters startup mode, the startup ROM routine (permanently resident in the system) takes control. It looks for a disk that is powered on and running. It loads the TRACK ZERO BOOTSTRAP from that disk into memory and starts executing the TRACK ZERO BOOTSTRAP code.

The task of TRACK ZERO BOOTSTRAP is to find a disk that contains a valid CMS label and a file called SYSBOOTSTRAP. After both these conditions are met, the TRACK ZERO BOOTSTRAP code loads the SYSBOOTSTRAP file and starts its execution.

NOTE

Once SYSBOOTSTRAP starts to execute, its action depends on the setting of the SYSTEM/REMOVABLE switch. If this switch is set to SYSTEM, the SYSBOOTSTRAP program will load the WARMSTART utility. If the switch is set to REMOVABLE, it will load the COLDSTART utility from a disk labelled B900RL1.

Therefore, the operator is responsible for selecting whether the system will COLDSTART or WARMSTART when it is put in startup mode by setting the SYSTEM/REMOVABLE switch appropriately.

During startup, Bank A of the hexadecimal display lights will indicate which startup software is in operation. Bank A will indicate the status of the program's execution. To identify the startup program currently in control, refer to table 9-13.

Table 9-13. Bank A Hex Displays for Startup Programs

Value Range (Bank A)	Program Executing
@00@-@1F@	SYSTEM STARTUP ROM
@20@-@2F@	TRACK ZERO BOOTSTRAP
@30@-@3F@	SYSBOOTSTRAP
@40@-@9F@	SYSCOLDSTART
@A0@-@FF@	SYSWARMSTART

B 900/CP 9500 Bootstraps

TRACK ZERO BOOTSTRAP receives control directly from the startup ROM routine. Its responsibility is to locate, load and pass control to the SYSBOOTSTRAP program.

It searches the various disks on the system for the disk which contains the SYSBOOTSTRAP. This disk must be ready and have a valid CMS disk label.

NOTE

When warmstarting or coldstarting B 920/CP 9570 or CP 9500-2 systems with disk pack, a disk other than disk pack must be on-line and have a valid 3.05 TRACK ZERO BOOTSTRAP.

SYSBOOTSTRAP (the system dependent bootstrap) is loaded by and receives control from TRACK ZERO BOOTSTRAP.

SYSBOOTSTRAP is responsible for:

1. Checking the system Random Access Memory (RAM).
2. Locating the disk which contains the COLDSTART/WARMSTART utilities, the system software file, SYSDSCP (disk processor code) and file SYSHOSTLOAD if a B 9387 disk pack drive is on-line.
3. Loading COLDSTART or WARMSTART (depending on the setting of the SYSTEM/REMOVABLE switch) and SYSDSCP into the B 900/CP 9500.
4. Loading the B 9387 Disk Pack Drive Controller (DPDC) with the file, SYSHOSTLOAD, if the B 9387 DPDC is on-line.
5. Passing control to COLDSTART or WARMSTART.

The COLDSTART/WARMSTART disk is one which is found by SYSBOOTSTRAP to be ready, to have a valid CMS disk label and to contain the required system files.

SYSBOOTSTRAP uses the same method of searching for the COLDSTART/WARMSTART disk as was used to find the SYSBOOTSTRAP disk. The search sequence is as follows: from the fastest disk device to the slowest, from the highest I/O channel to the lowest, with each channel being searched from the lowest disk drive to the highest.

To load and execute the stand-alone utility COLDSTART, the disk must have a pack-id of B900RL1 and contain the system files SYSCOLDSTART and SYSDSCP. If the B 9387 DPDC is on-line, the B900RL1 disk must also contain the file SYSHOSTLOAD.

To load the stand-alone utility WARMSTART, the required system files are SYSWARMSTART and SYSDSCP. If a fixed disk exists on the system, the two files (SYSDSCP and SYSWARMSTART) need not reside on the same fixed disk drive, as long as both are on fixed disk. When no fixed disk is present, both required files must reside on the same removable disk. Also, if the B 9387 DPDC is on-line, the WARMSTART disk must also contain the system file, SYSHOSTLOAD. The search sequence is as follows: from the fastest disk device to the slowest, from the highest I/O channel to the lowest, with each channel being searched from the lowest disk drive to the highest.

Table 9-13 shows the format of the hex display during TRACK ZERO BOOTSTRAP and SYSBOOTSTRAP execution. The primary status of TRACK ZERO BOOTSTRAP and SYSBOOTSTRAP execu-

tion can be determined by the value appearing in bank A. Banks B to H contain additional information on TRACK ZERO BOOTSTRAP and SYSBOOTSTRAP status. For details of the identifiers found in banks B to H of table 9-13 refer to tables 9-14 and 9-15.

Table 9-14. TRACK ZERO BOOT Error Displays

A Bank Display	Description
@20@	NORMAL EXECUTION – NO ERRORS TRACK 0 BOOT is searching for a SYSBOOTSTRAP: C Bank = Most digit = channel, Least digit = drive D Bank = disk drive id
@21@	Disk operation outstanding: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H Bank = disk IOC status.
@22@	Invalid disk operation length: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H Bank = disk IOC status
@23@	Invalid disk drive id: B Bank = disk operation being performed – @80@ = read, @20@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H Bank = disk IOC status
@24@	Invalid channel: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H Bank = disk IOC status
@25@	Disk drive went not ready: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H Bank = disk IOC status
@26@	Disk device error: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit, = channel, least digit = drive D Bank = Drive id read E through H Bank = Disk IOC status
@27@	Key search failure: B Bank = disk operation being performed –

Table 9-14. TRACK ZERO BOOT Error Displays
(continued)

A Bank Display	Description
	@80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H bank = disk IOC status
@28@	Disk drive invalid: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E through H Bank = disk IOC status
@2E@	No SYSBOOTSTRAP can be found. Note: a valid device error may have been detected on an otherwise valid disk. Ensure that the startup disk drive is ready.
@2F@	SYSBOOTSTRAP has loaded successfully and will take control.

Table 9-15. SYSBOOTSTRAP Error Displays

A Bank Display	Description
@30@	SYSBOOTSTRAP has gained control and is relocating itself: B Bank = 00 – SYSBOOTSTRAP is relocating itself in memory. E Bank = If not blank, the relocate could not complete, reason unknown.
@31@	Memory check out is being performed: B Bank = @00@ – Disk processor memory @01@ – SYSCOLDSTART/SYSWARMSTART MEMORY @02@ – COLDSTART/WARMSTART BUFFER MEMORY C 1 Bank = Bus address C 2 Bank = Page address E Bank = If not blank: check was unsuccessful, reason unknown.
@32@	SYSBOOTSTRAP is performing a search for a file: B Bank = @00@ search for SYSWARMSTART or SYSCOLDSTART @01@ search for SYSDSCP C Bank = most digit = channel, least digit = drive D Bank = drive id read E bank = 00 = search in progress 01 = search successful 02 = search unsuccessful
@33@	SYSBOOTSTRAP is loading a file to memory: B Bank = 00 – SYSCOLDSTART or SYSWARMSTART is loading to OS processor 01 – SYSDSCP is loading to disk processor C Bank = most digit = channel, least digit = drive E Bank = 00 – load in progress 01 – load complete else – load failure; reason unknown

Table 9-15. SYSBOOTSTRAP Error Displays
(continued)

A Bank Display	Description
@34@	SYSHOSTLOAD loading: If C and E through H are blank B Bank = segment number being loaded D Bank = 01 = loading from disk to OS memory 02 = transferring from memory to B 9387 If C and E through H are not blank B Bank = most digit = 0, least digit = channel C Bank = ERROR 29 = BAD IOC STATUS before segment transfer complete 30 = BAD IOC STATUS after segment transfer complete 31 = B 9387 returned bad result after transfer complete 32 = B 9387 did not return result of a segment 33 = B 9387 won't enter segment load D Bank = segment number If E Bank = 31 E – H Banks = result descriptor sent by B 9387 else E Bank = IOC STATUS F – H Banks = decimal number of digits of the segment transferred.
@3B@	Execution was interrupted by an unexpected REQ from an IOC: B Bank = request register
@3C@	Disk exception condition: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive D Bank = drive id read E Bank = @21@ – a disk operation was requested and has not yet completed @22@ – an invalid disk operation length was requested @23@ – unrecognizable disk drive id @24@ – an invalid channel is being addressed
@3D@	Device error: B Bank = disk operation being performed – @80@ = read, @20@ = search on key, @10@ read drive id C Bank = most digit = channel, least digit = drive, D Bank = drive id read E Bank = disk IOC status
@3E@	SYSBOOTSTRAP has encountered a hardware detected error: B Bank = IC error status If data access C and D Banks = M1 register E and F Banks = MAX register G and H Banks = M2 register If micro address C and D Banks = UMAR register E and F Banks = UMARX register

Table 9-15. SYSBOOTSTRAP Error Displays
(continued)

A Bank Display	Description
@3F@	SYSBOOTSTRAP terminating: E Bank = blank – normal termination 00 – normal termination 01 – cannot find OS processor 02 – cannot find a SYSCOLDSTART/SYSWARMSTART disk 03 – not able to set processor freeze on itself

WARMSTART FUNCTIONAL DESCRIPTION

The WARMSTART utility starts up when the COLDSTART utility or SYSBOOTSTRAP passes control to it. WARMSTART is responsible for:

1. Determining the hardware configuration by associating each processor that it finds and its local memory size and the appropriate processor attribute (that is, Operating System, Task Processor, Data Comm Processor or Disk Processor).
   WARMSTART uses these attributes to verify the specifications found in the SYSCONFIG file and to load the proper system firmware files.

2. Determining that all required system files are available by searching all ready disks on the system. The search proceeds from the fastest drive to the slowest. (The order is described in table 9-13.) All required system files must be found to reside either entirely on the system fixed disk assemblage or entirely on a single removable disk.

3. Assigning appropriate roles in the software system to each component of the hardware system as follows:

   1) If the user configuration specifications in the SYSCONFIG file are satisfied by the hardware currently available, WARMSTART configures the software system according to the SYSCONFIG file.
   2) If the current hardware system cannot satisfy the user configuration specifications in the SYSCONFIG file, or if the user has specified that WARMSTART is to use the default assignments, then WARMSTART will use the default configuration. (See Default Configuration, following.)

4. Loading system firmware as required by the software system assignments.

5. Passing control to the operating system (that is, the MCP).

NOTE

If the system disk has just been loaded with release software via COLDSTART, then the version of SYSCONFIG that is used by WARMSTART is the version supplied on the release disk(s). If the CONFIGURER program has been executed since the last COLDSTART, the version of SYSCONFIG that is used by WARMSTART is that produced by the most recent execution of CONFIGURER. To restore the original release version to the system disk, do either a new COLDSTART replace or execute CONFIGURER with the proper parameters.

Default Configuration

The B 900/CP 9500 will WARMSTART using a "default configuration" if the specifications in the SYSCONFIG file conflict with the actual hardware configuration, or if the user has requested that default values be used for those areas of SYSCONFIG that may default.

The three areas that could cause the system to default are the Buffer Memory, Task Processor and Data Comm Processor assignments. Each of these areas is considered independent and only those areas with invalid requests will be assigned default values.

Buffer Memory Defaults

If a conflict arises between the amount of memory assigned in the SYSCONFIG file for buffer memory and the amount of memory available on the OS processor, then Buffer Memory will be assigned as follows:

Memory Pages on OS Processor	Total Buffer Memory	Data Comm Buffer Memory
More than 4	80 K	40 K
4	80 K	40 K
3	64 K	32 K
2	32 K	16 K

Task Processor Defaults

If a conflict exists between the task processor assignments in the SYSCONFIG file and the task processors actually available on the system, the following default configuration will be used:

- all task processors with more than 128 KB of memory will be assigned all interpreters.
- task processors with 128 KB of memory or less will be assigned one interpreter, alternating between MPLII and COBOL.

Data Comm Processor Defaults

If a conflict arises between the Data Comm Processor (DCP) assignments in SYSCONFIG and the Data Comm Processors actually available on the system, WARMSTART will assign all the Data Comm Processors physically on the system as Data Comm Processors. The logical DCP numbers assigned to each processor will increase from zero, starting with the DCP at the highest bus address.

To determine if the system has defaulted in any of these areas, the "GT CF" command can be used to display the system configuration. This command is described under Utilities and Intrinsics.

To tailor the SYSCONFIG file for specific configurations, the CONFIGURER program must be used. A repeat WARMSTART can then be done which will load the desired configuration. (The CONFIGURER program is described under Utilities and Intrinsics.)

WARMSTART has two ways of reporting its status to the operator. When a SPO is present on the system, WARMSTART attempts to direct messages to the SPO device. In both operator attended and unattended operation, WARMSTART uses the hexadecimal display to indicate the status of its execution. During WARMSTART, Bank A is primary indicator of WARMSTART's progress through its code, while Banks B to H provide additional information on the status of WARMSTART.

Table 9-16 outlines the contents and meanings of values appearing in the display banks during WARMSTART.

Table 9-16. General Description of WARMSTART Messages

Bank A	Errors detected by WARMSTART are displayed in Bank A. WARMSTART errors can be broken down into the following ranges:

Value	Meaning
@A0@	NORMAL EXECUTION NO ERROR
@A1@-@AF@	WARMSTART INTERNAL ERRORS
@B0@-@BF@	SYSTEM ERROR
@C0@-@CF@	HARDWARE ERROR

Bank B	Bank B contains a count of successfully executed logical operations. This bank has a maximum possible value of @FF@ (255 decimal).
Bank C	In the event of an error, this bank represents the last operation attempted during which an error occurred. (During normal operation, it is periodically updated to reflect the operation in progress.) The following table is a decode of the contents of bank C.

Value	Operation
@00@	PROCESSOR INTERFACE CONTROL (PIC) COMMAND
@10@	LIST PERIPHERALS
@11@	LOCATE OS PERIPHERAL
@12@	DETERMINE HARDWARE CONFIGURATION
@13@	LIST REMOTE PERIPHERALS
@14@	LOCATE SYSTEM DISK
@20@	PRIME LOCAL DEVICE SOFTWARE CONTROLLER
@21@	PRIME DISK PROCESSOR PARAMETERS
@30@	ALLOCATE WARMSTART BUFFER MEMORY
@31@	SYSTEM CONFIGURATION
@40@	HANDLE DISK PROCESSOR CTM PREP
@50@	MEMORY CHECKOUT – ATTACHED
@51@	MEMORY CHECKOUT – UNATTACHED
@60@	PREPARE FIRMWARE SEGMENT LOAD
@70@	LOAD FIRMWARE SEGMENT
@72@	COMPLETE WARMSTART TABLE
@80@	DISPLAY MESSAGE TO SPO
@EO@	WARMSTART COMPLETE – PASS CONTROL TO OS

A Bank	Warmstart Error Displays Description
@A0@	Normal execution, no errors: B Bank – number of logical operations performed C Bank – operation in process
@A1@	Opcode not recognized: B Bank – number of logical operations performed. C Bank – operation in process E-F Bank – current offset G-H Bank – next offset
@A2@	Operation not completed successfully: B Bank – number of logical operations performed. C Bank – Operation in process E-F Bank – current offset G-H Bank – next offset

Table 9-16. General Description of WARMSTART Messages
(continued)

A Bank	Description
@A3@	WARMSTART table entry not recognized B Bank – number of logical operation performed. C Bank – operation in process
@A8@	Invalid operation for selected device: B Bank – number of logical operations performed C Bank – operation in process D Bank – operation type: 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type: 00 = disk, 01 = SPO, 02 = processor, 03 = memory F Bank – if disk: most digit = logical drive number or if processor memory: most digit = bus, least digit = page G, H Bank – device dependent parameters
@A9@	Device not recognized B Bank – number of logical operations performed. C Bank – operation in process D Bank – operation type: 00 = command, 01 = read, 02 = write, 03 = search, E Bank – device type: 00 = disk 01 = SPO, 02= procesor, 03 = memory F Bank – if disk: most digit = logical drive number or if processor or memory: most digit = bus, least digit = page G, H Bank – device dependent parameters.
@AA@	Invalid unit for this device: B Bank – number of logical operations performed. C Bank – operation in process D Bank – operation type 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type: 00 = disk, 01 = SPO, 02 = processor, 03 = memory F Bank – if disk: most digit = logical drive number or if processor memory: most digit = bus, least digit = page G, H Bank – device dependent parameters
@AB@	Source descriptor invalid – does not exist or exceeds physical attributes: B Bank – number of logical operations performed C Bank – operation in process D Bank – operation type 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type 00 = disk, 01 = SPO, 02 = processor, 03 = memory F Bank – if disk: most digit = logical drive number or if processor memory: most digit = bus, least digit = page G, H bank – memory address or error
@AC@	Destination descriptor invalid – does not exist or exceeds physical attributes: B Bank – number of logical operations performed C Bank – operation in process

Table 9-16. General Description of WARMSTART Messages
(continued)

A Bank	Description
	D Bank – operation type: 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type: 00 = disk, 01 = SPO, 02 = processor, 03 = memory F Bank – if disk: most digit = logical drive number or if processor memory: most digit = bus, least digit = page G, H Bank – memory address of error
@AD@	Device not on system B Bank – number of logical operations performed. C Bank – operation in process D bank – operation type: 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type: 00 = disk, 01 = SPO, 02 = processor, 03 = memory F Bank – if disk, most digit = logical drive number, or if processor or memory: most digit = bus, least digit = page G, H Bank – device dependent parameters
@B0@	System disk not found: B Bank – number of logical operations performed. C Bank – operation in process
@B1@	SYSCONFIG file error: B Bank – number of logical operations performed C Bank – operation in process
@B2@	Firmware segment not found B Bank – number of logical operations performed C Bank – operation in process D Bank – file being accessed: 03 = SYSMCP, 04 = SYSICP, 05 = SYSDSCP, 06 = SYSMPLII, 07 = SYSCONFIG, 08 = SYSLANGUAGE, 09 = SYSTRANSLATE E Bank – segment type: 02,06,0A = resident code, 03 = resident data
@B3@	Unable to load a firmware segment, exceeds memory attributes: B Bank – number of logical operations performed. C Bank – operation in process D Bank – processor type: 01 = OS, 02 = DCP, 03 = task, 04 = disk, 05 = buffer
@C1@	Unable to access OS processor B Bank – number of logical operations performed. C Bank – operation in process
@C3@	Insufficient memory to WARMSTART: B Bank – number of logical operations performed C Bank – operation in process
@C4@	ID in ROM invalid: B Bank – number of logical operations performed C Bank – operation in process

Table 9-16. General Description of WARMSTART Messages
(continued)

A Bank	Description
@C8@	Device error: B Bank – number of logical operations performed C Bank – operation in process D Bank – operation type: 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type: 00 = disk, 01 = OST, 02 = processor, 03 = memory F Bank – if disk: most significant digit = logical drive number, or if processor or memory: most significant digit = bus, least significant digit = page G Bank – general status (see description for @C9@) H Bank – device error status (see description in @C9@)
@C9@	Device not ready: B Bank – number of logical operations performed. C Bank – operation in process D Bank – operation type: 00 = command, 01 = read, 02 = write, 03 = search E Bank – device type: 00 = disk, 01 = SPO, 02 = processor, 03 = memory F Bank – if disk: most digit = logical drive number or, if processor or memory: most digit = bus, least digit = page G Bank – general status 00 – controller error (internal to DP) 01 – command successful 02 – command unsuccessful 03 – device error aborted command 04 – not used 05 – descriptor error H Bank – device error status 00 – no error 01 – seek timeout 02 – head off cylinder 03 – sequence error (internal to DP) 04 – parity error 05 – sector not found 06 – illegal address 07 – status word error 08 – data error 09 – write inhibited

COLDSTART FUNCTIONAL DESCRIPTION

COLDSTART is a stand-alone utility whose primary function is to create and maintain a system disk and user disks. COLDSTART operates in two distinct modes, Operator Attended Mode and Operator Unattended Mode. Both modes and their functions are discussed. Users should be aware of which mode of COLDSTART their system is operating under, as it is not selectable by the operator. COLDSTART's mode of operation is determined by the system hardware configuration. Basically, a system will operate in Operator Attended Mode if the system has a SPO, thus allowing COLDSTART to display information to and request information from the operator via the SPO device. If a system does not have an SPO device, then COLDSTART will communicate with the operator via the hexadecimal keypad and display. Operator Unattended Mode refers to COLDSTART functions being performed according to the settings of the panel switches. Refer to paragraphs headed COLDSTART – Operator Unattended Mode.

B 900/CP 9500 system software is distributed on release disk(s) labelled B900RL1, B900RL2 and so on. It is the user's responsibility to maintain these disks with system software patches (fixes) which are distributed in the form of B 900/CP 9500 flashes. Users must update these disks to maintain their system and system disk. The maintenance of the release disks (that is, applying system software patches) is not a function of COLDSTART. The Field Patch Program (FPP) which runs under MCP control is used to apply system software patches to the release disks. After patches have been applied to the release disks, a COLDSTART function may be used to replace the files on the system disk. The Field Patch Program is described in detail in this section under Utilities and Intrinsics.

FE (Initialize MTR Disk)

This function initializes a disk for Maintenance Test Routine (MTR) usage. Old disk contents are deleted, the disk surface is qualified and a log in the disk label is set to indicate it is an MTR disk.

Six specific tracks on any given MTR disk are reserved for use by MTR routines. This is done by setting "Bad Sector" indicators in the disk's non-file (available space) directory. If these six tracks are found to be bad at any location, MTR initialize terminates. Success of this function means that those six tracks are flawless and unconditionally reserved for MTR usage.

The first request issued when the FE function has been selected is for identification of the MTR disk's drive. The following message appears:

ENTER DRIVE-ID OF DISK <3 CHARACTERS, SUCH AS "DKA">

Respond with a three-letter drive mnemonic.

Example:

DMF denotes that the MTR disk is a mini disk, residing on drive F.

A fixed disk (DF=) cannot be used as the MTR disk.

The system's next message is:

VALUES FOR DISK-NAME, OWNER'S ID, SERIAL-NUMBER AND MAX-NUMBER-OF-FILES-ON-DISK MUST NOW BE ASSIGNED. DO YOU WISH TO ASSIGN THESE VALUES YOURSELF OR HAVE THE SYSTEM DO IT FOR YOU?

ENTER "SELF" OR "SYSTEM"

If "SYSTEM" is selected (default mode), all further information needed by FE initialize is derived from the default values. No further information is required from the operator. Refer to "COLD-START DEFAULT ASSIGNMENT" later in this section.

If "SELF" is selected, FE initialize will generate the disk directory and label using the information supplied to the following queries:

ENTER DISK-NAME

Enter the name of the disk being initialized for MTR usage. The name is selected by the operator and may be up to seven alphanumeric characters long, beginning with a letter.

The system responds with:

ENTER SERIAL-NUMBER

Enter a serial number up to six digits long which may be chosen at random if there is no need to keep a record of MTR disk serial numbers.

The system now requests the owner's id via the message:

ENTER OWNER'S-ID

This id name may be from 1 to 14 alphanumeric characters long.

The next information required is the maximum number of files to be allowed on the disk.

ENTER MAX-NUMBER-OF-FILES-ON-DISK

The possible range is from 1 to 2804 files.

Bad sectors within the six tracks reserved for usage by the MTR will cause failure of the FE initialize and the message "BAD MTR TRACK" to be displayed. Other tracks, which will later contain MTR software, are allowed up to a total of 50 bad sectors. The message reported to the user is:

TRACK # <number> BAD FOR <number> SECTOR(S)

When FE initialize ends successfully, the following message is displayed.

END FE

Increasing an Existing Fixed Disk Assemblage

All fixed disk drives on a B 900/CP 9500 system that are powered on and ready at COLDSTART initialize time are initialized as a single fixed disk assemblage. A procedure is available for adding one or more fixed disk drive units to an existing fixed disk assemblage without having to re-initialize the existing assemblage. The process is outlined in the following procedure:

1. Bring the system up under COLDSTART control.

2. Power off all fixed disk drives that are not to be initialized. This is to protect the files on these units from erasure by the initialize function.

3. Power on the fixed disk drive(s) that are being added to the fixed disk assemblage.

4. Use the initialize function to initialize the drive(s) to be added. Specify the same number of files as those specified on the original assemblage. A unique label will be requested for each drive initialized.

5. After the initialize function terminates normally, restore power to all fixed disk units.

6. Rewarmstart the system by following the WARMSTART procedure described under WARMSTART.

The MCP will automatically add the new drives to the fixed disk assemblage once the system has been warmstarted.

NOTE

When adding one or more drives into an existing fixed disk assemblage, the additional drives must be placed in an available slot or channel, higher than any other channel which is being used or which was used. When replacing a disk or channel on the system, the new disk must be cabled into the same channel from which the old disk was removed. It also must be given the same packid and initialization parameters.

This procedure has not been qualified for 211 disks.

COLDSTART Disk Selection

All disks have been assigned priority according to their speed. The fastest disk device is the fixed disk, followed by disk pack, cartridge disk, then mini disk.

In Operator Attended Mode, the operator may select which drive/disk the desired COLDSTART function is to act upon. In Operator Unattended Mode, COLDSTART searches for the highest priority device because the operator cannot make specific requests to COLDSTART.

COLDSTART Default Assignments

The FE, IN and RF functions of COLDSTART allow the operator the option of assigning his own disk attributes, or defaulting to predetermined values. The default values are always used in COLDSTART Operator Unattended Mode. In the COLDSTART Operator Attended Mode, the message

ENTER "SELF" OR "SYSTEM"

is displayed on the SPO. If "SYSTEM" is entered, the system will assign the number of files to be allowed on disk, serial number, packid and owner-id as shown in table 9-17.

Table 9-17. Disk Attribute Default Values for IN, FE and RF Functions

Information Needed	Removable Disk Defaults	Fixed Disk Defaults
Number of files to be allowed on system disk.	256 **	2805 **
Serial number of disk used as system disk.	000000 (EBCDIC)	000000 (EBCDIC)
Packid of disk used as system disk	000000A *** (ASCII)	0000001* (ASCII)
Owner-id of disk	"Removable disk"	"Fixed disk"

Table 9-17. Disk Attribute Default Values for IN, FE and RF Functions
(continued)

NOTES:

* If there is more than one fixed disk, the first has the packid as shown; each additional disk packid is incremented by 1.

** 2805/256 includes the SYSMEM file. 2804/255 is the actual number of user files which can reside on the disk.

*** "A" is the third letter of the device mnemonic. For example, DMB would get 000000B.

COLDSTART primarily uses banks A and D to indicate its status. Bank A is used to display a code which corresponds to COLDSTART's status or an error message. A list of codes appearing in bank A and messages appearing on the SPO during COLDSTART is shown in table 9-18.

COLDSTART Messages

COLDSTART Error Displays

The function being performed is shown in bank D and defined in table 9-19.

Table 9-18. COLDSTART Error Displays

A Bank Display	Description
@40@	NO ERROR – normal operation.
@41@	ERROR ON TRACK ZERO – an attempt was made to initialize a disk that has a bad sector on Track 0 (Track 0 is required to be good).
@42@	ERROR ON MTR TRACK – an address required for an FE track has a bad sector.
@43@	UNABLE TO ALLOCATE DIRECTORY SPACE – there are not enough contiguous good sectors for the disk directory size chosen within the area to which the B 900 assigns directories.
@44@	TOO MANY BAD SECTORS – an attempt was made to initialize a disk that has more than 50 bad sectors.
@45@	UNABLE TO INITIALIZE CYLINDER ON DISK PACK – the initialize or reformat function encountered a disk error.
@46@	AVR – invalid disk label found while initialize function was attempting to replace track 0 and bootstrap.
@47@	UNABLE TO READ AVAILABLE TABLE – initialize or reformat function encountered error on source available table.
@48@	AVR – invalid disk label was found while the initialize function was rebuilding the available table.
@49@	RELOCATE – unable to decode address while initializing.
@4B@	DEVICE ERROR – BAD TRACK ZERO – an attempt was made to initialize a disk that has a bad sector on Track 0.

Table 9-18. COLDSTART Error Displays
(continued)

A Bank Display	Description
@4C@	UNABLE TO WRITE SECTOR ADDRESSES – error encountered on destination disk by initialize or reformat function.
@4D@	UNABLE TO WRITE AVAILABLE TABLE – error encountered on destination disk by initialize or reformat function.
@4E@	UNABLE TO WRITE TRACK 0 BOOTSTRAP – error encountered on destination disk by initialize or reformat function.
@4F@	UNABLE TO WRITE DISK FILE HEADERS – error encountered on destination disk by initialize or reformat function.
@50@	UNABLE TO WRITE DISK FILE NAME LIST – error encountered on destination disk by initialize or reformat function.
@51@	UNABLE TO WRITE PPIT – error encountered on destination disk by initialize or reformat function.
@52@	AVR – invalid label encountered while initialize function attempting to write SYSMEM header.
@53@	UNABLE TO WRITE SYSMEMFILE – error encountered on destination disk by initialize function.
@54@	UNABLE TO WRITE LABEL – relabel function encountered error on destination disk.
@55@	UNABLE TO LOCATE PPIT ENTRY – relabel function encountered error on source disk.
@56@	UNABLE TO WRITE PPIT ENTRY – relabel function encountered error on destination disk.
@57@	SYSINITBOOT NOT STORED IN MEMORY – COLDSTART cannot find B900RL1/SYSINITBOOT
@59@	UNABLE TO READ DISK FILE HEADERS – disk processor encountered error.
@5A@	SYSINITBOOT NOT STORED IN MEMORY – see @57@
@5B@	SYSINITBOOT NOT STORED IN MEMORY – see @57@.
@5F@	UNABLE TO READ PPIT – disk processor encountered error processing source disk.
@60@	CONFLICTING PPITS – the PPITs found are not identical on all fixed disk drives.
@61@	CONFLICTING PPITS – see @60@.
@62@	UNABLE TO WRITE PPIT – disk processor encountered error processing destination disk.
@63@	UNABLE TO WRITE PPIT – see @62@.
@64@	CONFLICTING PPITS – see @60@.
@65@	CONFLICTING PPITS – see @60@.

Table 9-18. COLDSTART Error Displays
(continued)

A Bank Display	Description
@67@	AVR – invalid label encountered by load or replace function while building file list.
@68@	FILE AREA SIZE TOO LARGE – COLDSTART encountered a file with an area larger than 65535 sectors.
@69@	UNABLE TO REMOVE FILE AREAS – the load or replace function encountered error on destination disk.
@6A@	DISK ALREADY CONTAINS FILES – the LD function has detected that the disk is not properly initialized or reformatted.
@6B@	UNABLE TO READ DISK FILE NAME LIST – load or replace function encountered error on source disk.
@6C@	UNABLE TO WRITE DISK FILE HEADER – load or replace function encountered error on destination disk.
@6D@	UNABLE TO ALLOCATE FILE AREA – load or replace function encountered error on destination disk.
@6E@	UNABLE TO WRITE DISK FILE HEADER – load function encountered error on destination disk.
@6F@	FILE NOT REPLACED – RP function terminated.
@70@	UNABLE TO READ DISK FILE HEADER – replace function encountered error on source disk.
@71@	UNABLE TO WRITE DISK FILE HEADER – replace function encountered error on destination disk.
@72@	CHECK LOAD – LD incomplete, all required system files were not loaded.
@73@	INVALID DISK NUMBER – disk number was not FF or in range 00 through 0F.
@75@	INVALID SYSINITBOOT FILE – COLDSTART cannot find correct B900RL1/ SYSINITBOOT file.
@76@	DISK NOT READY – COLDSTART is trying to access a disk that is not on line.
@77@	UNABLE TO DECODE ADDRESS – initialize function encountered error during address.
@78@	POWER OFF FAILURE – disk processor detected error while attempting to power off disk drive.
@79@	NO FIXED DISKS ON LINE – fixed disk is not ready.
@7A@	INVALID FUNCTION – an invalid function code was entered.
@7B@	DEFAULT PACK ALREADY PRESENT – a disk with the default name to be used is already on the system.
@7C@	AVR – disk is either write inhibited or has an invalid name.

Table 9-18. COLDSTART Error Displays
(continued)

A Bank Display	Description
@7D@	DRIVE NOT ON LINE – the specified disk drive is not on line.
@7E@	INVALID DEVICE TYPE – disk cannot be fixed or pack (DF or DP).
@7F@	NO SYSTEM PSEUDO PACK ONLINE – FF was entered but there is no system disk on line.
@80@	DISK NOT ONLINE – a disk ID of a not ready disk was entered; if the next release disk is required, bring it on line and enter FF to continue.
@81@	MULTIPLE DISKS ONLINE WITH SAME NAME – duplicates are not allowed.
@82@	SPECIFIED DISK NOT PERMITTED FOR LOAD OR REPLACE – Load/Replace must be performed on the system pseudo-pack or a removable disk.
@83@	FILE NOT LOADED – an attempt was made to load an invalid file type.
@84@	DISK NOT ONLINE – B900RL1 disk is required on line now.
@85@	DISKS NOT COMPATIBLE – binary copy requires the disks to be of the same type.
@86@	READ ERROR ON TRACK ZERO – specified disk has bad sector on track 0.
@87@	WRITE ERROR ON TRACK ZERO – specified disk has bad sector on track zero and cannot be written to.
@88@	INPUT ERROR DURING COPY – a read error occurred on source disk that could not be isolated to exact sector.
@89@	INPUT ERROR DURING VERIFICATION – a read error occurred on target disk that could not be isolated to exact sector.
@8A@	ERRORS OCCURRED DURING COPY AND VERIFICATION – binary copy may have aborted.
@8B@	WRITE ERROR ON TARGET DISK – target disk has disk error and cannot be written to.
@8C@	SOURCE AND TARGET ARE SAME DISK – the source and target disks specified reference the same disk; they must be separate.
@8D@	UNABLE TO INITIALIZE 211 DISK – initialize function encountered error attempting to initialize the fixed disk.

During COLDSTART Bank D will contain a code reflecting the COLDSTART function currently operating. The COLDSTART functions are outlined in table 9-19. This information should be used in conjunction with the information in table 9-18.

Table 9-19. COLDSTART Functions

D Bank Display	Description
@01@	ENTER DATE – system will accept date entry.
@02@	ONLINE – list disks online.
@03@	?EX – explanation for required response.
@0F@	WARMSTART – system is warmstarting.
@10@	INITIALIZE – enter disk name and parameters for initialize.
@11@	INITIALIZE – write sector addresses.
@12@	INITIALIZE – perform read/write tests.
@13@	INITIALIZE – delete bad tracks.
@14@	INITIALIZE – calculate directory size.
@15@	INITIALIZE – write new label.
@16@	INITIALIZE – write track zero bootstrap.
@17@	INITIALIZE – rite non-file-directory (available table).
@18@	INITIALIZE – write disk-file-header-list.
@19@	INITIALIZE – write disk-file-name-list.
@1A@	INITIALIZE – write pseudo-pack identifier table (fixed-disks only).
@1B@	INITIALIZE – write SYSMEM.
@1C@	INITIALIZE – pseudo-pack identifier table fixup.
@20@	LOAD – enter disk-name for load function.
@21@	LOAD – continue with next release disk or exit.
@22@	LOAD – build list.
@23@	LOAD – removing old file.
@24@	LOAD – load new file.
@30@	REFORMAT – enter disk-name for replace function.
@34@	REFORMAT – calculate directory sizes.
@35@	REFORMAT – write new label.
@36@	REFORMAT – write track 00 (label and track 0 bootstrap).
@37@	REFORMAT – write non-file-directory (available table).
@38@	REFORMAT – write disk-file-headers.
@39@	REFORMAT – write disk-file-name-list.
@3A@	REFORMAT – write pseudo-pack identifier.
@3B@	REFORMAT – write SYSMEM.
@3C@	REFORMAT – pseudo-pack identifier table fixup.
@50@	BINARY COPY – enter target disk-id for binary copy.
@51@	BINARY COPY – enter target disk-name.

Table 9-19. COLDSTART Functions
(continued)

D Bank Display	Description
@52@	BINARY COPY – enter source disk-id.
@53@	BINARY COPY – update track 00.
@54@	BINARY COPY – copy.
@55@	BINARY COPY – verify.
@70@	FE INITIALIZE – enter disk-name for FE initialization.
@71@	FE INITIALIZE – write sector addresses.
@72@	FE INITIALIZE – perform read/write tests.
@73@	FE INITIALIZE – delete tracks.
@74@	FE INITIALIZE – calculate directory sizes.
@75@	FE INITIALIZE – write new label.
@76@	FE INITIALIZE – write track 0 bootstrap.
@77@	FE INITIALIZE – write non-file-directory (available table).
@78@	FE INITIALIZE – write disk-file-header-list.
@79@	FE INITIALIZE – write disk-file-name-list.
@7A@	FE INITIALIZE – write pseudo-pack identifier table (fixed disk only).
@7B@	FE INITIALIZE – create system entry.
@80@	IDLE LOOP – wait for next function to be entered.
@90@	REPLACE – enter disk-name for replace function.
@91@	REPLACE – continue with next release disk or exit.
@92@	REPLACE – build list.
@93@	REPLACE – remove old file.
@94@	REPLACE – copy new file.
@96@	REPLACE – rename new file.
@97@	REPLACE – replace bootstrap.
@A0@	RELABEL – enter disk-name for relabel function.
@A1@	RELABEL – update disk label.
@A2@	RELABEL – update pseudo-pack identifier table.

DUMPS FUNCTIONAL DESCRIPTION

Read Only Memory (ROM) Dump Routine

The ROM dump routine is a software debugging tool which resides in the system ROM. It is used when the MCP is unable to produce a system dump file for analysis by the System Dump Analyzer program (SYSANALYZER). It allows the dumping of all contents of system RAM, (Random Access Memory) to a CMS compatible removable disk. This disk must be in the loader I/O channel.

Disk Format

The data being dumped will be written onto the dump disk beginning at the first good sector after the directory. The dumped data skips across any bad sector(s) encountered during the dump process. If a bad sector is found, the first good sector after it is used.

The dumping algorithm described in the previous paragraphs results in the overwriting of the contents of the files already on the disk. The disk directory is not updated to reflect this corruption.

The system must first locate the dump disk and locate the end of the disk directory.

Dumping begins in the first byte following the directory end address. If this would cause the dump to begin in a bad sector, however, the first byte of the next good sector is used.

Data is dumped in 180-byte records. The first record contains a validity string, which indicates that the disk contains a ROM dump. The validity string is:

B900PROMDUMP0.Q

The first nine bytes of each page of memory dumped will contain a validity string of the following format:

Bus address: one byte
Page address: one byte
Page size: two bytes
Error status: one byte
IC-Status 1: one byte
IC-Status 2: one byte
Next page: two bytes, disk sector address for start of dump of next page.

The respective memory page contents, from its zero address to its limit, will be dumped immediately following its descriptor. The descriptor will start on a record boundary. Each memory page dumped will take up only the number of records necessary for it and its respective descriptor.

The last record written by the ROM dump will be either:

1. A terminator record, when the disk is able to contain a complete dump (that is, a dump consisting of the entire system RAM memory dump, the validity record and the terminator record) indicated by a blank display.

   or

2. The last record that could be dumped for the current memory page, when an end of disk condition occurred (indicated by 12 displayed in bank D).

The terminator record, if written, consists of the following string:

"B900PROMDUMPED"

ROM Dump Messages

Hardware detected errors are handled by presenting error relevant information on the hexadecimal displays in one of the following formats:

Bank D	Bank A	Bank B	Bank C	Bank E	Bank F	Bank G	Bank H
@08@, @18@, @28@, @38@, @58@, @88@, @C8@, @FF@	OFFSET		- - -	STATUS	- - -	MAX REG	
@04@, @10@-@13@	- - -	- - -	- - -	- - -	- - -	- - -	- - -
@01@-@03@	SECTOR	ADDR	- - -	DCPS	DCSS	- - -	- - -

Table 9-20 lists the meanings of error codes found in bank D of the hexadecimal display. For a decode of the identifiers and codes found in banks A-C and E-F refer to table 9-21 ROM Dump Display Reference Table.

Table 9-20. Bank D Error Codes

Error Code	Possible Causes	Suggested Action
08	Hardware detected memory error, either a data read parity error or memory limit occurred during an execution of Dump-To-Display.	Re-enter Dump-To-Display option as keypad will be enabled for entry of next option (option idle state).
18	An error occurred during the setup for, or execution of, the Dump-to-Disk option.	System clear is necessary to restore the processor to a running state.
28	Dump routine could not find a page in memory while dumping to disk.	Same action as @18@.
38	Dump routine failed while checking disk label prior to dumping to disk.	Same action as @18@.
58	A bad area was found on the dump disk.	Same action as above.
88	Read after write error was detected after Dump-to-Disk was complete.	Same action as @18@.
C8	A write error was detected when dumping to disk.	Same action as @18@.
FF	A fatal error. A hardware detected error has occurred which gives an unexpected error status.	System clear is necessary.

Table 9-20. Bank D Error Codes
(continued)

Error Code	Possible Causes	Suggested Action
04	The selected dump disk has gone from operational to non-operational.	Restore dump disk to operation state. Dump will automatically continue from point of interruption.
10	No memory was found to dump.	Read/Write memory has not been found to exist on the hardware system. The keypad is left enabled or is in option idle state.
11	A dump disk was not found. Dump-To-Disk requires a write enabled, operational disk on the dump disk channel.	Insert a dump disk into a valid drive and power up. The keypad is enabled or is in option idle state. It is necessary to respecify the Dump-To-Disk option.
12	The end of the dump disk was reached before the dump was complete.	See action for error code 11.
13	A valid CMS disk label was not found on the first write enabled, operational disk.	Power off invalid disk and insert a valid dump disk in the correct drive.
01	A disk seek error has occurred.	The error may have been caused by either the disk or the disk drive unit. Reinitiate the Dump-To-Disk using another disk and/or alternative drives where available.
02	A disk read error has occurred.	Same action as for 01.
03	A disk write error has occurred.	Same action as for 01.

ROM Dump Display Reference Table

Tables 9-21 and 9-22 contain ROM Dump display information.

Table 9-21. ROM Dump Display Reference Table

Identifier	Display Bank(s)	Meaning
DCPS	E	Disk controller primary status.
DCSS	F	Disk controller secondary status.
OFFSET	A, B	Data offset or micro offset where the error occurred.
MAX REG	G, H	The contents of the max hardware register at the time of the failure.
SECTOR ADDR	A, B	The disk sector address where the error was detected.
STATUS	E	The STATUS code found in bank "E" may be decoded as shown in table 9-22.

Table 9-22. Bank E Display for ROM Dump

Bank E STATUS Bit	Name	Value	Meaning
0	Write/read access	1	A write operation was being performed when the error occurred.
		0	A read operation was being performed when the error occurred.
1	Data/micro access	1	A data access was in progress when the error occurred.
		0	A micro instruction was being fetched when the error occurred.
2	Memory limit error	1	An attempt was made to access non-existent memory.
3	Boundary error	1	A boundary error has occurred.
4	Write parity error	1	A write parity error has occurred.
5	Read parity error	1	A read parity error has occurred.
6	Error state	1	An error has occurred since the last clear or reset error status word command.
		0	An error has not occurred since the last initialize or reset error command.
7	Second error	1	A second error has been detected before the error state bit was reset. Occurrence of a second error will freeze the processor.
		0	A second error has not occurred.

NOTE Bank E status bit 0 is the least significant bit. For example, if Bank E contains a @22@, this indicates that a read parity error occurred while doing a data access.

SPECIAL CONSIDERATIONS

This subsection contains information which does not apply to one individual section such as COLDSTART or WARMSTART but applies to B 900/CP 9500 systems in general.

Pseudo-Pack Concepts

The concept of pseudo-packs (also called pseudo disks) is unique to the B 900/CP 9500. Pseudo-packs apply only to systems with fixed disks.

This concept allows multiple fixed disk drives to be handled as one large contiguous disk.

The fixed disk media can be utilized in two different ways:

1. It is possible for space on all the fixed units to be logically concentrated into one large available area. Files can then be placed on the fixed disks without the operator being concerned about the physical disk on which they are located.

2. It is possible for the operator to assign files to a physical disk drive if necessary.

Many users run several different application programs during the course of a day. Often, these programs require the disks to be labelled with specific names. The concept of pseudo-packs allows the user to maintain a table of these disk names on the fixed disk. This means the user neither has to relabel the disk every time he wants to run an application, nor to modify the programs to change the disk name they require at the moment.

A table of pseudo-pack names is created during COLDSTART when the fixed disk is initialized. The names of all fixed disks on the system at that time are placed in the table. This entry is referenced for system files and user files assigned to the system disk.

As you become familiar with the computer, the convenience of adding pseudo-pack names to the table will be more obvious to you.

They are added with a simple command: AP (Assign Pseudo-pack).

The syntax of this command is explained in the Utilities and Intrinsics section. It would be most frequently used as:

```
AP NEWDISK/
  or
AP NEWDISK/  ON DKA
```

(where DKA is the name of the physical unit).

Pseudo-packs are manipulated like any physical disk. For example, they can be relabelled using the RL command.

Files are accessed by giving the disk name followed by the file name.
Example:

```
PD FIXEDA/PRMASTER
```

The system pseudo-pack has one unique characteristic which is very convenient for the operator. A file that is placed on the system pseudo-pack (called SYSPACK) can be accessed using only the file name.
Example:

```
PD PRMASTER
```

Using PD SYSPACK/PRMASTER, which follows the rule, is perfectly correct and also will access the file PRMASTER. Omitting the disk name is merely a convenient shortcut for the operator.

Refer to section 4 of this manual for additional information about pseudo-pack concepts.

Data Communications

The data communications system connects users at terminals other than the SPO to the computer.

A detailed explanation of how this system operates is contained in the Computer Management System (CMS) Data Communications Subsystem Reference Manual, form 1090909. However, a brief overview of data communications may be helpful for the operator.

The B 900/CP 9500 systems are designed for the efficient use of data communications lines. A basic system can have many terminals attached to it.

Probably the best way to visualize this system is as terminals which are some distance from the system itself.

In a school system for example, the computer processor, SPO, disk files and printer are probably located in one room in an administration building. However, the people who need to use the system are in other rooms in the administration building and also in separate buildings such as various elementary and high school offices.

Operators at these terminals, other than the SPO are called "remote" users. The remote terminals are attached to the computer through a communications system. Sometimes these remote terminals are wired directly to the processor, but most often they are connected through telephone lines.

When Burroughs supplies programs to operate your computer system, a package which operates these data communications lines is included. Each time a system is started, the operator can choose whether or not to start (bring up) the data communications system.

Until the computer operator types the appropriate command into the system, none of the remote users can operate their terminals.

At the end of the working day, the system operator brings down the data comm system before shutting off the processors and disks. This also requires a special command which is supplied by Burroughs.

Remote users can communicate with the system once the data comm system has been started. However, compared to the system operator they have less authority. There are also slightly different rules governing the way remote users can communicate with the system.

You should be aware that remote users cannot use the computer until their system is activated, that they have slightly different operating rules and that there are some differences in the way they communicate with the system.

Enhanced Terminal SPO

The Enhanced Termnal SPO (ETS) feature allows data comm terminals to enter SCL (System Control Language) commands. A terminal designated as a terminal SPO can ready a printer, assign a device, enter a MX, DS a program or enter any other commands which previously could only be entered from the local SPO.

To allow a terminal to be a SPO, the following conditions must be true.

1. Using the CONFIGURER utility, the Operator Control Flag field in the SYSCONFIG file must be set to ALL.

2. In the NDL, the SPOCAPABLE flag in the terminal section must be set equal to TRUE. This statement is not required in the NDL and will default to SPOCAPABLE=TRUE.

3. Also in the NDL, if a station references a SPO capable terminal, the station's SPO capability is controlled by the SPO=TRUE flag in the station section. This flag defaults to false. If the SPO flag is set to FALSE or if the statement is not present, a REDEFINE.STATION communicate may be issued.
   The REDEFINE.STATION communicate (RS) is explained in the GEMCOS User's Manual (form 1106796). Bit 4 of the TYPE field is now used to set the SPO capability of a station.
   To find out what the TYPE field of a station contains enter:
   $OL S <lsn> D
   where lsn is the logical station number.
   The TYPE field can then be reset so bit 4=1, making the station SPO capable:
   $RS <lsn>/TYPE=@nnnn@
   where nnnn are the four hex digits defining the station type.
   The line associated with the station being redefined should be made not ready before the command is issued and should be made ready after the command (refer to the GEMCOS User's Manual form 1106796).

SPO capable stations can be switched between two modes. The station will come up in "application" mode. This mode corresponds to the normal operation of the station and it will behave as in previous releases.

Depressing the "SPECIFY" key switches the station into "SCL" mode. On stations with two pages of memory, SCL mode will correspond to the second physical page. On stations with one page of memory both modes will map onto the same (only) physical page.

There is a new option in CONFIGURER to allow a key other than SPECIFY to be used to switch from SCL mode to application mode. Any other key or combination of two to four keys may be designated. This feature is designed for those users who have application programs which use the SPECIFY key.

The key(s) designated to switch modes in ETS are defined in the SYSCONFIG field DC REMOTE SPO MODE CHANGE. Once the SYSCONFIG file has been changed, to designate mode change keys, the system must be re-warmstarted and data comm must be reloaded before the changes will take effect. The cursor must be in the home position in order to switch modes if mode change keys ae defined.

In SCL mode, the station will act like the local SPO. SCL commands can be sent to the MCP, and the responses will be returned to the station. If a program is started from an ETS station in SCL mode, any accept and display messages, and a copy of the beginning and end of job banners will be routed back to that station's SCL page. Depressing the SPECIFY key again will return the station to application mode, the cursor will be returned to its last position and forms mode will be re-entered, if required.

The current state of the station will be indicated by status line messages displayed on the right hand side of the bottom line on the screen. When the station is in application mode and there are no SPO messages waiting to be displayed, the last eight characters of the status line will be blank. If there are SPO messages waiting to be displayed while in application mode, the status line will display "SCL".

SCL mode is indicated by the letters "SCL" in reverse video (dark letters on light background). If, when in SCL mode, there are application messages waiting to be displayed, the status line will display "APPL" in reverse video.

In general, an ETS station will only receive those SPO messages and can only issue those SCL commands that are needed to control programs which it has started (for example, AX, DS, ST). If a MX is entered, the response will contain only those programs started from the SPO requesting the MX.

The exception to this is when the station becomes the system SPO. The system SPO has the ability to manipulate the entire system and, for this reason, the unsolicited messages that give the state of the system (such as hardware error messages and device status messages) are routed only to the system SPO. In addition, every time a BOJ/EOJ banner is issued to any other SPO on the system, a copy will also be issued to the system SPO. The system SPO is permitted to issue certain restricted SCL commands (for example, AP, PO, PR). A list of these commands is given in table 9-23. There can only be one system SPO at any one time. At WARMSTART, the system SPO capability defaults to the local SPO, if there is one. The SCL command "CTL ON" requests system SPO capability, and "CTL OFF" releases that capability. (See section 4.)

Table 9-23. SCL Commands And Issuing SPO

Mnemonic	Function	Issued By
AD	Assign Device	System SPO.
AP	Add Pack	System SPO.
AT	At Hostname	Any SPO.
AX	Accept	Originating SPO or system SPO.
CA	Candidate	Any SPO.
CH	Change	Any SPO.
CL	Clear	Originating SPO or system SPO.
DB	Direct to Backup	Originating SPO or system SPO.
DC	Data Communications	Any SPO.
DP	Discontinue and dump	Originating SPO or system SPO.
DS	Discontinue	Originating SPO or system SPO.
DT	Date – Time	Changed by system SPO, interrogated by any SPO.
EX	Execute	Any SPO.
FD	Forms Define	Any SPO.
GO	Go	Originating SPO or system SPO.
HN	Host Name	Any SPO.
IR	Initiate Recall	Any SPO.
KX	Interrogate filesizes	Any SPO.
LB	Continue recall	Any SPO.
LF	Continue recall	Any SPO.
LG	Log Go	System SPO.
LS	Log Stop	System SPO.
MX	Mix	Any SPO.
NW	Network	Any SPO.
OL	On Line	Any SPO.
PD	Print Directory	Any SPO.
PG	Purge Device	System SPO.
PO	Power Off	System SPO.
PR	Priority	System SPO.
RD	Reserve Device	System SPO.
RL	Relabel	System SPO.

Table 9-23. SCL Commands And Issuing SPO
(continued)

Mnemonic	Function	Issued By
RM	Remove	Any SPO.
RY	Ready	System SPO.
SF	Substitute File	Originating SPO or system SPO.
SM	Multi-MCS data comm	Any SPO.
SN	Serial Number	System SPO.
SO	Set PRBK Option	System SPO.
ST	Stop	Origination SPO or system SPO.
SV	Save	System SPO.
TO	Test Option	Any SPO.
VF	Vertical Format	System SPO.

NOTE The system SPO is a SPO that has successfully issued a "CTL ON" SCL command. An originating SPO is the SPO that started the program in question.

Tape Concepts

The B 900/CP 9500 systems can store information on tape as well as on disks and packs.

When compared to disk, tape storage has some advantages and some disadvantages. The advantages are:

1. Tapes tend to be a less expensive storage media than disks.
2. Tapes can make it easier to move information from one computer system to an unrelated system.
3. Tapes use less external storage space than disk.

Disadvantages of tapes are:

1. They are more susceptible to being corrupted by exposure to magnetic fields.
2. They are less satisfactory than disk for long-term storage as the images tend to fade (that is, storage over several years which varies depending on the quality of the tape purchased).
3. It takes longer to read information stored on tape into the computer system than it does to read the same information stored on disk. It is also faster to write information to disk than it is to write to tape.

Tape Commands

A separate set of commands is used to manipulate information stored on tape. The most frequently used are the variations of the LD command: LOAD, ADD, DUMP and UNLOAD. These are discussed in section 4 of this manual.

OUTPUT DEVICE = " device-id SYSMEM " <peripheral type>

Disk Pack Concepts

Disk packs vary somewhat from disks in the method by which they are handled.

1. A file called SYSHOSTLOAD must be present on the system and on-line during WARM-START or COLDSTART if the computer system is to recognize and access a disk pack device. SYSHOSTLOAD is usually stored on the system disk, but may be located on any device that is on-line. During the system startup, SYSHOSTLOAD is loaded to the disk pack controller.

2. Turning on the disk pack controller also sends power to the disk pack drive. However, an operator must press the READY button on the drive after a pack has been loaded. Files on that pack can then be accessed by the system.

3. If the controller is powered-off or taken off-line, pack(s) cannot be accessed until the system has been re-started.

Disk Pack Used To Store System Software

It is possible to place the system software on a pack device. There may be some advantage to doing so, as more physical room will be available.

To do so:

1. The system must be coldstarted by loading system files to the pack using a B900RL1 disk.
2. The B900RL1 disk is then removed.

The pack is now the home of the system disk. The operator must remember, when warmstarting the system when pack is the system disk, that some disk unit (fixed, cartridge or mini) must be on-line at warmstart time.

This is because the system must access a bootstrap program in order to WARMSTART and this program is contained only on track zero of individual disks. Once the system has contact with a bootstrap program, it can recognize the system files located on pack.

Testing Recording Surface of Disk Devices

Non-Disk Pack Devices

Bad sectors may cause INITIALIZE to terminate. In Unattended mode, this will cause COLDSTART to terminate. There are two ways this can occur for non-disk pack devices.

1. The disk contains more than 50 bad sectors. This is indicated to the operator by the message: TOO MANY BAD SECTORS

2. There is a bad sector in track zero, which is indicated to the operator by the message: TRACK ZERO HAS BAD SECTOR

COLDSTART uses these messages to indicate the associated disk media is not fit for use on B 900/ CP 9500 systems.

Disk Pack Devices

The first phase of the INITIALIZE function for disk pack drives is called "INITIALIZE – VERIFY – RELOCATE", or IVR. IVR is performed on a cylinder by cylinder basis, with all passes being performed on one cylinder before moving on to the next. The passes performed on each cylinder are as shown in table 9-24.

Table 9-24. Pack Initialization Procedure

Pass	Description
@00@	Initialize, sector addresses and a data pattern are written.
@01@	Verify, sector addresses and the data patterns written in pass @00@.
@02@	Initialize data fields and sector data are written with a new data pattern.
@03@	Verify new data pattern written in pass @02@.

During the actual IVR process, the pass number is displayed in Bank E of the hex display, while Banks G and H contain the decimal cylinder number of the cylinder being IVR'ed. The disk pack devices have the following number of cylinders:

206 disk pack: 815, numbered 0000-0814
205 disk pack: 407, numbered 0000-0406

IVR Error and Failure Conditions

The B 9387 DPDC status is displayed for all IVR error and failure conditions as follows:

<controller mnemonic> STATUS: <extended result descriptor>

The extended result descriptor is the B 9387 DPDC status. The extended result descriptor has the same breakdown as the extended result descriptor that is logged in maintenance logs and is 46 bytes long.

The following error and failure conditions may occur during an IVR:

1. If pass @00@ fails on any cylinder, then the IVR aborts with the message:
   UNABLE TO INITIALIZE CYLINDERS

2. If a verify operation fails for any reason, an attempt is made to relocate the sector where the failure occurred. If the relocate fails, IVR aborts with the message:
   SECTOR XXXXX NOT RELOCATED
   CYLINDER XXX HEAD X SECTOR XX

Sector Relocation

Sector relocation is done only during the IVR process. The system will attempt to move (relocate) a sector that has been found to have an error to a spare sector on the cylinder. There can be a maximum of five relocates per cylinder. COLDSTART will display messages to indicate the success or failure of the relocation process and the sector address of the sector that was or was not successfully relocated.

If a successful relocation does occur, COLDSTART displays:

<Controller mnemonic> STATUS <extended result descriptor>
CYLINDER CCCC HEAD H SECTOR SS RELOCATED F

This indicates that the sector with the error has been moved (relocated) to a spare sector on the cylinder and the pass which detected the data error is performed again.

If a sector relocation fails for any reason, then the IVR aborts and COLDSTART displays:
<Controller mnemonic> STATUS <extended result descriptor>
CYLINDER CCCC HEAD H SECTION SS NOT RELOCATED ON <DFM>
TOO MANY BAD SECTORS

NOTE

The message TOO MANY BAD SECTORS is correct in this instance. It is an indication that the associated disk pack is not fit for use on B 900/ CP 9500 systems.

The formula used to convert the cylinder head sector display into a binary sector address is:

206 DISK PACK: 445 * CYLINDER + 90 * HEAD + SECTOR = ADDRESS
205 DISK PACK: 445 * CYLINDER + 90 * HEAD + SECTOR = ADDRESS

The approximate length of time required to perform an IVR is as follows:
206 disk pack media: 25 minutes per drive
205 disk pack media: 12 minutes per drive

Miscellaneous

If @FF0000@ appears in banks "E", "F", "G" and "H" respectively and the other banks are blank, then there is a parity error in disk processor memory. This is an indication of a hardware failure which will require technical assistance.

Backup Operating System and Disk Processors

Some systems are equipped with two additional switches on their control panel. These switches are labelled as: OS PRC1/OS PRC2 and DKS PRC1/DSK PRC2. The function of these switches is to bypass faulty hardware in the operating system or disk processors. Normally, the system is configured so that the OS PRC1/OS PRC2 switch is in control of the OS processor when the switch is in its primary position (that is, OS PRC1) and the DSK PRC1/DSK PRC2 switch is in control of the disk processor when the switch is in its primary position (that is, DKS PRC1). If either the OS processor or the disk processor has faulty hardware, then the operator may continue operating the system by switching to a backup OS or disk processor by putting one or both of the switches in the secondary positions (OS PRC1 and DSK PRC2). However, these switches are useful only if the system has the additional hardware required to support this function. Basically, the system must have two additional processors which are identical to the primary OS and disk processors. These two additional processors may function as additional task processors in accordance with WARMSTART defaults and SYSCONFIG parameters. By switching the OS PRC1/OS PRC2 and DSK PRC1/DSK PRC2 switches to their secondary positions, the two additional task processors now become the OS and disk processors and the old OS and disk processors become task processors. The procedure used to set either the OS or disk backup processors is as follows:

1. Logically power-off the system disk using the PO command.
2. Press either the OS PRC1/OS PRC2 switch to OS PRC2 or the DSK PRC1/DSK PRC2 switch to DSK PRC2.
3. Rewarmstart the system using the WARMSTART procedure described under WARMSTART.

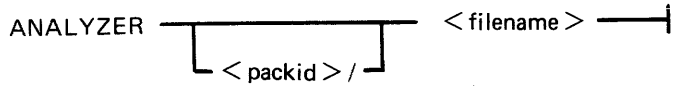
UTILITIES AND INTRINSICS

The following commands are unique to the B 900/CP 9500. Each command is followed by its syntax, an explanation of the command and at least one example.

ANALYZER (User Program Dump Analyzer)

When a COBOL, RPG, or MPLII program is dumped (DP intrinsic, section 3), a file is created on a disk with a name DMFILnn (where n is the mix number of the program dumped). This utility provides an analysis of this file printed on the line printer.

The ANALYZER utility can be initiated in any mix. The initiating syntax is shown below:



where: <packid> is the name of the pack where the dump file resides. If none is specified, system pack is assumed.

<filename> is the name of the dump file created when program was DP'ed.

NOTE

When a program is dumped, the dump file is automatically placed on the same disk as the program object code.

The system responds with:

<mix no>/ANALYZER DISP: PURGE DUMP FILE?
<mix no>/ANALYZER REQUIRES DATA. USE "AX".

The operator can then respond with AX <mix no> followed by YES or NO. (YES removes the original dump file.)

The system then displays:

<mix no>/ANALYZER EOJ

CAUTION

When analyzing a COBOL or RPG user program, if the object code file is removed or changed between the time the program is DP'ed and the time ANALYZER is initiated, the analysis is likely to be incomplete.

Output Messages

Message	Possible Causes	Suggested Action
PACKID TOO LONG <packid>	A packid containing more than seven characters was entered.	Re-enter initiating message, using correct packid.
FILE ID TOO LONG <filename>	The file name containing more than twelve characters was entered.	Re-enter initiating message, using correct file id.
INVALID CHARACTER IN FILE	The initiating message contained characters other than alphanumerics, commas or spaces.	Re-enter initiating message, using only legal characters.

(continued)

Message	Possible Causes	Suggested Action
<filename> NOT FOUND OR IN USE	One of the following: 1. <packid> was of valid length and contained valid characters, but identified wrong pack. 2. <fileid> was of valid length and content, but dump file is not on disk.	If file is not on disk, install disk containing dump file. Otherwise re-enter initiating message.
INTERP NOT SUPPORTED	Dump file is from program not handled by COBOL or MPLII interpreters.	Cease trying to analyze dump file.
NO SPECIFICATION GIVEN	Error in initiating message not covered by preceding messages.	Re-enter initiating message.

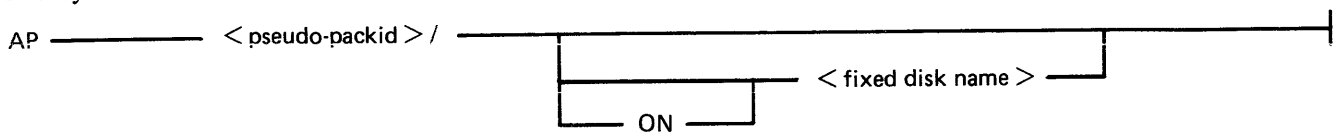
AP (Assign Pseudo-Pack)*

*Pseudo-pack and pseudo-disk are interchangeable terms in this context.

The AP intrinsic is used to create a pseudo-pack on fixed disk. For more details on pseudo-packs refer to Special Considerations.

The pseudo-pack cannot be created on any removable media.

The syntax for this command is:



NOTE
The slash following the pseudo-packid is required.

The naming format of a pseudo-pack is the same as it is for any other disk name. The name may be up to seven characters long and must begin with a letter.

When a peripheral is specified as follows:

AP PSEUDO1/ ON DFA

the information written to that pseudo-pack goes only to the fixed disk specified (DFA), not to any other part of a fixed disk unit.

However, if no peripheral is specified as follows:

AP

the information written to that pseudo-pack can go to any section of any fixed disk where there is available space.

Consequently, creating an unrestricted pseudo-pack can resolve a NO USER DISK condition which occurs on a single physical unit, since this allows the operator to spread a file over multiple disk units.

Note that ALL fixed disks must be physically on and ready for the command to work.

The name of the newly-created pseudo-pack is entered into the Pseudo-Pack Identifier Table (PPIT) which is a list of all pseudo-packs declared on the system.

Examples:

AP PSEUDO1/
– adds unrestricted pseudo-pack "PSEUDO1" to PPIT.
Therefore you can access all of the fixed disk using PSEUDO1.

AP PSEUDO2/ ON DFA
– adds pseudo-pack "PSEUDO2" to PPIT and restricts it to fixed disk A.
Therefore, you can access fixed disk A using PSEUDO2.

Once a pseudo-pack has been created, it cannot be removed unless the fixed disk is initialized. However, the pseudo-pack can be relabelled.

Message	Possible Causes	Suggested Action
PSEUDO-PACK <packid> ADDED TO SYSTEM	The command was completed successfully.	None – normal operation.
AP <packid>/INVALID – DISK NAME INCORRECTLY FORMED	The pack-id specified is not legal.	Re-initiate with a valid pack-id.
AP <packid> IGNORED – CANNOT ADD ANY MORE PSEUDO-PACKS TO SYSTEM AN EXISTING PSEUDO-PACK MUST BE USED	The PPIT is full.	Use RL intrinsic to relabel an existing pseudo-pack that is no longer required.
AP IGNORED – PACKID IS ALREADY ONLINE	A duplicate pack condition has been detected.	Re-initiate with a different pack-id.
AP <packid> IGNORED – ALL DISKS BELONGING TO FIXED DISK ASSEMBLAGE MUST BE ONLINE BEFORE COMMAND ISSUED. ENSURE DISKS ARE POWERED ON.	The peripheral specified is not ready.	RY the specified device and re-initiate.
AP <packid>/INVALID – DEVICE TYPE NOT ALLOWED WITH THIS COMMAND	The peripheral specified is not a fixed disk.	Re-initiate specifying a valid peripheral.
AP <packid>/ON <fixed disk name> INVALID	The fixed disk specified is not on the system.	

AT (At Host)

This is only used for BNA. See the B 900/CP 9500 Burroughs Network Architecture Operation and Installation Guide (form 1163102). The AT intrinsic allows the user to enter a request via any BNA network terminal and specify which machine (HOST) is to handle the request.

The syntax for this command is:

where <hostname> is a maximum of 17 characters long.

Valid characters include upper and lower case A-Z, digits 0-9, the hyphen () and the underline ( ). The special characters (comma and underline) may not be the first character in the hostname, nor may two or more appear consecutively.

Output Messages

Message	Possible Causes	Suggested Action
"AT" IGNORED – SYSTEM IS NOT IN NETWORK MODE	BNA Master MCS Program (BNAMASTER) is not currently in the mix and logged on as a programmatic SPO.	Request rejected.
"AT" IGNORED – BNAMASTER PROGRAM IS TEMPORARILY UNABLE TO ACCEPT COMMANDS	BNA Master's SPO queue is full.	Request rejected.
"AT" IGNORED – SYNTAX OF HOSTNAME IS INVALID	Specified hostname is incorrectly formed.	Request rejected.

The programmatic SPO header will have the input flag set and the text length, SPO id, SPO tag and SPO size fields primed with the appropriate contents. The system SPO flag will be set if this command came from the system SPO.

BINARY.COPY (Duplicate Disk Media)

The BINARY.COPY utility allows rapid duplication of disks of the same size and type under MCP control. However, both disks must be CMS disks.

Before executing BINARY.COPY, the operator must reserve the disk drives on which the BINARY-.COPY is being performed by the RD command. When the utility begins execution, the operator will be requested to assign the required disk drives (using the AD command) which were previously reserved. (See section 3 for an explanation of RD and AD.)

As a precaution, the operator should also write-disable the disk to be copied. However, since the system does not recognize a write-disabled disk, only a BINARY.COPY can be done. To do anything else, the disk must be write-enabled.

CAUTION

The entire source disk is copied to the target disk. Any files previously on the target disk are destroyed.

The following table illustrates the procedure for executing BINARY.COPY.

Operator Enters	System Messages
RD <device mnemonic>	<device mnemonic> <packid>/RESERVED
RD <device mnemonic>	<device mnemonic> <packid>/RESERVED Note: the write-disabled disk does not list its packid.
BINARY.COPY	ENTER TARGET PACKID <mix no>/BINARY.COPY REQUIRES DATA USE "AX"
AX<mix no><target packid>	
THIS PACKID WILL BE PLACED IN THE LABEL OF THE TARGET DISK AFTER THE COPY IS COMPLETED. THE PACKID ENTERED SHOULD NOT BE THE SAME PACKID AS THE SOURCE DISK.	
	<mix no>/BINARY.COPY WAITING SOURCE/SYSMEM
	<device mnemonic> DEVICE ASSIGNMENT REQUIRED
AD<mix no><device mnemonic> (source)	<mix no>/BINARY.COPY WAITING TARGET/SYSMEM
	<device mnemonic>
	DEVICE ASSIGNMENT REQUIRED
AD<mix no><device mnemonic> (target)	

All sectors are copied from the source disk to the target disk. If errors are encountered on the source disk, the affected sectors will be copied as binary zeroes and the operator will be informed. The utility will display these messages both on the SPO and on a line printer report.

After the copy is complete, the target disk is read to ensure no parity errors exist on it. This message is also printed on the SPO and on the line printer.

Sample Printout:

```
**************************************
BINARY COPY FROM DKD TO DKC

CHECKING TARGET DISK FOR INPUT ERRORS

READ CHECK ON TARGET DISK COMPLETE

DKD TO DKC COPIED AS PPD3

**********NO ERRORS IN COPY**********
```

The drives that were reserved must now be made "ready" using RY <device mnemonic>.

Example:
To copy disk GEORGE on DKA to HARRY on DKB:
Enter:

```
RD DKA
RD DKB
BINARY.COPY
```

The system will then ask for the name of the target disk. Enter:

```
AX <mix no> HARRY
```

The system will then ask for the drive the source disk is on. Enter:

```
AD <mix no> DKA
```

The system will then ask for the drive the target disk is on. Enter:

```
AD <mix no> DKB
```

The binary copy utility will now begin executing. When it is finished, enter:

```
RY DKA
RY DKB
```

Special Considerations

For a disk pack system configured as follows, special procedures are required:

```
Loader disk: two 3/6 mini drives, or,
    one 1 MB mini drive
System disk: two 205/206 disk pack drives
```

With this configuration, BINARY.COPY is unable to execute from the system disk if the disk packs are to be duplicated. Thus, to use BINARY.COPY, the MCP must be run from the loader disk drive. This requires the following files to be present on the mini disk:

```
SYSBOOTSTRAP
SYSDSCP
SYSHOSTLOAD
SYSWARMSTART
SYSMCP1
SYSMCP2
SYSICP
SYSMPLII
SYSLANGUAGE
SYSTRANSLATE
```

SYSCONFIG
BINARY.COPY
SYS-SUPERUTL

The mini disk should be B 900 initialized with only 20 files declared. The TL utility and log files must not be on the disk.

The following information must be specified in the SYSCONFIG file to execute BINARY.COPY from a mini disk.

BUFFERMEM	128
DC BUFFER	0
1 MPLII TASK PROCESSOR	128
MAXMIX	4

Output Messages

Message	Possible Causes	Suggested Action
ONE OR MORE NON-BLANK CHARACTERS ARE EXPECTED	A pack-id containing imbedded blanks was entered.	Check input and re-enter.
INVALID CHARACTER IN DISK-NAME <packid> <pointer>	A pack-id containing a character other than A-Z, 0-9 or – was entered. <pointer> is an asterisk appearing beneath the first invalid character.	Check input and re-enter.
SOURCE AND TARGET DISKS ARE NOT THE SAME TYPE	Both disks must be the same type of media.	Use the CMS copy utility to copy unlike disk types.
BINARY.COPY FROM <source disk mnemonic> TO <target disk mnemonic>	Utility is in copy phase.	None, normal operation.
TRACK ZERO ON <disk mnemonic> HAS A BAD SECTOR	An error exists on track zero of specified disk.	If target disk is specified, try another disk. If source disk is specified, use backup disk.
OUTPUT ERROR ON TARGET DISK – TRY ANOTHER DISK	A fatal error was encountered on the target disk.	Try another disk.
TRANSIENT INPUT ERROR IN <start address> TO <end address>	When reading a block of 32 sectors, an error occurred; but when trying to determine which of the 32 sectors has the error, the error did not occur.	Run CHECK.DISK utility against source disk, then use XD utility to delete the bad sectors.
INPUT ERRORS OCCURRED ON THESE SECTORS	This message heads a list of 1 to 32 decimal sector addresses which had errors and were	Run KA utility against target disk. Check list of sectors in error against KA. If sectors are

(continued)

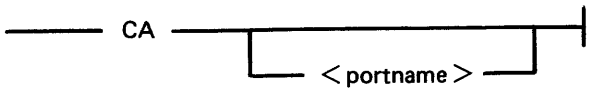
Message	Possible Causes	Suggested Action
	copied as binary zeroes. This message appears on line printer report only.	within file areas, remove files from target disk and recopy from backup disk using COPY utility.
<number> SOURCE DISK INPUT ERRORS ENCOUNTERED	The <number> is the total number of sector addresses listed as in error in the preceding message.	None.
<source/target> DISK IS NOT A CMS DISK	The indicated disk does not contain a standard CMS label.	Replace indicated disk with a CMS disk.
CHECKING TARGET DISK FOR INPUT ERRORS	Utility is verifying target disk.	None, normal operation.
READ CHECK ON TARGET DISK COMPLETE	Indicates verification phase is complete.	None, normal operation.
<source disk mnemonic> TO <target disk mnemonic> COPIED AS <target pack-id>	Indicates copy phase is complete.	None, normal operation.

CA (Candidate)

This is only used for BNA. Refer to the B 900/CP 9500 Burroughs Network Architecture Operation and Installation Guide (form 1163102).

The CA intrinsic allows the user to inquire about the candidate entries on the port candidates for match list.

The syntax for this command is:



where <portname> is a maximum of 17 characters long. Valid characters are upper and lower case A-Z, digits 0-9, the hyphen (-) and the underline ( ).

Output Messages

Message	Possible Causes	Suggested Action
"CA" IGNORED - CANDIDATE LIST EMPTY	Port Candidate for Match list is empty.	Request rejected.
<faulty SCL input> IGNORED - SPECIFIED PORT NOT FOUND IN LIST	Specified port name is not found in the Port Candidates for Match List	Request rejected.
END CA	Entries found on list; end of candidate descriptions.	None.

If entries are found on the list, the response will consist of a report on each candidate.

The local port response is:

```
CANDIDATE <port name> [<may-subport-index>]
MY HOST = <my-host-name>, YOUR HOST = <your-host-name>,
MY NAME = <my-name>, MY USERCODE = <my-usercode>,
YOUR NAME = <your-name>, YOUR USERCODE = <your-usercode>,
PROGRAM = <my-codefile-name>, ON <my-family-name>,
SECURITY = <security-type>, MATCHING STATUS = <matching status>
```

The remote port response is:

```
CANDIDATE <port name>
MY HOST = <my-host-name>, MY NAME = <my-name>,
MY USERCODE = <my-usercode>, YOUR NAME = <your-name>,
PROGRAM = <my-codefile-name>, ON <my-family-name>,
MATCHING STATUS = <matching status>)
```

CONFIGURER (System Configuration Utility)

When the system is warmstarted, its configuration is determined from information found in the SYSCONFIG file.

The CONFIGURER utility allows the user to make a new SYSCONFIG file, modify the existing SYSCONFIG file, print the existing SYSCONFIG file, or display the existing file on the SPO. The SYSCONFIG file is used:

1. By the B 900/CP 9500 WARMSTART code to determine the system's logical configuration as desired by the user.
2. To store parameters that must be maintained across WARMSTARTs.

CONFIGURER is an MPLII program that runs on any CMS system. Its code may be executed from any disk.

CONFIGURER displays messages to the user and accepts responses via the system SPO using the CMS "AX" (accept) intrinsic. These responses determine its actions.

NOTE

The SYSCONFIG file is instrumental in determining how the system resources are allocated for optimal use and performance. Therefore, it is strongly recommended that only those persons familiar with the particular system's physical characteristics be allowed to alter the SYSCONFIG file. A Burroughs field engineer can assist in identifying the system's physical characteristics.

Table 9-25 summarizes the structure of SYSCONFIG.

Table 9-25. Structure of SYSCONFIG File

(D)	Task processor assignments.
(D)	Data comm processor assignments.
(D)	BNA processor assignment.
(D)	SNA Processor Assignment
(D)	B974 Processor Assignment
(D)	Buffer memory allocation.
(D)	Mix limit.
(D)	System operator control flag.
(I)	System message logging on or off.
(I)	Filled log file handling scheme cyclic or non-cyclic.
(I)	Size of each primary log file.
(I)	Size of each secondary system message log file.
(I)	Size of each secondary maintenance log file.
(I)	Disk-name for secondary log files.
(I)	Size of each printer backup file.
(D)	Data comm NDL file name.
(D)	Data comm reconfiguration file-name.
(I)	BNA host name.
(I)	System message dictionary file-name.
(D)	Line printer translation table name.
(I)	System start-up zip text.
(I)	System power off message.
(I)	DC REMOTE SPO MODE CHANGE
	I – system independent. D – system dependent

PRINT

Selecting CONFIGURER's print option causes the utility to find a SYSCONFIG file that has the specified (or default) file name and list its contents on the line printer.

An example of PRINT option's output is shown in figure 9-5. This example shows the default values of the SYSCONFIG file as they are provided with the 3.05 B 900/CP 9500 CMS software release.

Initiation

The initiating syntax for CONFIGURER is as follows:

Operator Enters	System Messages
CONFIGURER	<mix-no>/CONFIGURER COMMENCING <mix-no>/CONFIGURER DISP: <mix-no>/CONFIGURER DISP: DO YOU WANT TO MAKE A NEW SYSCONFIG FILE, PRINT AN EXISTING SYSCONFIG FILE, DISPLAY AN EXISTNG SYSCONFIG FILE, MODIFY AN EXISTING SYSCONFIG FILE, OR END THE CONFIGURER PROGRAM? ENTER MAKE, PRINT, DISPLAY, MODIFY, OR END. <mix-no>/CONFIGURER REQUIRES DATA. USE "AX".
AX <mix-no> <option>	

Entering AX <mix-no> END sends CONFIGURER to end-of-job. The following message informs the user when CONFIGURER has terminated.

<mix-no>/CONFIGURER TERMINATED

When selecting PRINT, DISPLAY, MAKE, or MODIFY, CONFIGURER will ask for a filename. If no filename is specified, the default is "SYSCONFIG". A pack-id may also be specified, or it will default to the system disk pack-id (seven ASCII zeros or SYSPACK). Therefore, if null input is entered for these two fields, the default is "0000000/SYSPACK".

NOTE

The task processor and data comm processors are assigned by WARMSTART as follows: WARMSTART identifies the OS and disk processors, and divides the remainder into those that are I/O capable and those that are not. The I/O capable processors are designated as data comm processors, and the non-I/O capable as task processors.

The data comm processor with the highest physical bus address is assigned logical DCP number 0 (zero). The next highest bus address is assigned DCP number 1 and so on.

WARMSTART also determines how much physical memory is attached to each task processor. All interpreters are assigned to task processors that have greater than 128 kB of memory. Otherwise COBOL and MPLII are assigned alternately. Data comm buffer space is allocated at half the amount of total buffer memory.

```
B 900 CONFIGURER LISTING OF 0000000/SYSCONFIG
          AT 13:43:56 ON 10/20/83


  NUMBER OF TASK PROCESSORS      =    5
    INTERPRETER(S) IN TP 1       = MPLII/COBOL
          MEMORY ON TP 1         = 256 K BYTES
    INTERPRETER(S) IN TP 2       = MPLII/COBOL
          MEMORY ON TP 2         = 256 K BYTES
    INTERPRETER(S) IN TP 3       = MPLII/COBOL
          MEMORY ON TP 3         = 256 K BYTES
    INTERPRETER(S) IN TP 4       = BASIC/MPLII
          MEMORY ON TP 4         = 256 K BYTES
    INTERPRETER(S) IN TP 5       = MPLII/COBOL
          MEMORY ON TP 5         = 128 K BYTES


NUMBER OF DATA COMM PROCESSORS    = 1
     BUS ADDRESS OF LOGICAL DCP 0 = 8


  NUMBER OF BNA PROCESSORS          = 0
  NUMBER OF SNA PROCESSORS          = 0
  NUMBER OF B974 PROCESSORS         = 0
  TOTAL SIZE OF BUFFER MEMORY  = 80 K BYTES
    SIZE OF DC BUFFER SPACE    = 40 K BYTES


  MIX LIMIT = 11

  SYSTEM OPERATOR CONTROL FLAG = LOCAL ODT

  SYSTEM MESSAGE LOGGING = ON

  FILLED LOG FILE HANDLING SCHEME = CYCLIC

  EACH PRIMARY LOG FILE = 32 SECTORS

  EACH SECONDARY SYSTEM MESSAGE LOG FILE = 32 SECTORS

  EACH SECONDARY MAINTENANCE LOG FILE = 32 SECTORS

  DISK NAME FOR SECONDARY LOG FILES = 0000000

  SIZE OF EACH PRINTER BACKUP FILE = 9999 RECORDS

  DATA COMM NDL FILE NAME = 0000000/NDLSYS

  DATA COMM RECONFIGURATION FILE NAME = SYSRECON

  BNA HOST NAME = ' ' (UNDEFINED)

  SYSTEM MESSAGE DICTIONARY FILE NAME = SYSLANGUAGE

  LINE PRINTER TRANSLATION TABLE NAME = NO TRANSLATION

  SYSTEM START-UP ZIP TEXT = <EMPTY>

  SYSTEM POWER OFF MESSAGE = SYSTEM HAS BEEN POWERED DOWN

  DC REMOTE SPO MODE CHANGE = @1B22FFFF@
```

Figure 9-5. Sample SYSCONFIG File

MAKE

Selecting the CONFIGURER utility "MAKE" mode causes the following to occur:

1. A workfile is created to serve as a temporary container for the SYSCONFIG file.
2. The CONFIGURER program gives the operator the option of allowing the program to automatically build SYSCONFIG using the standard (default) values, displaying the standard values that are used in the AUTO mode on the SPO, or manually building a customized SYSCONFIG file.
3. After all fields have been generated, the workfile is closed with remove. It is given the filename which the user specified, or allowed to default to SYSCONFIG.
4. The user is then asked to select another option of CONFIGURER to proceed or end the program.

MODIFY

Selecting the MODIFY option causes the following to occur:

1. Configurer finds the SYSCONFIG file having the specified (or default) filename.
2. A workfile is opened to serve as a temporary SYSCONFIG container, and the contents of the current SYSCONFIG file are copied into the workfile.
3. The user is presented with a numbered list of all the fields in a SYSCONFIG file. The fields to be modified are selected and entered by a list of corresponding numbers.
4. When MODIFY mode is finished, the workfile is closed with remove, and given the specified (or default) file name.

The following paragraphs describe the fields of SYSCONFIG and the values which may be assigned to them while in the MAKE or MODIFY mode. In general, null input is valid and will cause a default value to be assigned to the field in question. Default values are given in parentheses (default =).

The SYSCONFIG file begins with a system dependent area that contains information needed to WARMSTART the physical system. This section specifies the processors and buffer memory.

Number of Task Processors

Enter a number from 0 to 6. Depending on the system there can be up to six task processors on the system. If there is no requirement for the DCP (data comm processor) on the system, the DCP can be used as a task processor. If the user enters 0, the system will assign the task processors at WARMSTART.

Interpreter Assignments

For each task processor enter MPLII, COBOL, BASIC, NW OPTION. The NW (network interface) option loads the SYSBLAISE interpreter for running SNA applications.

MPLII must be assigned to at least one task processor or the utility will loop back and request the task processor assignments again. For each task processor, a decimal number between 67 and 1024 must be entered, representing the number of K bytes of memory the processor will have.

NOTE

The upper limit for the memory in a task processor corresponds to 16 pages of local memory. This reflects the address capacity of the hardware registers. Task processors currently support a practical limit of 512 KB for this parameter. The BNA TP must be configured with 512 KB; any TP currently supporting more than one interpreter must have a minimum of 128 KB.

Number of Data Comm Processors

Enter a number from 0 to 4 to specify the number of processors to be reserved for data comm. If the user selects MODIFY and changes the number of data comm processors, the system will display the amount of total buffer and data comm buffer memory allocated and ask the user if these values must also be changed.

If a data comm processor is assigned, the user must also specify the bus address and the logical DCP number for the processor. The physical bus address relates to the hardware configuration of the system, and requires an integer input from 1 to 15. The logical DCP number is a value that is defined as part of the NDL process. This field requires input of 0 to 7.

Number of BNA Processors

After the information for the data comm processors has been collected, the user has the option of assigning a BNA processor to the system. (Refer to the B 900/CP 9500 Burroughs Network Architecture Operation and Installation Guide, form 1163102). The standard response to the prompt is "NO". If the user does assign a BNA processor to the system, the allowable format for a host name is explained in the prompt. (Default = NO.) See BNA HOSTNAME field later in this section. The system will allocate 512 KB of memory to a BNA processor.

Number Of SNA Processors

The user has the option of assigning a processor as an SNA processor for the purpose of running SNA interface environmental software, in particular the PU.T5 IBM interface program. (Refer to the B 930/CP 9580 SNA PU.T5 Operator's Guide, form number 1173127.) The system will allocate 512 KB of memory to this processor.

Number Of B 974 Processors

The ability exists to dedicate a processor as a B 974 processor, in preparation for an application package not yet released at this PCN. The user may assign from 64 KB to 1024 KB of memory to a B 974 processor.

Buffer Memory

The size specified in this field represents the amount of memory store to be allocated by WARMSTART for system buffer memory use. CONFIGURER requests the total amount required, where 1 KB = 1024 bytes. The portion of buffer space needed for data comm is taken out of this total. The amount entered for data comm buffer space, prompted after total buffer memory is allocated, must be at least 10 KB less than the total. If CONFIGURER detects this is not true, an error message is displayed. A minimum of 2 KB is required for data comm buffers; the maximum value is 1014 KB. As previously mentioned, the default value for DC buffer space is one half of the amount declared for total buffer memory. See table 9-26, Default Values. The user must enter an integer 10 to 1024, or 12 to 1024 if data comm will be run, representing the total kilobytes of buffer memory.

Mix Limit

The maximum mix limit supported is 31 jobs. The user must enter a decimal integer between 1 and 31. No more than this number of jobs will be run in the mix at one time for system execution. (Default = 11).

Table 9-26. Default Values

Available OS Processor Memory	Total Buffers	DC Buffers
4 pages or more	80 kB	40 kB
3 pages	64 kB	32 kB
2 pages	32 kB	16 kB

Control Flag

The control flag field indicates whether or not the system is to allow a message control system (MCS) program to act as the SPO. (This allows a data comm device to be a remote SPO). (Default = local SPO.)

Logging

This field is a flag designating whether system message logging is on or off. The user can enter ON or OFF depending on whether he wishes all SPO messages to be recorded in log files. (Default = ON.)

Log File Handling Scheme

This field is a flag designating whether filled log files are handled in a cyclic or non-cyclic fashion. Cyclic refers to the type of logging where messages could get overwritten when all the primary log files are full. Operator intervention is required to print or transfer the full set of primary log files to prevent losing entries. Non-cyclic refers to the type of logging where more than one holding file is created when all the primary log files are full. No operator intervention is necessary as no entries will be lost. See section 4 for more information on logging. The user must enter CYCLIC or NON-CYCLIC. (Default = CYCLIC.)

Primary Log File

The primary log files are created and maintained by SYS-SUPERUTL. They are labelled SYS-LOG-01 through SYS-LOG-04. The user must enter a filesize in sectors (from 32 to 16,383) for the primary log files. (Default = 32.)

Secondary Log File

The secondary log files, created and maintained by TL, include SYS-LOG-HOLD, SYS-MLGJJJNN and SYS-HLDJJJNN. "JJJ" is the Julian date and "NN" is the number of the log file. SYS-LOG-HOLD is used when CYCLIC logging is being performed; SYS-HLDJJJNN is used when CYCLC logging is running. The user must enter the filesize in sectors (from 32 to 262, 140 sectors). (Default = 32.)

Maintenance Log File

The data in this secondary log file is a record of all maintenance entries, stored in a non-cyclic fashion. That is, a new maintenance file is created when the current one is full. Maintenance entries are not lost or overwritten. SYS-MLGJJJNN is used by both CYCLIC and NON-CYCLIC logging. The user must enter the files in sectors, from 32 to 262,140. (Default = 32.)

Disk Name For Secondary Log Files

The user is prompted to enter the pack-id where the secondary log files (including the maintenance log files) will reside. 0000000 denotes the system disk. (Default = 0000000.)

Printer Backup File

The user should enter an integer between 1 and 999,999 to indicate the number of records each printer backup file will generate. (Default = 9999.)

NDL File Name

This field contains the file name of the NDLSYS file that is to be used in setting up the CMS data comm subsystem. The required input is "NDLSYS" or null input to allow the default value "NDLSYS" to be assigned, as this field is currently not allowed to vary. (Default = 0000000/NDLSYS.)

Reconfiguration Filename

This field specifies the name of the file on the system pack to be used by the MCS as a container for information regarding the data comm subsystem mapping. Currently, this field may not vary; the user must enter SYSRECON, or allow the default value to be used. (Default = SYSRECON.)

BNA Hostname

If the user specified a BNA processor to be configured on the system, this field must contain a valid BNA hostname. The hostname may be a maximum length of 17 characters consisting of letters, digits, underline and hyphen. The name may not have two separators (underlines or hyphens) next to each other. If the user did not specify adding a BNA processor to the configuration, no prompt for hostname is presented. (Default = null input.)

System Message Dictionary Filename

This field holds the name of the file used to generate system messages to the SPO. The user must enter a valid CMS filename specifying the dictionary file to be used. (Default = SYSLANGUAGE.)

Line Printer Translation Table

This field names the translation table to be used for the printer. The user may enter NO TRANSLATION, or from 1 to 20 alphanumeric characters for the translation table name. (Default = NO TRANSLATION.) See table 9-27 for translation table names.

Table 9-27. Translation Table Names

VERSION 1 Australia Canada Korea New Zealand USA VERSION 1 A This version is to be used with IBM compatible RJE application packages: 360/20, 3780, SNA-RJE VERSION 2 Bahamas Barbados Cont'd	VERSION 5 Austria Czechoslovakia Germany Switzerland VERSION 6 Brazil Portugal Portuguese West Africa Portuguese East Africa VERSION 7 Argentina Cont'd

Table 9-27. Translation Table Names
(continued)

Bermuda Central Africa Ceylon East Africa Fiji Ghana Guyana Holland Hong Kong Indonesia Jamaica Kenya Malaysia Malta Netherlands Nigeria Singapore South Africa Taiwan Thailand Trinidad United Kingdom USA Zimbabwe VERSION 3 Algeria Belgium France French Guiana French West Indies Malagasy Republic Morocco Tunisia West Africa Zaire VERSION 4 Italy	Chile Columbia Costa Rica Dominican Republic Ecuador Honduras Mexico Nicaragua Panama Peru Phillipines Puerto Rico Spain Uruguay Venezuela USA VERSION 8/8A Denmark Norway VERSION 9 Yugoslavia (Croatian) VERSION 10 Finland Sweden VERSION 14A Turkey VERSION 16A Iceland VERSION 24 Katakana

These tables are to be used for the B 9249-2/3/4 line printer.

The B 9249-30/50 requires translation only for the following countries:
Denmark, Norway – specify Version 8A.
Turkey – specify Version 14A.
Iceland – specify Version 16A.

The B 924-E printer requires "Version 24 128 char" to be specified.

System Start Up Zip Text

This field contains the program that the CMS SYS-SUPERUTL is to zip as soon as it starts executing (at WARMSTART). The user may enter from 0 to 255 valid ASCII alphanumeric characters that constitute a valid program name. Null input indicates no start up zip text is to be used. (Default = <empty>.)

Power Off Message

This message is displayed on the SPO when the system disk is powered off. It is therefore the last message displayed on operator shutdown of the system. Enter from 0 to 80 valid ASCII alphanumeric characters. (Default = SYSTEM HAS BEEN POWERED DOWN.)

DC Remote SPO Mode Change

This field specifies which key (or keys) must be depressed on this terminal to switch from application mode to SPO mode of the terminal. Refer to the paragraph on ETS. The mode change keys, or flip string must have a length of 1-4 keys, and must be entered in ASCII or hex. ASCII input must be enclosed by quotes and hex input must be enclosed by @. The word "SPCFY" is also valid input, as is null input. Both of these indicate the Specify key is to be used for the ETS flip string. (Default = @1B22FFFF@ which defines the Specify key.)

DISPLAY

Selecting CONFIGURER'S display option will list on the SPO the contents of the SYSCONFIG file with the filename or diskname/filename the user specifies. The format is the same as is shown in the PRINT option example.

DUMPFILE.UTL (System Dump File Segmenter)

DUMPFILE.UTL allows a large dumpfile to be divided into a number of smaller files. This is especially useful when a dumpfile is too large to fit on one disk. The most convenient method of sending a dumpfile to a Burroughs support group is on a mini disk. If the dumpfile is too large to fit on one mini disk, DUMPFILE.UTL can be used to divide the dumpfile into parts. The parts can then be sent on several mini disks.

DUMPFILE.UTL will also reconstruct the file in its original form. The number of files created by DUMPFILE.UTL depends on the size of the original dumpfile.

The syntax for executing DUMPFILE.UTL is:

DUMPFILE.UTL will prompt the user to enter one of two options by displaying the following messages:

```
<mix no>/DUMPFILE.UTL BOJ PR IS A
<mix no>/DUMPFILE.UTL DISP: ENTER P TO PARTITION A FILE INTO PARTS
<mix no>/DUMPFILE.UTL DISP: ENTER R TO RECONSTRUCT A FILE FROM ITS
PARTS
<mix no>/DUMPFILE.UTL REQUIRES DATA USE "AX"
```

The user must respond by entering a "P" or an "R" using the AX intrinsic. For example, AX <mix no> P

File Partitioning

If the user selects file partition, then DUMPFILE.UTL prompts the user to enter the disk-id and file-id of the system dumpfile to be partitioned and the disk-id of the disk which will hold the file partitions by displaying the following messages:

```
<mix no>/DUMPFILE.UTL DISP: ENTER DISK-ID OF SYSTEM DUMPFILE
<mix no>/DUMPFILE.UTL REQUIRES DATA USE "AX"
<mix no>/DUMPFILE.UTL DISP: ENTER FILE-ID OF SYSTEM DUMPFILE
<mix no>/DUMPFILE.UTL REQUIRES DATA USE "AX"
<mix no>/DUMPFILE.UTL DISP: ENTER DISK-ID OF OUTPUT DISK
<mix no>/DUMPFILE.UTL REQUIRES DATA USE "AX"
```

NOTE

The first DISK-ID refers to the disk where the dumpfile is originally located. The user must respond with valid disk-ids and file-ids when requested. DUMPFILE.UTL will display a message for each partition created and a completion message when partitioning is complete. See following example.

```
<mix no>/DUMPFILE.UTL DISP:<OUTPUT DISK-ID>/ DMFILEPART00 CREATED
<mix no>/DUMPFILE.UTL DISP: <OUTPUT DISK-ID>/DMFILEPART01 CREATED
<mix no>/DUMPFILE.UTL DISP: <OUTPUT DISK-ID>/DMFILEPART02 CREATED
<mix no>/DUMPFILE.UTL DISP: <OUTPUT DISK-ID>/DMFILEPART03 CREATED
<mix no>/DUMPFILE.UTL DISP: FILE PARTITIONING COMPLETE
<mix no>/DUMPFILE.UTL DISP: PROGRAM TERMINATING
<mix no>/DUMPFILE.UTL EOJ
```

When the space on the output disk is filled, the user will be prompted to power off that disk and insert a new one.

File Reconstruction

If the user selects file reconstruction, then DUMPFILE.UTL will prompt the user to enter the disk-id and file-id for the recreation of a system dumpfile followed by the disk-id of the disk that contains the system dumpfile parts. For example:

```
<mix no>/DUMPFILE.UTL DISP: ENTER DISK-ID OF DESTINATION
<mix no>/DUMPFILE.UTL REQUIRES DATA USE "AX"
<mix no>/DUMPFILE.UTL DISP: ENTER FILE-ID OF DESTINATION
<mix no>/DUMPFILE.UTL REQUIRES DATA USE "AX"
<mix no>/DUMPFILE.UTL DISP: ENTER DISK-ID OF INPUT DISK
<mix no>/DUMPFILE.UTL REQUIRES DATA USE 'AX"
```

The user must respond with the appropriate disk-ids and file-ids as they are requested. DUMPFILE.UTL will display a message as each dumpfile part is copied to a single system dumpfile followed by a message that indicates file reconstruction is complete. For example:

```
<mix no>/DUMPFILE.UTL DISP: <INPUT DISK-ID>/DMFILEPART00 COPIED
<mix no>/DUMPFILE.UTL DISP: <INPUT DISK-ID>/DMFILEPART01 COPIED
<mix no>/DUMPFILE.UTL DISP: <INPUT DISK-ID>/DMFILEPART02 COPIED
<mix no>/DUMPFILE.UTL DISP: <INPUT DISK-ID>/DMFILEPART03 COPIED
<mix no>/DUMPFILE.UTL DISP: FILE RECONSTRUCTION COMPLETE
<mix no>/DUMPFILE.UTL EOJ
```

Output Messages

Message	Possible Causes	Suggested Action
OUTPUT DISK SPACE EXHAUSTED – ENTER NEW OUTPUT DISK. POWER OFF OUTPUT DISK USING PO COMMAND. THEN INSERT NEW OUTPUT DISK	When partitioning a large dumpfile, more than one mini disk may be required to accommodate all output files created.	Power off all current output disks using PO intrinsic and insert a new output disk.
NEXT FILE IN SEQUENCE IS REQUIRED. INSERT NEXT OUTPUT DISK.	The dumpfile which is being reconstructed has an input file which is resident on another disk.	Power off current input disk using PO SCL command and insert the other output disk.
INVALID ENTRY	This message is displayed if something other than one of the two options ("P" or "R") is entered at the start of the program.	Re-enter AP for file partition or an R for file reconstruction.
ENTERED INPUT TOO LONG	An improper amount of characters has been entered.	Check input and re-enter.
A VALID DISK-ID IS 1-7 CHARACTERS IN LENGTH	More than seven characters have been entered for a pack-id.	Check the pack-id entered and re-enter correctly.

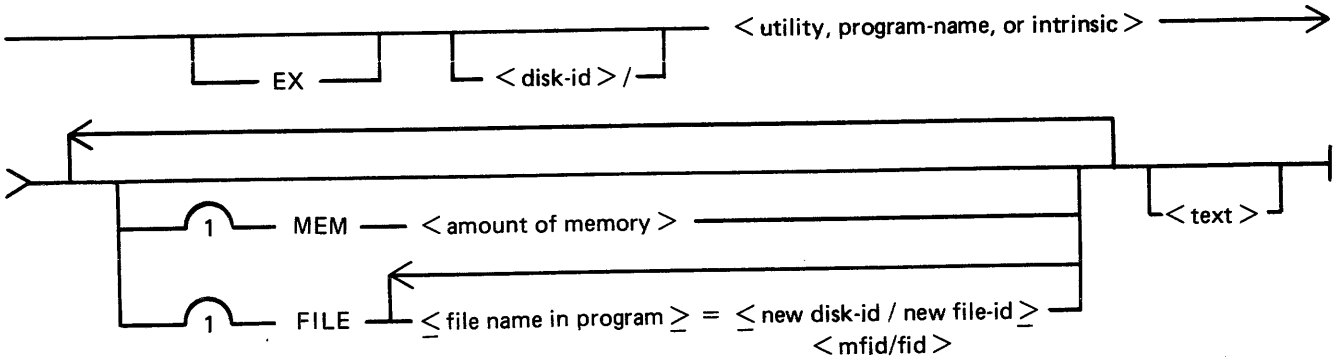
(continued)

Message	Possible Causes	Suggested Action
A VALID FILE-ID IS 1-12 CHARACTERS IN LENGTH	More than 12 characters have been entered for file-id.	Check the file-id entered and re-enter correctly.
NO CHARACTERS IN INPUT	A null input was entered where a pack-id or file-id is expected.	Determine input expected and enter it.
INVALID CHARACTERS IN INPUT	An invalid character was entered.	Check input and re-enter.
A VALID FILE-ID OR DISK-ID MAY BE CONSTRUCTED FROM LETTERS A-Z, THE NUMBERS 0-9, AND THE SPECIAL CHARACTERS HYPHEN (-) AND POINT (.)	Invalid characters have been detected in the pack-id or file-id entered.	Check all characters entered in either the pack-id or file-id and re-enter as necessary.
UNABLE TO OPEN OUTPUT FILE PROGRAM TERMINATING	This message is displayed during file partitioning or reconstruction if DUMPFILE.UTL is unable to open the first output file.	Re-execute DUMPFILE.UTL.
UNABLE TO OPEN FIRST INPUT FILE PROGRAM TERMINATING	DUMPFILE.UTL is unable to open the first input file. (DMFILEPART00)	If complete system dumpfile is still available, then run DUMPFILE.UTL to repartition the file.
FILE NAME DOES NOT MATCH CHECK STRING IN FILE. FILES ARE EITHER CORRUPTED OR OUT OF ORDER	Message appears when reconstructing a dumpfile. A dumpfile part on the user designated disk is corrupted or was not created at the same time as the first dumpfile part.	Ensure that all dumpfile parts were created at the same time.
IRRECOVERABLE ERROR DETECTED IN READ	A fatal error is detected when reading a file.	Re-execute DUMPFILE.UTL to reconstruct or partition dumpfile.
IRRECOVERABLE ERROR DETECTED IN WRITE	A fatal error is detected when writing a file.	Re-execute DUMPFILE.UTL to reconstruct or partition dumpfile.

EX (Execute)

The EXECUTE command has an expanded use on the B 900/CP 9500. It allows the operator to start a program with several options from the SPO. EX can be used when the operator wishes to specify the amount of memory a program will need. The MCP can then assign the program to a processor with enough memory to run it. EX also allows the operator to substitute files before running the program.

The syntax for this command is:



The MEM option allows the operator to specify the amount of memory a program requires. This value is used by the MCP in determining the task processor best capable of executing the program.

The MCP compares the total amount of memory on each task processor with the amount of memory specified by the operator and avoids any processor with insufficient memory. The <amount of memory> is specified in bytes. For example:

EX TESTPROG MEM 30000

The FILE option allows the user to substitute files at program execution time. The user must know the internal file name of the file. This can be found by looking at a source listing of the program.

The FILE option allows a list of file names to be specified. Each file name pair must be separated by a comma, as the following shows:

EX TESTPROG<FILE MYFILE=SYSPACK/YOURFILE, OURFILE=SYSPACK/THEIRFILE>

Example:
To execute the COPY program from a disk called MYDISK, enter the following:

EX MYDISK/COPY MEM 35000 MYPROG TO YOURPROG

The <text> of this statement is "MYPROG TO YOURPROG"

Output Message

Message	Possible Causes	Suggested Action
LOAD FAILURE ON <prog-name> INSUFFICIENT REAL STORE	No task processor with sufficient memory is available.	Re-enter specifying a smaller real store size.

FPP (Field Patch Program)

The B 900/CP 9500 Field Patch Program (FPP) provides a means to update the B 900/CP 9500 release disks B900RL1 and B900RL2 with system patches. This utility requires no special mix and can be run on any CMS system.

After a patch has been applied to a B 900RL1 and B 900RL2 disk, the Replace function of COLD-START must be used to update the system disk with the patched files.

To protect system security, FPP makes no actual changes to the release disk until all the following conditions are true:

1. The following files must be present on the appropriate disk:

	1 MB		5.25 inch Disks		All Other Disks
File Name	**B900RL1**	**B900RL2**	**B900RL1**	**B900RL2**	**B900RL1**
SYSBLAISE	X		X		X
SYSBOOTSTRAP	X		X		X
SYSCOLDSTART	X		X		X
SYSDSCP	X		X		X
SYSHOSTLOAD	X		X		X
SYSICP	X		X		X
SYSINITBOOT	X		X		X
SYSMCP1		X		X	X
SYSMCP2		X	X		X
SYSMPLII	X		X		X
SYSTRANSLATE	X		X		X
SYSWARMSTART	X		X		X
SYSBNACP					X
SYSSNACP					X

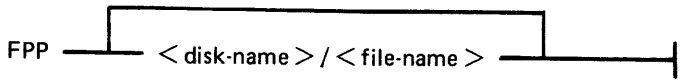
2. Every system file present on the B900RL1 (and B900RL2) disks must be at the patch level specified in the authorization code of the supplied patch.

3. The patch must be entered correctly.

NOTE

SYSBNACP, SYSSNACP and the COBOL interpreter (SYSCOBOL) need not be on the release disk, but if present, they must be at the correct patch level in accordance with condition 2.

The syntax for initiating this program is:



If no <disk-name>/<file-name> is entered, the program will look for a file named FPPPATCH on the system disk which contains the patch information. If this file is not found, FPP will create a temporary FPPPATCH file and accept the data, line by line. If a <disk-name> and <file-name> are supplied, the program will use that file as the FPPPATCH file. The user may create a data or sequence patch file with the following format:

1. The first record of the file must be the authorization code.
2. The subsequent records are the lines of the patch, one patch line per file record. The patch lines may be entered with or without spaces between the hex digit groupings.
3. The last record of the file must contain the word END in columns 1-3.

If a patchfile is supplied in the initiating message, FPP will execute and notify the user when it has completed successfully:

<mix no>/FPP TERMINATED

If the patch is to be entered line by line, the response to "FPP" as displayed on the SPO is:

<mix no>/FPP DISP:ENTER PATCHES
<mix no>/FPP REQUIRES DATA. USE "AX"

The user responds via the "AX" command with the authorization code. This code includes the check code, old version code and new version string. It is provided as the first line of the patch.

The authorization code is constructed in the format:

<check code><old version string><new version string>

where:

<check code>	= two groups of four hexadecimal digits.
<old version string>	= fourteen alphanumeric characters specifying the current release level of the files on the release disk.
<new version string>	= fourteen alphanumeric characters specifying the new release level to be given to the files on the release disk, when all files have been patched successfully.

The authorization code is verified. If not valid, the user is notified and given the chance to re-enter the line.

<mix no>/FPP DISP: RECORD 1 <error message>
<mix no>/FPP DISP: FIXUP PHASE
<mix no>/FPP DISP: <the line as entered>
<mix no>/FPP REQUIRES DATA. USE "AX"

FPP accepts each line of the patch in the same manner, reporting the record number of the created temporary file if invalid, and displaying the incorrect data. The user has the chance to correct his input. If the line is valid, FPP prompts the user to enter the next line of the patch. After the last line of the patch has been entered successfully, the user must supply:

AX <mix no> END

The B900RL1 and B900RL2 1 MB disks are processed in four phases.

Example for 1 MB mini disks:

B900RL1	Authorization Code is entered and verified and B900RL1 system files are checked for correct release level.
B900RL2	The SYSMCP1 and SYSMCP2 files are checked for the correct level. The lines of the patch to be applied to the MCP (those starting with M1 and M2) are read from FPPPATCH or the supplied patch file.
B900RL1	The check codes for any patches to system files on B900RL1 are verified and the patches written to those files.
B900RL2	The check codes for any patches to SYSMCP1 and SYSMCP2 are verified and the patches written.

When the patch has been successfully applied, the following message is displayed:
<mix no>/FPP TERMINATED

Output Messages

Message	Possible Causes	Suggested Action
INVALID MNEMONIC	One or both of the first two characters in a patch line are incorrect.	Check patch and re-enter.
INVALID HEXADECIMAL CHARACTER	A patch line contains some character other than 0-9 or A-F. Does not apply to first two characters in line which are the mnemonic.	Check patch and re-enter.
TOO MANY PATCH LINES HAVE BEEN ENCOUNTERED	More lines than expected by FPP were entered.	Check patch and re-enter.
INVALID NUMBER OF CHARACTERS IN PATCH LINE	The FPP program has encountered more than the maximum 52 characters in a valid patch line, or less than the valid 10.	Check patch and re-enter.
DISK CHECK CODE INCORRECT	The authorization code has an incorrect digit in the first eight characters, or has more or less than eight expected digits for checkcode.	Verify input, obtain correct code from Burroughs representative.
VERSION STRING IS INCORRECT	The authorization code has an incorrect digit in the last 28 characters, or the version string in file is not that of authorization code.	Check version strings on files. Verify input, obtain correct code from Burroughs representatives.
SYSMCP1 IS CORRUPTED	File is corrupted or contains invalid version string.	Verify input was entered correctly, check version string of file, COMPARE file to known good copy or use backup patch disk.
<file> CANNOT BE LOCATED	FPP cannot find file specified on disk specified (or system disk).	Check input, re-enter.
INVALID INITIATING MESSAGE	Patch filename exceeds twelve characters in length.	Check input, rename patch file.
INVALID CHARACTER IN INITIATING MESSAGE	The filename specified was incorrectly formed.	Check SOG for allowable format.
INCORRECT NUMBER OF CHARACTERS	Authorization code requires 36 characters.	Check authorization code.

(continued)

Message	Possible Causes	Suggested Action
LINE CHECK DIGIT DOES NOT COMPARE	The entered check digit does not compare with the internally generated check digit.	Check patch line of code.

Valid Mnemonics For Patches

Mnemonic	Filename
M1	SYSMCP1
M2	SYSMCP2
BT	SYSBOOTSTRAP
BK	SYSBLAISE
BZ	SYSINITBOOT
CK	SYSCOBOL
KS	SYSCOLDSTART
DP	SYSDSCP
HL	SYSHOSTLOAD
MK	SYSMPLII
TK	SYSICP
TR	SYSTRANSLATE
WS	SYSWARMSTART
BN	SYSBNACP
SN	SYSSNACP

GT CF (Get System Configuration)

This command allows the operator to display the system configuration. The following information is displayed on the SPO screen:

Bus address of each processor
Type of processor
Amount of memory on each processor in K bytes
Total amount of buffer memory
Amount of data comm buffer memory

The syntax for this command is:

Each task processor will have displayed the interpreter(s) assigned to it (for example, COBOL/MPL).

The information displayed by this command should match the user SYSCONFIG file. If there are any discrepancies, the operator should request technical assistance.

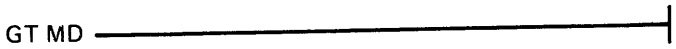
Sample Output:

1.	OS	128
2.	MPLII	256
3.	DISK	32
4.	MPLII	256
5.	COBOL/MPLII	256
6.	COBOL/MPLII	256
8.	DCP 0	64
BUFFER MEMORY		128
DC BUFFERMEM		40

GT MD (Get Full Memory Dump)

This command allows the operator to dump the contents of memory to an internal dump file at will and does not require the system to be rewarmstarted. The dump file can then be printed or copied to another disk to be printed at a later time.

The syntax for this command is:



The MCP will temporarily suspend all jobs in the mix, display the message:
"SYSTEM DUMP INITIATED"
write the contents of memory to the dump file, display the message:
"SYSDMFILEnn CLOSED"
and then resume normal mix activity.

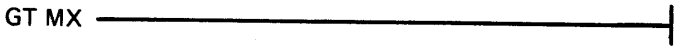
SYSANALYZER must now be run to convert the hex code to a readable format.

Note that GT MD also creates a new dumpfile for a future dump.

GT MX (Get Mix)

Display Mix with Task Processor Assignment

This command allows the operator to inquire about the status of the programs currently processing and the task processor to which they are assigned. The syntax for initiating this command is:



The information displayed is the same as for a normal MX with one exception. The processor in which the program is executing is displayed following the program status. For example,

01/PD B EXECUTING (PROC D)

indicates the utility PD is executing in PROCESSOR D. The letter assigned to each task processor corresponds to the letter displayed on the hexadecimal display lights in Bank A.

Example:
The difference in the response to the two commands is shown below:

```
MX RESPONSE
    01/PD B EXECUTING
    03/CMSCANDE C SUSPENDED WAITING ON ACCEPT
    04/COPY B EXECUTING
    END MX

GT MX RESPONSE FOR SAME MIX:
    01/PD B EXECUTING (PROC D)
    03/CMSCANDE C SUSPENDED WAITING ON ACCEPT (PROC A)
    04/COPY B EXECUTING (PROC A)
    END MX
```

HN (Hostname)

This intrinsic is used for BNA. Refer to the B 900/CP 9500 Burroughs Network Architecture Operation and Installation Guide (form 1163102).

The HN intrinsic allows the user to interrogate the hostname of the local host.

The syntax for this command is:

Output Message

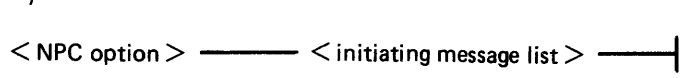
HOSTNAME = <hostname>

NPC900 and NPC900P

The NDL post compilers are MPL programs which take an NDL S-code file as input and generate one or more DCP microcode files. These files are loaded in the appropriate DCP (data comm processor) at Data Comm Subsystem (DCSS) initialization time.

Before executing NPC, the NDL program file must be produced via an NDL compilation.

Syntax



The NPC options are NPC900 and NPC900P. NPC900 is used to generate code for use in a DCP equipped with a DCI line adaptor. NPC900P is used with Line Adaptor II hardware. Both types of line adaptors cannot be used on the same DCP.

The initiating messages are

DBG
LIST
FILE <pack-id>/<file-id>
WORK <pack-id>
STATUS
RANDOM
DCP <dcp list>

More than one initiating message option may be entered by putting a blank space between the selections.

DBG

This is the DEBUG option used for compiler debugging. It causes pertinent DCP run-time information to be dumped to printer. The default for this option is the suppression of the debug listing.

LIST

This option provides for the production of the microcode listing for each of the code files generated. The default for this option is the suppression of microcode listings.

WORK

This option forces NPC work files to specified non-system disks. The default for this option is that the work files reside on system disk.

STATUS

This option causes the NPC to send display messages to the SPO at various times during execution. This option informs the operator that the program is not looping or otherwise hung. It is particularly useful when compiling large files with numerous terminal types. The default for this option suppresses all status messages.

FILE

This is the file equate statement. It allows the user to specify the PACK-ID/FILE-ID of the input file. The <pack-id> and <file-id> must conform to CMS definitions.

NOTE

If the FILE option is used, all generated microcode is on the specified <pack-id>.

Any deviation from the defined syntax causes the termination of NPC.

DCP

The DCP option allows the identification of the logical DCP for which the code file is to be generated. The DCPs are identified by their logical, rather than physical number, as specified in the SYSCONFIG file. <dcp-list> consists of the logical DCP number or several numbers separated by colons.

Example:
DCP (0 2)

RANDOM

The NPC compiler will normally generate a random identification number and insert it in the NDLSYS file and the code file. This number is checked when the code file is loaded into the DCP to ensure that the program is compatible with the current NDLSYS file. The RANDOM option will cause the existing number in the NDLSYS file to be placed in the code file that is about to be generated. This will allow the new code file to be compatible with the code files generated in a previous execution of either NPC900 or NPC900P.

Examples:

1. EX NPC900 FILE NDLCODE
   1) NPC looks for a file called NDLCODE on system disk.
   2) No listing is produced.
   3) All generated DCP files are on system disk.

2. EX NPC900 LIST FILE USER/NDLCODE
   1) NPC looks for a file called NDLCODE on disk "USER".
   2) A listing is produced.
   3) All generated DCP files are on disk "USER".

3. NPC900P FILE NDLSYSA RANDOM DCP (0 1)
   1) NPC900P looks for a file called NDLSYSA.
   2) Code is to be generated only for DCP 0 and DCP 1.
   3) The ID number currently in NDLSYSA will be inserted into the DCP code file(s) now being generated rather than calculating a new number.

Error Messages

The NPC emits error messages when any one of the following conditions occur:

Firmware file exceeds memory size.	Generation of the current DCP file has terminated because NPC has detected that it requires more memory than is available in the DCP (64 kB).
Invalid S.OP	An op-code located in the NDL program is not recognized by the NPC. The NDL program file could be computed, or incompatible levels of NDL and NPC have been used.
Unable to resolve S-level branch address	The NDL program file has been corrupted, or a bug exists in either NPC or the NDL compiler.
Invalid physical line number detected.	The NDL source line section specifies an address outside the range of 1-31.
More than 16 terminal types on file.	Firmware file references more than 16 terminals.

Code suppressed – terminals cannot run on lines	In rare circumstances, due to a total lack of compatibility between line types and terminal types, the NPC will refuse to generate a codefile. For example, if all terminals were full duplex, but no lines specify full duplex, this situation will arise.
Search RAM exceeded.	The NDL program has encountered a hardware limitation on the DCP. This situation should never occur (search RAM maximum size is 128), but if it does, the NDL must be re-written.
Unsuggested S-OP	An NDL S-OP has been decoded and recognized but is unsupported on the B 900/CP 9500.
Unresolved micro-address Subroutine nesting error	These two messages do not usually reflect a problem with the NDL, but are problems with the NPCs. No action is necessary.

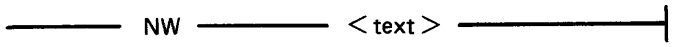
For more information on NPC900 and NPC900P, refer to the NDL Reference Manual (form 1090925).

NW (Network)

This intrinsic is used for BNA. Refer to the B 900/CP 9500 Burroughs Network Architecture Operation and Installation Guide, form 1163102.

The NW intrinsic allows the user to enter text via any BNA network terminal and have this text passed on to the BNA Master Program.

The syntax for this command is:



Output Messages

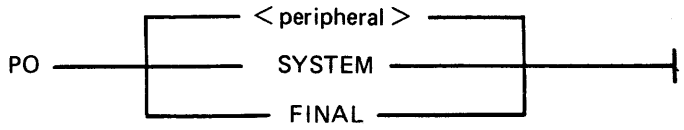
Message	Possible Causes	Suggested Action
"NW" IGNORED – SYSTEM IS NOT IN NETWORK MODE	BNA Master MCS Program (BNAMASTER) is not currently in the mix and logged on as a programmatic SPO.	Request rejected.
"NW" IGNORED – BNAMASTER PROGRAM IS TEMPORARILY UNABLE TO ACCEPT COMMANDS	BNA Master's SPO queue is full.	Request rejected.

The programmatic SPO header will have the input message flag set and the text length, SPO id, SPO tag and SPO size fields primed with the appropriate contents. The system SPO flag will be set if the command came from the system SPO.

PO (Power Off)

The PO command allows the operator to power off the system or user disks.

The syntax for this command is:



Enter PO <peripheral> to power off a user disk.

Enter PO SYSTEM to power off the system, if there are no tasks in the mix (NO PROGRAMS RUNNING).

Enter PO SYSTEM followed by PO FINAL if there are jobs in the mix which can be DSed. These two commands will DS the jobs and bring the system down.

Output Messages

Message	Possible Causes	Suggested Action
"PO" IGNORED – SPECIFIED DEVICE IS THE SYSTEM DISK – USE "PO SYSTEM"	PO <peripheral> was entered with <peripheral> being the system disk.	Use PO SYSTEM to power off system, or check disk name.
"PO FINAL" IGNORED – MUST BE PRECEDED BY "PO SYSTEM"	PO FINAL entered, but it was not immediately preceded by PO SYSTEM.	Enter PO SYSTEM first to power off system disk.
WARNING – PROGRAMS ARE IN THE MIX. ENTER "PO FINAL" TO TERMINATE THEM AND POWER OFF THE SYSTEM. OTHERWISE PO SYSTEM WILL BE IGNORED	PO SYSTEM was entered and a null mix condition did not exist.	See options 1 and 2 following.

Option 1

The user may complete the power off sequence despite the fact that programs are running. To do this, the user should enter "PO FINAL". All tasks in the mix will be DSed and the system will then be powered off.

Option 2

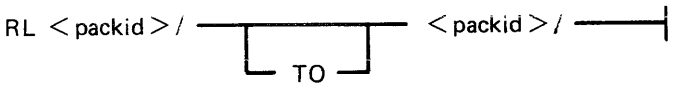
The user may choose to exit the power off sequence and try again later. To do this, the user may enter any other SCL command except PO SYSTEM or PO FINAL. The user may continue processing, or manually terminate the programs and re-enter "PO SYSTEM" command.

NOTE

On a system for which remote operations has been enabled, only the system SPO is permitted to power off the system.

RL (Relabel)

This intrinsic allows the operator to relabel the pack-id of any CMS disk without disturbing the contents of the disk. It also provides a mechanism for changing the name of a pseudo pack. The syntax for this command is:



The system responds with:

<pack-id>/ RELABELLED AS <pack-id>/

RL can also be done as a COLDSTART option.

The condition required for this command to be accepted is:

If a pseudo-pack-id is being relabelled, then all physical units that have pack-ids listed in the PPIT must be physically ready.

When the pack-id to be changed identifies a physical pack, (that is, not a pseudo-pack), this command changes the pack-id in the disk label. Therefore, the new pack-id will be on-line.

When the packid requested to be changed is a pseudo-pack, this command causes the entry for that PACK-ID in the PPIT to be changed to the new pack-id.

WARNING

RL does not scan the disk file header list and will neither notice nor attempt to correct disk file headers of dual pack file components or Key File Parameter Blocks (KFPBs).

Therefore, if a file is located on two disks and one of those disks is relabelled, the disk file header will still locate the rest of the file under the original disk name. Also the keyfile will point to the data file as being located on the original disk name.

Output Messages

Error Message	Possible Causes	Suggested Action
<pack-id> RELABELLED AS <pack-id>	The command was completed successfully.	None, normal operation.
RL <pack-id>/<pack-id> INVALID – DISK NAME INCORRECTLY FORMED	The pack-id specified is not legal.	Re-initiate RL with valid pack-id.
RL IGNORED – DISK <pack-id>/NOT PRESENT	The pack-id specified to be changed is not present.	Re-initiate RL with both valid pack-ids.
RL IGNORED – ALL DISKS BELONGING TO FIXED DISK ASSEMBLAGE MUST BE ONLINE BEFORE COMMAND ISSUED. ENSURE DISKS ARE POWERED ON.	The physical unit from the PPIT is not ready.	RY the specified device and re-initiate RL.

(continued)

Error Message	Possible Causes	Suggested Action
RL IGNORED – DISK <pack-id> ALREADY ON SYSTEM. REMOVE/ RELABEL DUPLICATE DISK OR CHOOSE DIFFERENT DISK NAME AND RETRY COMMAND	A duplicate pack condition is detected.	If possible PO the pack that is not being relabelled.

RLD (Release Level Display)

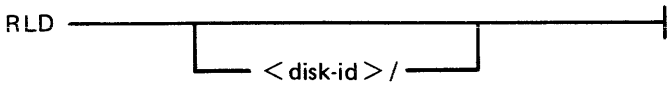
The Release Level Display (RLD) checks whether any of the following system files is present on a specific disk, and reports the release level of each of the following files.

A sample output is:

```
<mix no>/RLD DISP: <disk-id>/SYSBLAISE -       BK - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSBOOTSTRAP      BT - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSCOBOL -        CK - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSCOLDSTART      KS - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSDSCP -         DP - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSHOSTLOAD       HL - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSICP -          TK - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSINITBOOT -     BZ - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSMCP1 -         OS - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSMCP2 -         OS - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSMPLII -        MK - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSTRANSLATE      TR - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSWARMSTART      WS - RELEASE LEVEL 03.05.00
<mix no>/RLD DISP: <disk-id>/SYSBNACP -        UNABLE TO OPEN FILE
<mix no>/RLD DISP: <disk-id>/SYSSNACP -        UNABLE TO OPEN FILE
<mx-no>/RLD DISP: <disk-id>/SYSLANGUAGE        RELEASE LEVEL 03.05.00
```

The information appears on the SPO screen.

The syntax for initiating this utility is:



The disk that RLD is to examine for the system files is identified by <disk-id>. The default is the first release disk (pack-id "B900RL1"). The pack-id for the system disk is "0000000" or "SYSPACK".

The Release Level Display utility searches the disk identified by <disk-id> for system files. Each file search succeeds or fails individually, without affecting others.

For each file, RLD displays one of the following messages:

<disk-id>/<file-id> UNABLE TO OPEN FILE

indicates the file was not present on that disk.

<disk-id>/<file-id> READ ERROR ON HEADER

indicates the RLD utility was unable to extract release level information from the header of the indicated file.

<disk-id>/<file-id> <mnemonic> RELEASE LEVEL XX.YY.ZZ

identifies the disk being examined, the file found and the file release level.

Output Message

Message	Possible Causes	Suggested Action
INVALID PACK-ID	A pack-id was entered that contains illegal characters, more than 7 characters, is not followed by a slash or refers to an absent disk.	Check input and re-enter.

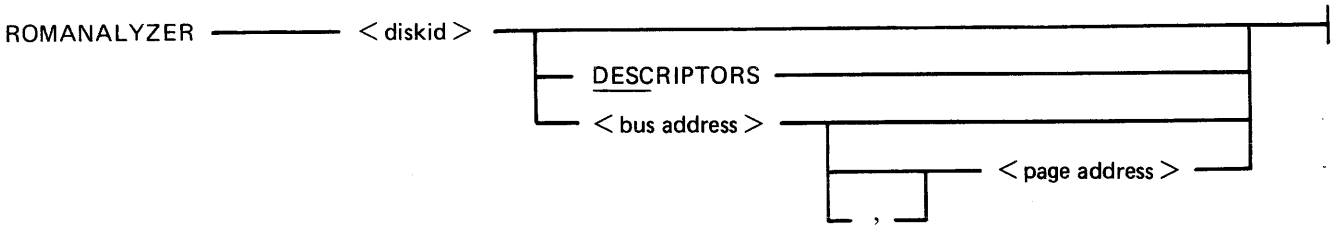
ROMANALYZER (ROM Dump File Analyzer)

NOTE

After taking a ROM dump, ROMCONVERT may be run to convert the dump to a system dump. Then SYSANALYZER may be run instead of ROMANALYZER.

The B 900/CP 9500 ROM Dump File Analyzer, ROMANALYZER, allows the user to print selected parts, or all, of the information written on a removable disk by the Read-Only Memory (ROM) Dump Routine.

ROMANALYZER can be run as a user job on any CMS system. ROMANALYZER communicates with the user via system accepts and displays.



where:

1. <disk-id> = disk-id of the removable disk containing the ROM dump file. The pack-id must identify a removable disk on which the ROM dump routine has created a dump file. No default is used for the dump file.
2. DESC = print only the page descriptors.
3. <bus address> – the ROMANALYZER must restrict itself to the pages of this bus address (see note).
4. <page address> = the page on the specified bus to which the program is to restrict itself (see note).

NOTE

To specify a <bus address>, use a single hexadecimal digit between 1 and F. To specify a <page address>, use a single hexadecimal digit between 0 and F.

If a bus address appears in the initiating message (with or without a page address), ROMANALYZER processes the page(s) involved and issues the following "ACCEPT" message. After processing those pages, the messages:

<mix no>/ROMANALYZER DISP: <BUS><EMPTY>:, <PAGE>
<mix no>/ROMANALYZER REQUIRES DATA. USE "AX"

appear on the screen.

When the ACPT message appears on the SPO, enter one of the following:

AX <mix no> <bus address>

or

AX <mix no> <bus address>, <page address>

ROMANALYZER continues to request additional bus/page data until a null input (AX <mix no>) is entered.

ROMANALYZER Output

The output generated by ROMANALYZER is directed to the line printer and will be in one of the following formats:

1. All descriptors and hex dumps for all pages as contained in a ROM dump file.
2. All descriptors for all pages with no hex dumps.
3. Descriptors and dumps for pages specified from the SPO via bus and/or page addresses.

Page Descriptor Format

For each page of memory in the ROM dump file, ROMANALYZER will provide the following information regardless of the options selected:

1. Bus address of dump
2. Page address of dump
3. Size (last physical address) of dump. (If the size field has a zero value, this means one of two things: either the disk was exhausted while dumping that page, or the disk was removed from the drive prematurely.)
4. Verbal interpretation of I.C. error status (at address @FFFC@).
5. Verbal interpretation of I.C. status 1 (at address @FFFB@).
6. Verbal interpretation of I.C status 2 (at address @FFFE@).
7. Address at which dump of next memory page begins.

If a hex dump is selected, this information will be printed prior to the hex dump for each page.

Format of Hex Dump

Each line of a hex dump that is generated by ROMANALYZER has the following format:

<bus address>/<page address>/<hex memory address><dump> <alphaversion>

where:

1. <bus address> = a single hexadecimal digit representing the bus address of the page being dumped (ranging from 1 to F).
2. <page address> = a single hexadecimal digit representing the address of the dumped page on a particular bus address.
3. <hex memory address> = four hexadecimal digits, representing the address within the specified page of the 64 bytes dumped in the given line.
4. <dump> = a hexadecimal dump of 64 bytes.
5. <alphaversion> = the character representation of the same 64 bytes. A byte that does not form a recognized ASCII character is represented by a blank.

Examples:

ROMANALYZER <pack-id>

ROMANALYZER will print the descriptor and the hex dump for all pages of the ROM dump file.

ROMANALYZER <pack-id> DESC

ROMANALYZER will print only the descriptors for the pages generated on a particular disk.

ROMANALYZER <pack-id> <bus address>

ROMANALYZER will print the descriptors and hex dumps for all pages of a particular bus address.

ROMANALYZER <pack-id> <bus address> <page>

ROMANALYZER will print the descriptor and hex dump for a particular page.

Output Messages

The following messages denote irrecoverable errors encountered during processing of the dump file. Either the dump file was not created properly or has become corrupt.

Message	Possible Causes	Suggested Action
***ERROR – NO INITIATING MESSAGE	A pack-id was not specified in initiating message.	Re-initiate ROMANALYZER with a valid pack-id.
***ERROR – DISK NOT AVAILABLE	Cannot find disk specified or has found a bad disk.	Move disk to different drive and re-initiate ROMANALYZER.
FATAL ERROR – BUS PARAMETER TOO LONG	A space or comma was not found after DESC, DESCRIPTORS, or the bus address.	Re-initiate ROMANALYZER with proper syntax.
FATAL ERROR – PAGE PARAMETER TOO LONG	A single digit page address (0-F) was not found in the initiating message.	Re-initiate ROMANALYZER with valid bus/page address.
FATAL ERROR – INVALID BUS ADDRESS PARAMETER	The bus address specified is not a single hex digit between (1-F).	Re-initiate ROMANALYZER with valid bus address.
FATAL ERROR – INVALID PAGE ADDRESS PARAMETER	The page address specified is not a single hex digit between (0-F).	Re-initiate ROMANALYZER with valid page address.
BUS NOT FOUND	Specified bus address was not found in the dump file.	ROMANALYZER will request a new bus address and continue execution.
PAGE NOT FOUND	Specified page address was not found in the dump file.	ROMANALYZER will request a new page address and continue execution.

ERROR – EOF READ SYSMEM, 1*

ERROR – ERR READ SYSMEM, 1*

ERROR – ERR SYSMEM AT.ADDR + 1*

ERROR – EOF SYSMEM AT.ADDR + 1*

ERROR – ERR SYSMEM INITIAL SEC*

ERROR – EOF SYSMEM INITIAL SECTOR*

ERROR – BAD LEADING SECTOR*

ERROR – LEADING SECTOR NOT ZERO'D OUT*

ERROR – ERR IN FILE READ*

ERROR – EOF IN FILE READ*

ERROR – GET.RECORD READ*

EOF – GET.RECORD READ*

DESC FOUND IN GET.RECORD

ERROR – TRAILING SECTOR NOT ZERO'D OUT*

ROMCONVERT (ROM Dump File Converter)

The ROMCONVERT utility structures the ROM dump file into a format required by the SYSANALYZER utility. The syntax for initiating the utility is:

As a ROM dump may be spread over more than one disk, the utility will request from the operator the number of disks to be converted. The following messages will appear on the SPO:

```
<mix no>/ROMCONVERT DISP:
    ENTER THE NUMBER OF INPUT DISKS(S) <single digit>
<mix no>/ROMCONVERT REQUIRES DATA. USE "AX"
```

The operator responds, via the "AX" intrinsic, with a number from one to nine.
Example:

```
AX <mix no> <number of disks>
```

Next, the utility requests the pack-id of the source disk. The following messages will appear on the SPO:

```
<mix no>/ROMCONVERT DISP:
    ENTER SOURCE DISK-ID
<mix no>/ROMCONVERT REQUIRES DATA. USE "AX"
```

The user responds, via the "AX" intrinsic, with a pack-id (conforming to CMS requirements for a pack-id).

```
AX <mix no> <pack-id>
```

Next, ROMCONVERT will request the pack-id and file-id for the creation of the converted dump file by displaying the following messages:

```
<mix no>/ROMCONVERT DISP
    ENTER DESTINATION DISK-ID
<mix no>/ROMCONVERT REQUIRES DATA. USE "AX"
    (RESPOND WITH "AX" <mix no><PACK-ID")
<mix no>/ROMCONVERT DISP
    ENTER DESTINATION FILE-ID
<mix no>/ROMCONVERT REQUIRES DATA. USE "AX"
    (RESPOND WITH "AX" <mix no> <FILE-ID>")
```

The file-id is a name arbitrarily selected by the operator.

If the user indicated that the ROM dump is resident on more than one disk, the following messages are displayed on the SPO when the utility reaches the end of a disk which is less than the number of disks indicated:

```
INPUT DISK EXHAUSTED
REMOVE DISK USING PO SCL COMMAND
INSERT NEXT DISK AND ENTER SOURCE DISK-ID
```

NOTE

PO SCL command refers to the PO intrinsic.

The user should now power off the input disk, insert the disk on which the dump is continued and then enter the disk-id of the new disk via the "AX" command:

AX <mix no> <disk-id>

The utility will continue until it reaches the end of the file or until another disk is required.

The occurrence of any error sends the ROMCONVERT utility to EOJ.

Output Messages

Message	Possible Causes	Suggested Action
INVALID – RESPONSE MUST BE A SINGLE DIGIT, 1-9	An invalid number was entered.	Check input and re-enter.
NO CHARACTERS IN INPUT	No pack-id specified	Check syntax of pack-id and re-enter.
ENTERED INPUT TOO LONG. A VALID DISK-ID IS 1-7 CHARACTERS IN LENGTH. A VALID FILE-ID IS 1-12 CHARACTERS IN LENGTH	More then seven characters entered for pack-id or more than 12 characters entered for a file-id.	Check input and re-enter.
INVALID CHARACTER IN INPUT. A VALID DISK-ID OR FILE-ID MAY BE CONSTRUCTED FROM THE LETTERS A-Z, THE NUMBERS 0-9 AND THE SPECIAL CHARACTERS HYPHEN (-) AND POINT(.).	Invalid pack-id entered.	Check syntax of pack-id and re-enter.
UNABLE TO OPEN SYSMEM	No disk with the specified pack-id is present.	Place correct disk on line and re-enter.

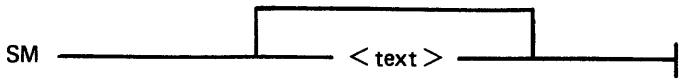
The following messages denote irrecoverable error conditions encountered during processing of the ROM dump file:

INPUT READ ERROR
OUTPUT WRITE ERROR
EOF ON INPUT
TOO MANY BAD AREAS ENCOUNTERED
INVALID NEXT DESCRIPTOR
OS PROCESSOR NOT FOUND
BAD DESCRIPTOR LOCATED
INVALID PAGE
INVALID BUS

SM (Multi-MCS Data Comm)

The SM request causes a data comm message to be placed on the MCS queue.

The syntax for that command is:



The header to the message text contains, among other things, six bytes formatted as follows:

Byte 0	Width of originating SPO (FF if programmatic SPO controls no SPOs or terminals).
Byte 1	Height in number of lines for output display (FF if programmatic SPO controls no SPOs or terminals; 01 if it displays to a printer).
Byte 2	The SPO-ID of the originating SPO.
Bytes 3-4	The tag of the originating SPO.
Byte 5	Flags byte: bit 7 (most significant bit) set indicates the SPO identified in bytes 2-4 is system SPO. bit 7 reset (value = 0) indicates that SPO is not the system SPO. bits 0-6 are unused.

If the program is without a SPO, bytes 0-1 will contain the size of the system SPO.

If the SPO-ID and tag size are RRL type, the SPO.INFO fields in the data comm message header will contain information about the target MCS's originating SPO, or about the target MCS.

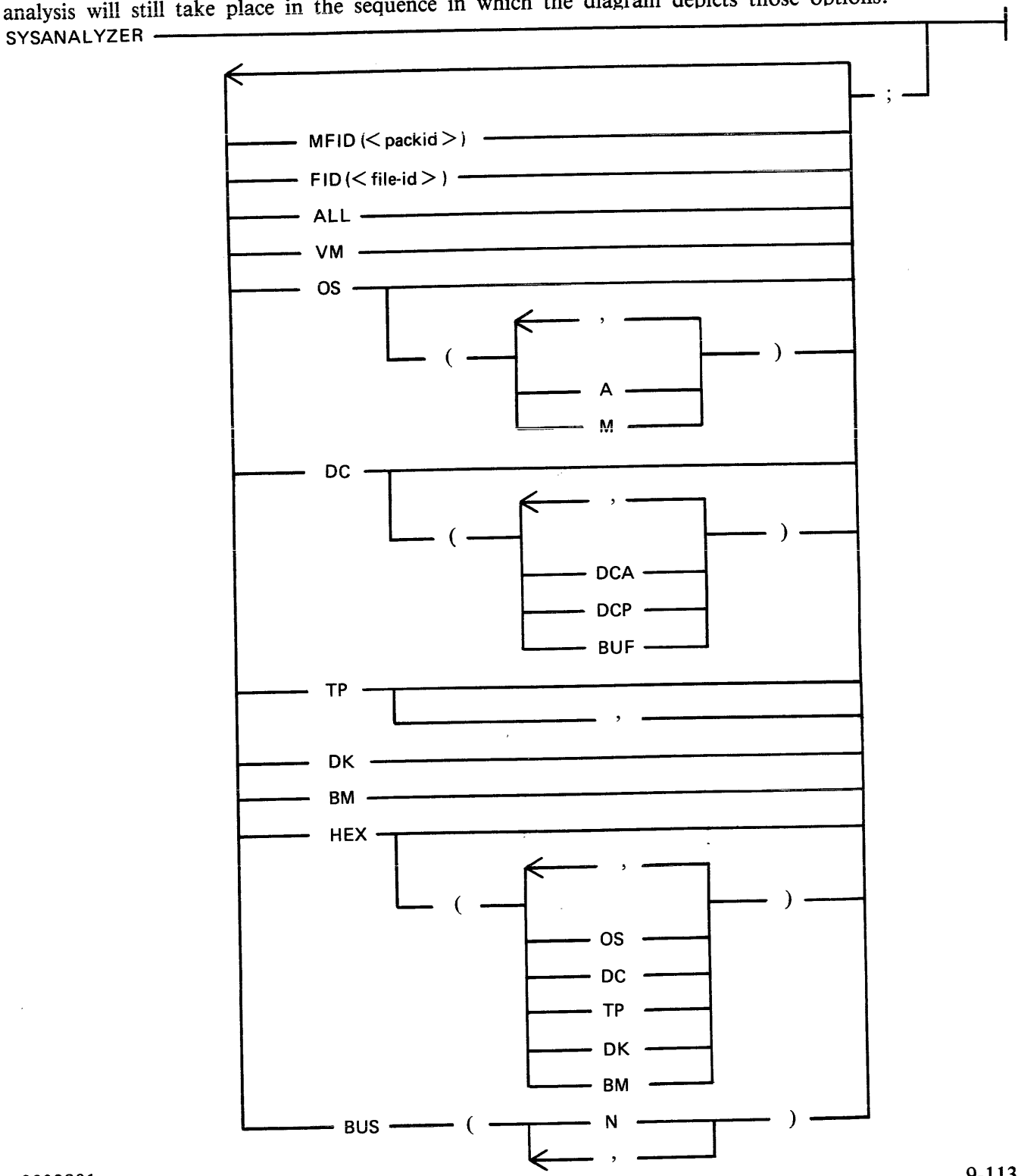
If bit 4 of byte 0 in the SPO.RUN header is set, this request is handled as if the RRL type were really a TID type tag.

If the SPO-ID and tag are TID type, the SPO.INFO fields in the data comm message header will contain information about the SPO that issued the SM request.

SYSANALYZER (System Dump Analyzer)

The B 900/CP 9500 System Dump Analyzer "SYSANALYZER" provides a detailed analysis of a system dump file.

Parameter values are supplied to SYSANALYZER via a single initiating message. The syntax of this message is depicted in the following railroad diagram. The options may be entered in any order. The analysis will still take place in the sequence in which the diagram depicts those options.



Examples:

1. SYSANALYZER. If the initiating message contains no options, the system analyzer uses default values equivalent to the following:
   MFID (<system pack>), FID (sysdmfil00), ALL, HEX;
   To enter only SYSANALYZER, the dump file must be on the system disk and labelled SYSDMFILE00.
   This syntax results in a complete printed listing plus a listing in hexadecimal of the dump file.

2. SYSANALYZER ALL, HEX; this syntax is equivalent to example 1.

NOTE

Only options "ALL" and "HEX" are advised for analysis requested by the user. "ALL" produces a formatted full system analysis and a Hex dump. "HEX" produces a full system dump. The user is advised to ignore the more specific analysis options unless asked to make more specific entries by a technical expert.

If the dumpfile is not labelled SYSDMFILE00 and/or is not on the system disk, either of the following syntaxes may be used:

3. SYSANALYZER FID (THISFIL), MFID (THATPCK), ALL;
4. SYSANALYZER MFID (THATPCK), FID (THISFIL), ALL;

Each of the two initiating messages above will result in the same analysis. The actual sequence of analysis is represented by the order of the options in the second message.

If the user enters the "FID" option and/or the "MFID" option, with no other options following, only the header page and dump file information will be output by the analyzer program.

If the "ALL" option is used, a hexadecimal dump of the disk processor is also included.

The following table defines the various options available when initiating SYSANALYZER. More detailed information on this utility may be obtained from the B 900/CP 9500 Systems Memory Dump User's Guide, form 1118478.

Option	Meaning
MFID( )	The <pack-id> specified is the pack on which the dumpfile resides.
FID( )	The <file-id> specified is that of the dumpfile to be analyzed.
ALL	Perform analysis of: Virtual Memory (see VM) Operating System (see OS) Data Comm Processor (see DC) Task Processor (see TP) Disk Processor (see DK) Buffer Memory (see BM)
VM	Perform analysis of virtual memory.
OS	Perform analysis of operating system. If no suboptions are specified, all information pertaining to OS is given. Suboptions of OS: (A) Activity management (M) Memory associated with OS processor.

(continued)

Option	Meaning
DC	Perform an analysis of each data comm processor. If no suboptions are specified, all information pertaining to DC is given. To restrict analysis to individual DCPs, use the "BUS" option. Suboptions of DC: (DCA) Include analysis of data comm-related parts of operating system. (DCP) Include analysis of data comm-related parts of disk processor. (BUF) Include analysis of data comm-related parts of buffer memory.
TP	Perform an analysis of each task processor. To restrict analysis to individual task processors, use the "BUS" option, described below.
DK	Perform analysis of disk processor.
BM	Perform analysis of buffer management.
VM	Perform analysis of virtual memory.
HEX	Print an unformatted hexadecimal dump of the memory or memories specified. If no suboptions are specified, print a full hex dump of the system. Suboptions for HEX; (OS) Print a hex dump of memory associated with the operating system. (DC) Print a hex dump of memory associated with the data comm processor(s). (TP) Print a hex dump of memory associated with the task processor(s). (DK) Print a hex dump of memory associated with the disk processor. (BM) Print a hex dump of buffer memory.
BUS(n)	Each "n" is a hexadecimal character that corresponds to a given physical bus address. When the BUS option is specified, only the specified bus addresses are considered to be in the dump file.

The "BUS(n)" option allows analysis of the specific task processor(s) or data comm processor(s) that the user wishes to examine. When this option is not used, all task processors are analyzed as a result of TP; all data comm processors are analyzed as a result of DC.

The OS processor memory must be available if the "DC(DCA)" or "VM" option is specified. To do a complete analysis of a DCP when the "BUS(n)" option is used, the bus address of the OS processor must be specified as well as the bus address of the given DCP.

The "BUS (n)" option may be used together with the "ALL" option. This causes a comprehensive analysis, but only of the bus addresses specified.

SYSANALYZER Error Messages

An error detected in the dump file by the analyzer, or an error in the analysis, causes an error message to be generated within the analysis printout. These error messages are bracketed by strings of "$$$" before and after the message (that is, "$$$ ERROR MESSAGE $$$"). Messages not bracketed in this manner are merely warnings.

SECTION 10
B 1800/B 1900 DEPENDENT SYSTEM SOFTWARE

CMS INTRINSICS

With the exception of the following, the intrinsics are as defined in section 3 Common Intrinsics.

SA	Set ASCII – output to all devices for which translation is approved is in ASCII code.
SE	Set EBCDIC – output to all devices for which translation is approved is in EBCDIC code.

For both the intrinsics, the approved devices are line printer and magnetic tape. There are no response messages to this translation switch being set either way. There are no means for the user to determine what the setting is, so, to be sure, the option required should be entered explicitly.

Intrinsics AP, FD, ND, and VF are defined but are not implemented in B 1000 systems.

Intrinsics CTL, HELP, LI, LO and CS have not been defined and are treated as user programs if entered on the SPO.

LT on B 1000 systems is a utility for loading translation tables to a chain type printer – see LT later in this section.

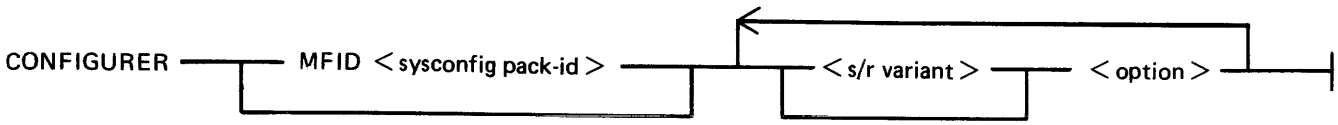
CMS UTILITIES

CONFIGURER

This utility creates and modifies the file "SYSCONFIG". Much of the information in SYSCONFIG is only used at CLEARSTART. However, some options may be acted on in the current session. If an item may have an effect in the current session, this is noted in its description. All information related to the Data Communications Sub-System (DCSS) is used at DCSS initiation. The SYSCONFIG file must be present on the system disk in order to CLEARSTART.

Syntax

The format of the initiating message is as follows:



The MFID option specifies the location of the SYSCONFIG file.

<s/r variant>:

SET	Causes an option to be set on.
RESET	Causes an option to be set off.
SO	Equivalent to SET.
RO	Equivalent to RESET.

If not specified, the default is SET.

<option>:

AREASIZE <num>	Enables the user to specify (in sectors) the minimum size of a disk area when several areas are to be created for a single file. SET and RESET need not be specified.
COUNT	Gives the number of CLEARSTARTs performed since the creation of SYSCONFIG file. Automatic – cannot be set or reset.
CREATE	Enables the user to create the file SYSCONFIG. Any existing file on the required pack is then removed, unless its file type value is not the standard @22@ value. In this case, a warning message is issued (see error messages). The CREATE option is always the first executable option. SET and RESET are not valid for it.
CYCLE	Prevents consolidation of the system messages into secondary log files. When a primary log file is filled up, it is reused, that is, overwritten. Even if CYCLE is set on, all maintenance message entries are consolidated into a secondary file.
DATE TIME	It is advisable to keep these options set on in order that the MCP maintains the correct system date and time.

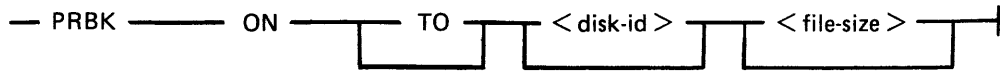
	SO DATE and/or SO TIME cause the date and/or time messages to be displayed on the SPO at CLEARSTART. If DATE or TIME is reset, the current system date and time are stored in SYSCONFIG and used at the next CLEARSTART.
DCPDUMP< num >	This datacomm option takes effect the next time datacomm is initiated in this session, or in a subsequent session. If set, instead of the MCS being loaded, the DCP <num> specified will have its memory dumped to a file called DUMPDCP<num> on disk. When the DCP has been dumped, the option is reset automatically. When set, the message DUMP ON DCP x appears in the DISPLAY, PRINT, or LIST option responses. For the SPO DCP on B 1990 systems, use the option SPODUMP.
DCSIZE <num>	This datacomm option specifies the amount of memory allocated for the datacomm message area in K bytes, at datacomm initiation time, in this session, or in a subsequent session. <num> must be in the range 5 to 310. SO and RO are not valid for this option.
DISPLAY	Causes a description of the SYSCONFIG file to be displayed on the SPO. SET and RESET are not valid for this option.
ERRLOG	This option switches maintenance logging on or off.
HELP	Causes the syntax diagram for CONFIGURER to be listed on a line printer. SET and RESET are not valid for this option.
LIST	Causes a description of the file SYSCONFIG to be either listed or written into a backup file. SET and RESET are not valid for this option.
LOG	Controls whether the logging facility will be initiated at CLEARSTART. SO LOG causes logging to be initiated at CLEARSTART. RO LOG inhibits logging, except the logging of hardware maintenance entries. When this option is reset, log file information is not displayed by a CONFIGURER PRINT or DISPLAY. The message "SPO LOGGING OFF" appears.
LOG.PID <pack-id>	Enables the user to specify the pack on which the log files will reside. SET and RESET are not valid for this option.
MASK @XXXX@	Where @XXXX@ is a 4-digit hexadecimal number representing the Channel Mask. RO MASK supplies the value @FFFF@ to the SYSCONFIG Channel Mask (all channels are thus enabled). SO MASK @XXXX@ gives the value XXXX to the Channel Mask. This represents, from right to left, the numbers of the channels wanted to be used by the system. Example: SET MASK @0203@ enables the MCP to handle only channels 0, 1 and 9. (the mask setting corresponding to the binary setting @0000001000000011@). This setting is stored in the SYSCONFIG file and recalled at each CLEARSTART time unless another mask value has been supplied via the RO MASK option. Note that setting the maintenance panel toggles at CLEARSTART time, with the value @FA@ in the first two digits, and a mask value in the four last digits, takes precedence over the SYSCONFIG mask.

MAXAREAS <num>
: Sets the maximum number of areas per disk file. The maximum value allowed for <num> is 16. SET and RESET need not be specified.

MAXUSERS <num>
: Sets the maximum number of concurrent users of a non-shared disk file. The maximum value allowed for <num> is 63 and the minimum value is 7. SET and RESET need not be specified.

MESSAGE FILE <file name>
: Enables the user to specify the desired message file.

PRBK



Controls how the backup file option is set at CLEARSTART.
<disk-id> denotes the pack on which backup files are to be created and searched for.
<file size> denotes the maximum size of a backup file.
SO PRBK... sets the backup file option at CLEARSTART causing backup files to be created.
RO PRBK... resets the backup file option at CLEARSTART.
When this option is reset, backup file information is not displayed by a CONFIGURER PRINT or DISPLAY.
When creating a printer backup file, the <file size> contained in SYSCONFIG will be used IF AND ONLY IF the file size of the output file originally directed to the printer has a value of 0.
Any other value will be used as a file size for the printer backup file and may cause an unexpected DS/DP of the program when that file size is exceeded.

NOTE

The printer backup options may be changed dynamically at run time using the SO intrinsic.

PRINT
: Generates a listing describing the SYSCONFIG file.
SET and RESET are not valid for this option.

RECONFIGURATION
: This option allows the user to modify the network configuration, using the appropriate MCS commands.

RELEASE.LEVEL
: Gives the release level that will be used for documentation needs. SET and RESET need not be specified.

REMOTECHAR <char>
: Specifies the control character for REMOTE SPO activities.
<char> denotes the desired character.
SO and RO need not be specified.

REMOTEFLAG
: Controls whether remote SPO activities will be allowed or inhibited.
SO REMOTEFLAG sets the REMOTEFLAG at DCSS initialization, allowing REMOTE SPO activity.
RO REMOTEFLAG resets this flag at DCSS initialization, inhibiting REMOTE SPO activity.
When this option is reset, the message "REMOTE SPO OFF" appears in a CONFIGURER DISPLAY or PRINT.

REMOVE
: Transforms all non-conditional CLOSE LOCK into CLOSE REMOVE.
Set this option with caution.

SAVE
: Causes the line printer to be saved when current printing terminates. It takes effect immediately.

SHARED FILES <num>	Sets the maximum number of shared files that can be opened. The maximum value allowed for <num> is 255, and the minimum value is 32. SET and RESET need not be specified.
SPO ADDRESS <num1 : num2>	Specifies the line on which the system SPO resides where num1 = DCP number num2 = line number of the specified DCP. Normally this is 0:0 and should remain so because this is what the stand-alone programs expect. This option may only be specified for B 1990 systems.
SPO DUMP	Causes the memory of the SPO DCP to be dumped to a reserved area during the next CLEARSTART. This option may only be specified for B 1990 systems.
SYS-LOG FILES SIZE <num>	Sets the size, in sectors, of the four primary log files named SYS-LOG-NN (the range of NN is 01-04). <num> is a decimal number from 32 to 16383. The SYS-LOG-NN files are single area files. SET and RESET are not valid for this option.
SYS-HLD FILES SIZE <num>	Sets the size, in sectors, of the secondary system files, that is, the log files created and maintained by TL in order to receive system entries for consolidation. <num> is a decimal number from 32 to 262140. (See NOTE below.) These files are not created if CYCLE is set on. SET and RESET are not valid for this option. NOTE: While CONFIGURER allows setting a file size > 262140 sectors, TL is unable to open such a file, as its size would exceed the maximum CMS file size of 1048560 records. In this case, TL goes to a DS/DP condition, for the following reason: On disks with the allocation unit not equal to 1, the maximum value of 262140 sectors would result in a file greater than the CMS maximum of 1048560 records. For such disks, the maximum size is the number of allocation units rounded down.
SYS-MLG FILES SIZE <num>	Sets the size, in sectors, of the secondary maintenance files, that is, the log files created and maintained by TL and which receive the maintenance entries and the hardware configuration area. <num> is a number from 32 to 262140 (see note for SYS-HLD FILES SIZE). These files are created for consolidation whether CYCLE is set on or off. SET and RESET are not valid for this option.
TEACH	Causes the syntax diagram for CONFIGURER to be displayed on the SPO. SET and RESET are not valid for this option.
TIME	See DATE option.
WRITE	Causes a backup file to be created with the description of the SYSCONFIG file. SET and RESET are not valid for this option.
ZIP ["<text>"]	Indicates a message, <text>, to be ZIPPED after CLEARSTART.

"<text>" denotes the message to be ZIPPED and must be a quoted string.
SO ZIP "<text>" sets the message to be ZIPPED.
RO ZIP resets the message to null. When this option is reset, no ZIP message will appear in a CONFIGURER PRINT or DISPLAY.
This facility is very convenient as it can be used to zip an MCS automatically after each CLEARSTART.

The SAVE FACTOR output by options PRINT or LIST gives the compile date of the version of CONFIGURER which created that SYSCONFIG.

Defaults

The values of the SYSCONFIG file included in the 3.05 release are as follows:

SPO LOGGING	ON
CYCLIC LOGGING	OFF
PRIMARY LOG FILES SIZE	256 @100@
SECONDARY LOG FILES SIZE	256 @100@
MAINTENANCE LOG FILES SIZE	256 @100@
MESSAGE FILE NAME	SYSLANGUAGE
DATE	ON
TIME	ON
RELEASE LEVEL	3.05.00
REMOVE OPTION	OFF
ERRLOG OPTION	ON
SAVE DEVICE OPTION	OFF
REMOTE SPO	ON
RECONFIGURATION	ON
REMOTE CHARACTER	*
CLEAR START COUNT	0
DC BUFFER SIZE	32 @20@
AREA SIZE	256 @100@
BACKUP DESIGNATION	0000000
BACKUP FILE SIZE	10000 @2710@
NUMBER OF SHARED FILES	32
MAX USERS	63
SPO ADDRESS	0:0
SPO DUMP	ON

Other datacomm information is reset.
ZIP information is reset.
The backup option, PRBK, is reset.

Update Messages

NOTE

If the option you are entering is already set, the response message is: "<packid>/SYSCONFIG" OK.

REMOTE FLAG SET [RESET] (REMOTE CHARACTER : "x")
RECONFIGURATION FLAG SET [RESET]
REMOTE CHARACTER "x" CHANGED TO "y"

DC BUFFER SIZE CHANGED FROM xx K-BYTES TO yy
MESSAGE FILE NAME "xxx" CHANGED TO "yyy"
SPO LOGGING SET [RESET] (LOG FILES SIZE : xxxx SECTORS
HLD FILES SIZE : 256 SECTORS, MLG FILES SIZE : 256 SECTORS)
DUMP DCPx SET [RESET]
BACKUP OPTION SET [RESET] (BACKUP DESIGNATE DISK : "<packid>",
BACKUP FILE SIZE = xxxxx RECORDS
If "TO <diskid>" option is specified, the response message is: BACKUP DESIGNATE DISK "<packid>" CHANGED TO "<diskid>"
If <file size> option is specified, the response message is: BACKUP FILE SIZE CHANGED FROM xxxxx RECORDS to yyyyy RECORDS
ZIP MESSAGE RESET
ZIP MESSAGE CHANGED TO " < zip text > "
CHANNEL MASK CHANGED FROM < xxx > TO < yyy > [RESET]
REMOVE OPTION SET [RESET]
ERROR LOGGING OPTION SET [RESET]
SAVE DEVICE OPTION SET [RESET]
CYCLIC LOGGING SET [RESET]
MAX. USERS CHANGED FROM < xxx > TO < yyy >
MAX. AREAS CHANGED FROM < xxx > TO < yyy >
MAX. NUMBER OF SHARED FILES CHANGED FROM < xxx > TO < yyy >
SPO DUMP SET [RESET]
SPO ADDRESS x:x CHANGED to y:y

Error Messages

< mix/prog > CANNOT OPEN < mfid/fid > FILE TYPE MISMATCH
< mix/prog > CANNOT OPEN < mfid/fid > PACK NOT PRESENT
< mix/prog > CANNOT OPEN < mfid/fid > DIRECTORY FULL
< mix/prog > CANNOT OPEN < mfid/fid > INVALID FILE NAME
< mix/prog > WAITING OPEN ON < mfid/fid > IN USE
< mix/prog > CANNOT OPEN < mfid/fid > DISK LOCKED
< mix/prog > CANNOT OPEN < mfid/fid > BAD BLOCK OR RECORD SIZE
< mix/prog > CANNOT OPEN < mfid/fid > BAD FILE SIZE
< mix/prog > CANNOT OPEN < mfid/fid > FETCH VALUE :@XXXXXX@
< mix/prog > WAITING CLOSE ON < mfid/fid > DUPLICATE FILE
< mix/prog > CANNOT CLOSE < mfid/fid > FILE TYPE MISMATCH
< mix/prog > CANNOT CLOSE < mfid/fid > FILE IN USE
< mix/prog > CANNOT CLOSE < mfid/fid > DISK LOCKED
< mix/prog > CANNOT CLOSE < mfid/fid > FETCH VALUE @XXXXXX@
< mix/prog > DATA ERROR ON READ FROM < mfid/fid > -RECORD XX
< mix/prog > INVALID KEY ON READ FROM < mfid/fid > -RECORD XX
< mix/prog > CANNOT READ < mfid/fid > -RECORD XX FETCH VALUE @XXXXXX@
< mix/prog > PARITY ERROR ON WRITE TO <mfid/fid > -RECORD XX
< mix/prog > NO USER DISK TO CREATE < mfid/fid >
< mix/prog > CANNOT CREATE < mfid/fid > - DISK LOCKED
< mix/prog > INVALID KEY ON WRITE TO < mfid/fid > -RECORD XX
< mix/prog > CANNOT WRITE < mfid/fid > -RECORD XX FETCH VALUE @XXXXXX@
< mix/prog > INVALID NUMBER OF LOG FILES

< mix/prog > INVALID LOG FILE SIZE
< mix/prog > INVALID DCP NUMBER
< mix/prog > INVALID DC BUFFER SIZE
< mix/prog > INVALID BACKUP FILE SIZE
< mix/prog > INVALID REQUEST (< option >)
< mix/prog > "< mfid/fid >" VERSION NUMBER MISMATCH – TRY ... "CREATE" OPTION

These messages are self explanatory.

When a SYSCONFIG file already exists on a pack and an attempt is made to CREATE a new SYSCONFIG file on the same pack, the previous one will be removed unless its file type value is different from @22@. In this case the following messages are issued:

< mix/prog > ** WARNING ** < fid > ALREADY PRESENT ON < mfid > FILE TYPE IS @XX@

< mix prog > ACPT (ANSWER "OK" OR ELSE)

By answering "OK", the user gives CONFIGURER the authorization to remove the previous SYSCONFIG file.

By entering any other answer, he prevents the removal of the previous SYSCONFIG file.

Warning Messages

< mfid/fid > REMOVED
< mfid/fid > CREATED
< mfid/fid > UPDATED
< mfid/fid > LISTED
< mfid/fid > OK

DC.INFO

The DC.INFO utility provides the user with a means of getting in-flight information about any of the DCPs which have been loaded.

Invoking this program creates a Dump file of the requested DCP memory, which can be analyzed by the DCP.ANALYZER utility.

Syntax

DC.INFO DUMPDCP <num>

where <num> represents the logical number of the DCP. Its value is an integer from 0 to 4, which is assigned in order of the occurrence of the DCP in the Channel Table, that is, the channels are scanned in numerical order (0 to 15) and the first DCP found becomes DCP 0, and so on.

Error Messages

Syntax errors cause the following message to be displayed:

VALID INITIATING MESSAGE IS DUMPDCP X

If the DCP for which a dump is requested is not connected, the error message is:

INVALID DCP NUMBER

If the DCP for which the dump is requested has not been loaded, the error message is:

CANNOT DUMP DCP<X> NOT IN USE

Completion Message

After the dump file has been successfully created, the message:

DUMPDCP<num> created

is displayed and the dump can be analyzed via DCP.ANALYZER:

DCP.ANALYZER FILE DUMPDCP<num> DUMP (see DCP.ANALYZER notes)

DCP.ANALYZER

A Data Comm problem can arise from a problem in the DCP itself. A DCP dump must then be taken and properly analyzed by DCP.ANALYZER.

How To Take A DCP Dump

There are three ways to get a DCP Dump:

1. Take an "in-flight" DCP Dump by invoking DC.INFO (see DC.INFO notes).

2. Set DCPDUMP in SYSCONFIG (see CONFIGURER notes). In this case, the dump(s) will be taken at the next Data Comm initialization time. This will cause DCCH to go to End of Job without executing the requested MCS. The options corresponding to the relative position of the DCP dumped will be reset in SYSCONFIG, and it is the responsibility of the operator to restart the Data Comm Sub-System.

3. On a B 1990 system the SYSCONFIG option SPO DUMP should be set (see CONFIGURER notes). In this case, a dump of the SPO DCP will be taken at the time of the next CLEAR-START. The dump is written to a reserved area of the MCP file and may later be transferred to a file DMFILjjjjjnn using the SDA utility.

   Other DCPs on a B 1990 may be dumped using method 2.

NOTE

If DCPDUMP is to be taken after a system hang or halt:

- on conventional B 1000 systems, do not press CLEAR button, press START button only.
- on B 1990 systems do not use MTR GO, use TAPE GO.

How To Analyze A DCP Dump

DCP.ANALYZER analyzes dumps created from either DCP-1, DCP-3 or DCP-4. Two modes of operation are provided: interactive mode and batch mode.

Interactive mode is invoked by not specifying an initiating message and may be used from either the SPO or a terminal. DCP.ANALYZER may be used with either a GEMCOS or TMCS interface.

The user may examine (display) and/or print selected portions of dump information using the commands provided.

Syntax

DCP.ANALYZER (from the SPO)

or
<MCS control char> DCP.ANALYZER (from a terminal)

Two kinds of reserved words are used by DCP.ANALYZER:

1. The Commands to specify an action to be executed, usually a file related function.
2. The Formatters to generate output formats.

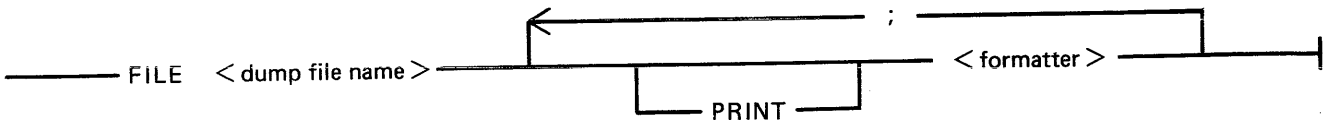
Commands and formatters are entered by using AX messages via the SPO or directly from a terminal.

If the output takes more than a page, then "..." appears at the bottom left hand corner of the screen, and the user must send a blank input to get the next page. For a terminal the cursor is positioned to send one space character.

A HELP function will be invoked if an entry, not recognized as a reserved word, is entered.

In batch mode, entered by specifying an initiating message, all output is printed. The user can choose which portions of the dump information are to be printed, or he may print the entire dump.

The initiating message syntax is:



where <dump file name> is the name of the DCP dump file to be analyzed and <formatter> is one of the formatters defined below.

Example:

To print the entire dump of DCP 0:

```
DCP.ANALYZER FILE DUMPDCP0 DUMP
```

Commands and Formatters

BYE	CONFIGURATION
CLOSE PRINTER	DUMP
DCP-1	LIEGE.HDR
DCP-3-4	LINE
DISPLAY	LINE.HISTORY
END	LINE.INFO
FILE	LINE.LINKAGE
FORMS	LINE.TABLE
GET	LIST.AVAIL
HELP	LIST.MSG
PRINT	LIST.QUEUE
WHAT	LIVM.HDR
	NETWORK
	RESV.MEM
	STATION
	STATION.HISTORY
	STATION.TABLE
	SUMMARY

Each command and formatter is defined below.

Where a numeric parameter or address is required, the value may be specified in two ways:

1. Decimal, for example, 255.
2. Hexadecimal, by specifying the hexadecimal value IMMEDIATELY after the special character "@", for example, @FF.

BYE

This command is used to exit the interactive mode.

CLOSE PRINTER

When a print is requested, the analyzer opens a printer file and keeps it open until End Of Job. The CLOSE PRINTER command allows the user to close the printer file without terminating the analyzer job.

CONFIGURATION

This formatter outputs the hardware line configuration of the DCP. It is a table describing all the line adapters found on that DCP (some of them may be unused by the NDL).

DCP-1

This command is used when the program cannot determine the type of dump file to be analyzed. The DCP-1 type will be forced.

DCP-3-4

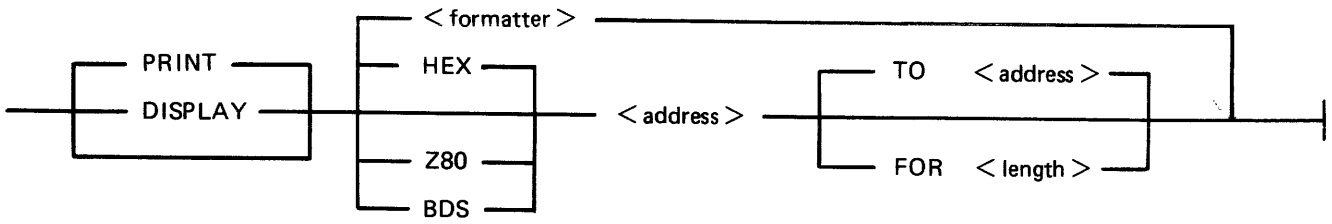
This command is used when the program cannot determine the type of dump file to be analyzed. The DCP-3 or DCP-4 type will be forced.

DISPLAY

This command may be used to display the dump file contents in hexadecimal or disassembled micro code.

DISPLAY <formatter> used in interactive mode will display the selected format but is redundant, since entering <formatter> only will give the same results.

The full syntax, common for DISPLAY and PRINT, is:



The type code HEX, Z80 or BDS specifies the format desired:

HEX for hexadecimal memory contents.
BDS for disassembled DCP-1 code.
Z80 for disassembled DCP-3 or DCP-4 code.

The default is the last type used, or HEX if no type has previously been specified.

The dump range may be specified from a start address to an end address or from a start address for a specified number of bytes. If only a start address is specified, a length of 255 bytes is assumed.

DUMP

This formatter will output all information for the file loaded. It successively executes the following formatters:

SUMMARY
CONFIGURATION
NETWORK
LINE.LINKAGE
RESV.MEM
LIST.AVAIL
LINE for each line declared in the NDL for the related DCP

END

This command is used to exit the interactive mode.

FILE <file-name>

This command is used to open the specified file and select it as a DCP-1 or DCP-3 dump file with the message "DCP-n DUMP FILE LOADED". If the dump type cannot be determined, the message "CANNOT DETERMINE DCP DUMP TYPE" is displayed and the dump type can be selected by the user via the use of the DCP-1 or DCP-3-4 commands.

If the file cannot be opened, the message "CANNOT OPEN <fid> [<n>]" is output where <n> is the CMS event number indicating the reason for the failure.

FORMS

This command will display all the formatters available, adding DCP-1 and DCP-3-4 to the list.

GET <file-name>

This command is the same as the FILE command.

HELP

This command will interactively display explanations on the analyzer program, all the available commands and formatters.

LIEGE.HDR <addr>

The Liege header found at the specified address is output. The Liege header is B 1000 systems implementation dependent.

LINE <lln>

LINE <memory address>

This formatter takes a parameter which is either a logical line number or the memory address of the line info table. The line info table, the line table and each attached station table is provided.

LINE.HISTORY <lln>

All message headers found in the available list referring to the logical line number specified are output. Since the buffers are allocated cyclically, these messages provide a history of activity on the line.

Following the messages, the line info table, line table and station tables for each attached station are output.

LINE.INFO <lln>

LINE.INFO <memory address>

The parameter may specify either a logical line number or the memory address of the required line info table. If memory address is specified, the user must ensure that it addresses a valid line info table.

The line info table format printed is B 1000 systems implementation dependent.

LINE.LINKAGE

This formatter outputs the structure of links between ready lines.

LINE.TABLE <lln>

LINE.TABLE <memory address>

The parameter may specify either the logical line number or the memory address of the required line table.

If a memory address is specified, the user must ensure that a valid line table address is specified.

The format of the line table printed is as specified for the NDL S-Machine and is documented in the Data Comm Subsystem Reference Manual, form no 1090909.

LIST.AVAIL

This formatter outputs the list of all available buffers in the DCP memory. When present, the message header of each buffer will be output.

Since the available buffer pool is allocated cyclically, the list of available buffers is a dynamic history of messages in the approximate order that they were processed.

LIST.MSG <addr>

The message header (LIEGE.HDR and LIVM.HDR) and the beginning of the message text found at the specified address is output. If the message is longer than 61 characters, there are a few "..." at the end of the line.

LIST.QUEUE <addr>

This formatter outputs the list of all the messages in the queue located at the specified address.

LIVM.HDR <addr>

The LIVM header found at the specified address is output. The LIVM header is defined for the NDL S-machine, and is documented in the CMS Data Comm Subsystem Reference Manual, form no 1090909.

NETWORK

This formatter provides information on the line configuration defined in the NDL program for the related DCP.

PRINT

This command is similar to the DISPLAY command but the output generated is directed to a printer.

Refer to DISPLAY for the complete syntax.

PRINT <formatter> option used in batch mode will print the selected format but is redundant, since entering <formatter> only will give the same results.

RESV.MEM

This formatter will output information regarding the state of the DCP and HOST interface at the time of the dump.

The reserved memory format printed is B 1000 system implementation dependent.

STATION <lsn>

STATION <memory address>

The parameter may specify either a logical station number or the memory address of a station table.

If a memory address is specified, the user must ensure that it is a valid station table address.

The station table is output together with any messages queued in the station queue.

STATION.HISTORY <lsn>

All message headers in the available buffer pool referring to the logical station specified are output. Since buffers are allocated cyclically, the list of messages provides a history of station activity.

Following the messages, the station table and station queue are output.

STATION.TABLE <lsn>

STATION.TABLE <memory address>

The parameter may specify either a logical station number or the memory address of the required station table.

If a memory address is specified, the user must ensure that it is a valid station table address.

The selected station table is output.

SUMMARY

This formatter outputs information regarding the generation of the DCP code file and also about the physical configuration of the lines on the DCP.

WHAT

This command will give information about the version level of the DCP.ANALYZER used and about the dump file analyzed.

DISKDUMP

The aim of the DISKDUMP utility is to make the B 1000 system more secure and to allow the user to install a straightforward recovery procedure. For that purpose it enables the user to get a quick on-line backup of a CMS disk. This backup is created on another CMS disk, but can transit via a tape.

Allowed disk devices are DP205, DP206, DP207, DP223, DP225.

Any program will execute normally provided it resides and accesses files on a disk other than the disk used as input to DISKDUMP.

For programs residing or accessing files on the input disk:

- if they are started before DISKDUMP, they may go on executing as long as they do not access SYSMEM, as SYSMEM is locked during the DISKDUMP run (so no access is allowed to the available table or disk file headers for modification of area allocation).

- an attempt to start them during the DISKDUMP execution will fail and the message "DISK <packid> IS CURRENTLY LOCKED BY ANOTHER PROGRAM" will be displayed.

The utility works in three phases:

1. A dump phase.
2. A comparison phase, which is automatically performed if I/O errors occur during the dump phase and which can be requested otherwise.
3. A copy phase which is executed if comparison errors are detected during the comparison phase (see paragraph entitled comparison errors).

Three functions are provided:

1. DSKDSK for a copy from disk to disk.
2. DSKMTP for a copy from disk to tape.
3. MTPDSK for a copy from tape to disk.

To each one of these functions may be added the comparison function:

AND COMPARE

The input file is named <input device id>/SYSMEM, the output file is named <output device id>/SYSMEM.

NOTE

As DISKDUMP works on a sector basis, one record represents one sector.

DSKDSK

The data in every sector of the input disk is moved into the corresponding sector on the output disk. When completed, both disks are identical, except for the output disk serial number, initialization date and owner's identification fields which are preserved if it was a CMS disk. In addition, the output disk name may be specified in the initiating message.

For error handling, see the following paragraphs.

DSKMTP

Every sector of the input disk corresponds to a record on the output tape. A bad sector on the disk produces a tape record filled with the error pattern "BADBAD...".

MTPDSK

Every record on the input tape is moved into a sector on the output disk. First data record on tape must be a copy of the CMS disk label.

A bad sector on the output disk is skipped, and the corresponding record from the tape is not copied.

AND COMPARE

Every record on the output device is compared with the corresponding record on the input device.

This phase is automatically invoked if it is specified in the initiating message, or if I/O errors occur during the dump phase. If not, the user is asked if he wants a data comparison:

DATA COMPARISON CHECK ? <YES – NO>

During the comparison phase, DISKDUMP memorizes the numbers of the records affected by I/O errors and identifies the files that were not copied correctly.

Unlabelled Open

If the output device is a disk, an unlabelled open is performed on it. If the input device is a disk, and the COMPARE option is specified, an unlabelled open is performed if it is not the system disk.

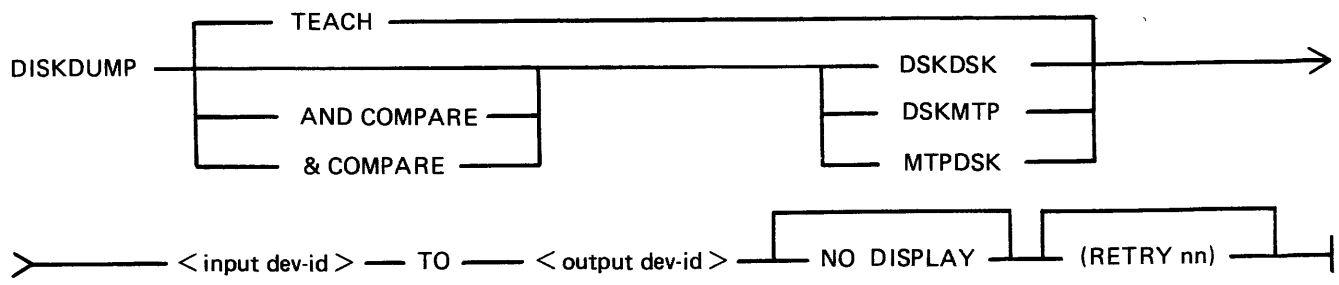
An unlabelled open in this context requires the operator to reserve the disk using the RD command, then assign it to DISKDUMP using the AD command.

The following messages are displayed in this case:

< mix/DISKDUMP > : SELECT <input – output> DEVICE, USING THE ...
... FOLLOWING COMMANDS :

< mix/DISKDUMP > : RD <DEVICE MNEMONIC>
< mix/DISKDUMP > : AD <MIX> <DEVICE MNEMONIC>

Program Initiation



NODISPLAY prevents the display of the COMPARISON error messages.

(RETRY nn) enables the user to set the maximum number of retries when a recoverable input error occurs. nn is an integer among [1, 2, 4, 8, 16, 32, 64]. If another integer, or no integer is specified, the number of retries is 24.

The TEACH option shows a simplified initiating message, omitting the TEACH, NODISPLAY and RETRY options.

BOJ Messages

<mix>/DISKDUMP B1900 VERSION [<compile date>] BOJ AT hh:mm:ss

<mix>/DISKDUMP INPUT DEVICE = " device-id SYSMEM " <peripheral type>
OUTPUT DEVICE = " device-id SYSMEM " <peripheral type>

Error Handling And Error Messages

Bad Sector Count In Label

If the Bad Sector Count field in the label of an input or output disk is not zero, the program will display a message asking whether or not the operator wishes to continue. To continue, the operator replies using the AX command. If the reply is "N" the program goes to End Of Job.

Comparison Errors

Any discrepancy found between two records during the comparison phase causes the utility to display the following messages:

< mix/prog> COMPARISON ERROR AT RECORD <xxxx> STARTING AT BYTE <yy>

< mix/prog> : INPUT (OUTPUT) ERROR ON < pid > AT SECTOR <xxxx>
..... AFFECTS FILE < filename>.

After the comparison phase, the files on which one or more comparison errors have been detected are copied sector by sector onto the output device (this is the copy phase). If these files were IN USE during the dump phase, the operation will only be performed if they are no longer IN USE during the copy phase.

If an input error occurs, retries are attempted and warning messages are displayed (see next paragraph). If these retries fail, the input sector is bad and the output sector is filled with the error pattern "BAD-BAD..."

Recoverable Input Errors

When an input error occurs, retries are attempted until one of them is successful or until the maximum number of retries allowed is reached. The maximum number is 24, unless otherwise specified in the initiating message.

If the input operation is successful after the retry, the following message is displayed:

< MIX/DISKDUMP > : ERROR OVERCOME ON " < mfid/fid > " ...
... (SECTOR < nnnn>) AFTER < nn > RETRY(IES).

If the input operation fails, the input sector is bad and is treated as such (see Bad Sectors Errors).

Error logging, as mentioned in the next paragraph, is logging internal to the DISKDUMP program. Entries for errors are made in two data segments, allowing 3000 COMPARISON and 3000 INPUT and OUTPUT errors to be recorded. Additional errors are not logged.

Bad Sectors Errors

1. Bad Sectors On The Input Device

If a sector cannot be read from the input device, even after the number of retries allowed, the corresponding sector on the output device is filled with the error pattern "BADBAD...". An INPUT error is logged. If the bad sector can be read during the comparison phase, a COMPARISON error is also logged.

2. Bad Sectors On The Output Device

If a sector of the output disk cannot be written into, it is skipped and the corresponding sector from the input device is not copied. An OUTPUT error is logged.

If the corresponding sector from the input device was filled with the "BADBAD..." pattern, no COMPARISON error is logged. Otherwise, a COMPARISON error is logged for this sector.

INPUT and OUTPUT Error Messages

When an error is logged as an INPUT or OUTPUT error, the following message is displayed:

```
<mix/DISKDUMP. : INPUT (OUTPUT) ERROR on <pid> AT SECTOR <xxxx> ...
    ... AFFECTS FILE <filename>
```

Fatal Errors

If a communicate error occurs while copying the first 32 sectors, or during the error location phase in the COMPARE, the execution aborts with the following message:

```
< Mix/DISKDUMP > PERMANENT I/O ERROR WHILE IN READ FROM ...
    ... ( WRITE TO ) " < mfid/fid > " - EXECUTION ABORTS.

< MIX/DISKDUMP > UNEXPECTED FCM [fcm] RECEIVED FOR ...
    ... " < mfid/fid > " - EXECUTION ABORTS.
```

Program Warnings

1. If the input disk is a system disk, it is opened via the "SYSMEM LOCK" facility, and if files were left open on this device, the following warning message is issued:

   ```
   < Mix/DISKDUMP > *** WARNING *** <nnn> FILES WERE OPEN ON ...
       ... < pid > WHEN DUMP BEGAN.
   ```

   The program scans the DFHs of the input disk and displays the file names that are in use. The temporary files are referred to as (TEMPORARY <Mix #><file #>). Temporary files and files in use are not dumped.

2. If an unlabelled open is performed on the input (output) disk, and if it is a CMS disk, DISKDUMP compares the label pack-id with the pack-id from the initiating message. If they do not match, the following warning is issued:

< Mix/DISKDUMP > *** WARNING *** INPUT (OUTPUT) DEVICE IS
<label pack-id> ... - NOT <specified pack-id>.
DO YOU WANT TO RESUME PROCESS ? <YES or NO>

The utility waits for an answer given via an ACCEPT.
If the answer is NO, DISKDUMP goes to End Of Job.
If the answer is YES, the process continues and the pack-id is the one specified in the initiating message (<specified pack-id>).

3. If an open or a close (other than unlabelled disk open) fails, the following message is displayed:

< Mix/DISKDUMP > [event] CANNOT OPEN/CLOSE " < mfid/fid > " ...
... (device - reel nb.)

End Of Job Messages

< mix/prog > : < xxxxxx > RECORDS DUMPED (case of DSKMTP)
COPIED (case of DSKDSK)
LOADED (case of MTPDSK).

< mix/prog > : < xx > ERROR(S) DETECTED ON INPUT DEVICE ....
.... < mfid/fid > (device)

< mix/prog > : < yy > ERROR(S) DETECTED ON OUTPUT DEVICE ...
.... < mfid/fid > (device)

Warnings

When the copy phase is complete, and if a disk has been open "unlabelled", DISKDUMP performs an AVR on that disk. Therefore, the activity is stopped during this time.

Copying a disk where files are actually used in output obviously produces comparison errors at COMPARE time.

Approximate Duration of the Process

DSKDSK of a 206 disk pack to a 206 disk pack: 8 min.
A comparison : 11 min.

DSKDSK of a 207 disk pack to a 207 disk pack :30 min.
A comparison : 45 min.

DSKMTP of a 206 disk pack to two 2400 feet tapes : 8 min.
A comparison : 10 min.

DSKMTP of a 207 disk pack to seven 2400 feet tapes : 30 min.
A comparison : 35 min.

LT (Load Line Printer Train)

This utility allows the proper chain of characters to be loaded on a 450/750 LPM printer. After CLEARSTART, the first time a 450/750 LPM printer is switched on-line, the message:

"LT REQUIRED FOR LP<x>"

is displayed, where <x> is the printer mnemonic. The operator has then to enter:

"LT LP<x> <chain type>"

where <x> is the printer mnemonic and <chain type> must be one of the following train names or train numbers:

For 1100/1500 Train Printers:

Train Name	Train Number	Description
EBCDIC18	001	18-character EBCDIC
FORTRAN48.NONSTD	002	48-character FORTRAN.NON STD
B300.B50048	003	48-character B300/B500
EBCDIC48	004	48-character EBCDIC
EBCDIC72	005	72-character EBCDIC
UKB3500.72	006	72-character EBCDIC
UKB6500.72	007	72-character EBCDIC
PORTUGAL.72	008	72-character EBCDIC
SPAIN.72	009	72-character EBCDIC
FINLAND.72	010	72-character EBCDIC
DENMARK.72	011	72-character EBCDIC
BCL72	012	72-character BCL
TURKEY.72	013	72-character EBCDIC
SWEDEN.72	014	72-character EBCDIC
ASCII72	015	72-character ASCII
EBCDIC96	016	96-character EBCDIC
EBCDIC-UPPER.CASE	016	96-character EBCDIC
EBCDIC.UPPER.CASEB	016	96-character EBCDIC
EBCDIC-LOWER.CASE	016	96-character EBCDIC
EBCDIC.LOWER.CASEB	016	96-character EBCDIC
KATAKANA	017	96-character KATAKANA
EBCDIC.A72	018	72-character alphatbetized EBCDIC
EBCDIC.N72	019	72-character numericized EBCDIC
RPG48	020	48-character RPG
OCR.A72	021	72-character OCR-A
OCR.B72	022	72-character OCR-B
FORTRAN48	036	48-character FORTRAN
THAI	052	144-character THAI

For 400/750 LPM Train Printers (PRINTER CONTROL 5 or 6):

Train Name	Train Number	Description
FORTRAN48	130	48-character FORTRAN
FORTRAN48.NONSTD	130	48-character FORTRAN.NON STD
B300/B500.48	131	48-character B300/B500
EBCDIC3.48	132	48-character EBCDIC-3
RPG48	140	48-character RPG
EBCDIC96	144	96-character EBCDIC
KATAKANA	145	96-character KATAKANA
EBCDIC3.16	254	16-character EBCDIC-3
EBCDIC3.64	255	64-character EBCDIC-3

Output Messages

The normal End Of Job message is:

"<train name> = <train number> LOADED ON LP<x>"

Error Messages

If an error is detected, in some cases a message will be displayed to indicate the problem, in other cases the program will terminate with the message:

<error number> ** LT ABORTS **

The following gives the error numbers and their meanings:

Error Number	Message And Description
01	"LP <x> NOT AVAILABLE FOR LT" Printer file could not be opened for a reason other than device not on system, printer in use, or "NO FILE" condition. The reason for the failure is unknown but is not a recognized CMS condition.
02	No buffer is available for LT.
03	Problem reported when loading translate table to printer.
04	Conditional failure when closing line printer.
05	Access failure on SYSTRANSLATE.
06	System type is not B 1000 CMS system.
07, 08	SYSTRANSLATE open failure other than "NO FILE", disk locked or file in use.
09	Conditional error on READ of SYSTRANSLATE.

Teach Option

"LT T" command causes the SPO to display:

1. The syntax diagram of the LT utility.
2. A list of the train names with their train number.

MEM.ANALYZER

A memory dump file can be obtained by executing the appropriate MEM.DUMP cassette in stand-alone mode (for operating instructions, refer to MEMORY.DUMP in Stand-Alone Utilities later in this section). This causes the contents of the memory to be written into a file named DMFIL00.

If this file exists when a memory dump is initiated, the memory dump will use file DMFIL0001, DMFIL0002, or the first free file up to DMFIL0099. If all these files are used, the memory dump will overwrite the contents of DMFIL00.

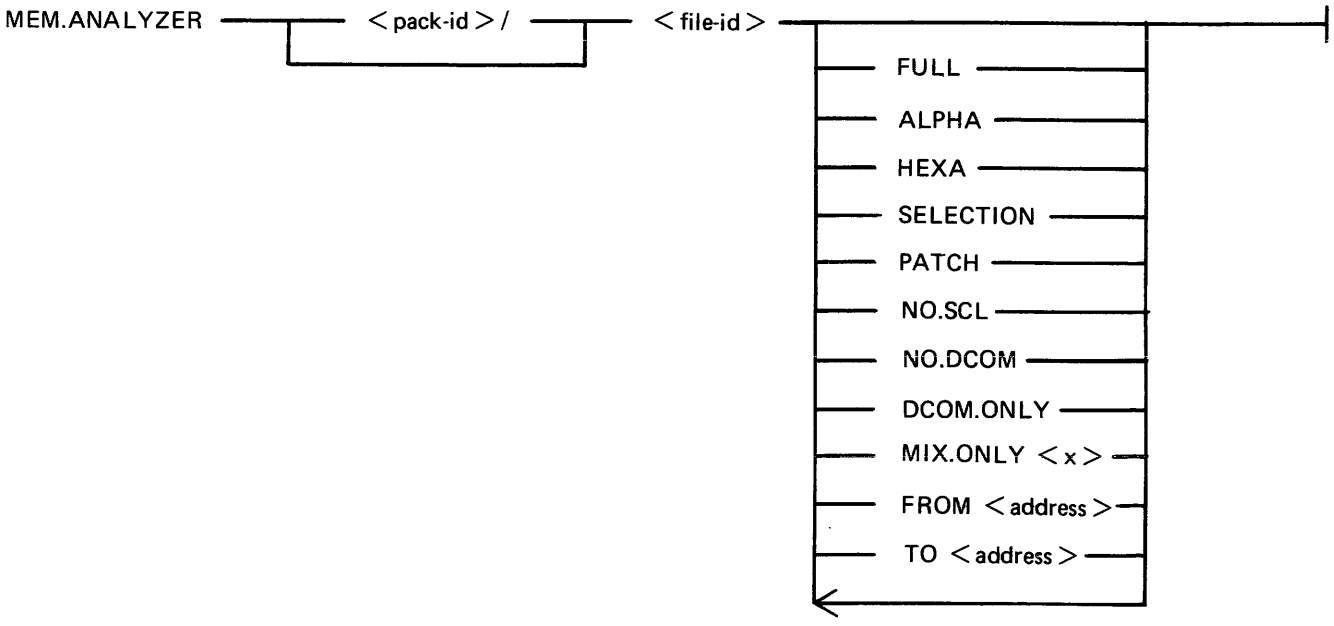
The file DMFIL00 is created on the system disk. If enough space is not available on that disk, or if memory is corrupted so that the location of that disk is destroyed, the program halts to enable the operator to direct the dump to another disk.

The system must then be restarted, with a CLEARSTART, and the MEM.ANALYZER program must be executed to obtain a formatted, analyzed listing of the memory dump file.

IMPORTANT

If a DCP dump is also required following CLEARSTART, (always the case on a B 1990), ONLY the START button must be pressed and NOT the CLEAR button. On a B 1990 use TAPE GO and not MTR GO.

The listing can be obtained by invoking MEM.ANALYZER with the following syntax:



<pack-id>	indicates the name of the device containing the dump file to be analyzed.
<file-id>	indicates the name of the dump file to be analyzed.

The options, entered in any order, have the following effects:

FULL	Print tables and all memory in hexadecimal.
ALPHA	Print memory contents in alphanumeric.
HEXA	Print memory contents in hexadecimal.
PATCH	Printing of the patch analysis is enabled.
NO.SCL	Disable SCL partition decoding.
NO.DCOM	Disable Data Comm module decoding.
DCOM.ONLY	Disable User-memory analysis.
MIX.ONLY <x>	In analysis of User-memory, only the mix specified by <x> will be considered.
FROM <address>	When ALPHA or HEXA is used (default is 0).
TO <address>	When ALPHA or HEXA is used (default is the maximum memory value). 1. The FROM and TO options do not require the specification of ALPHA or HEXA. The default is HEXA. 2. The FROM...TO.. option cannot be repeated to print different portions of memory in one run.
SELECTION	This option causes a series of ACCEPT/DISPLAY's to be executed on the SPO. These allow the operator to select sections of the dump listing to be printed on a printer. The syntax for these sections follows:

MAT	indicates the Memory Assignment Table portion of the listing.
VM	indicates the Virtual Memory Tables and VM statistics portion of the listing.
PHYSIO	indicates the PHYSICAL.IO information portion, that is, Channel Control Words, Device Control Blocks, Channel Control Blocks, etc.
DCOM	indicates Data Comm information.
SCL	indicates SCL information.
SEGMENT <integer-1> <integer-2>	indicates a data segment to be printed. <integer-1> is the data segment number in the Data Segment Table. <integer-2> is the Mix number of the associated program.
FIB <integer-1> <integer-2>	indicates the FIB to be printed. <integer-1> is the data segment number of the segment containing the FIB in the Data Segment Table. <integer-2> is the Mix number of the associated program.
MIX <integer>	The entire User Memory Analysis section (that is, ICB, DST, PST, File information) for the Mix indicated by <integer>.
ICB <integer>	indicates the ICB of the Mix indicated by <integer>.

DST <integer>	indicates the Data Segment Table of the Mix indicated by <integer>.
PARTITION <integer>	indicates a hexadecimal dump of the partition of the mix indicated by <integer>.
PART LAYOUT <mix num>	indicates the layout of segments in memory for the specified mix.
PROG SEGMENT <seg num> <mix num>	prints the content (in hex) of the specified program segment of this mix.
END	terminates the selection process.

NOTE

The printer file used in this selection process cannot be diverted to backup. For a selective listing which can be diverted to backup, use the other MEM.ANALYZER options.

NOTE

Each dump submitted with an FTR must use the default option.

This program analyzes the input file specified and a complete listing gives:

- A form to be filled in by the user.
- A form to be filled in by the Burroughs representative.
- The contents of the main registers (some registers must be noted manually – see MEMORYDUMP later in this section).
- The last 12 entries in the Address Stack.
- Scratchpads.
- Hints.
- Mix table.
- Monitor stack analysis.
- Memory assignment table.
- Dual processor table.
- Virtual memory tables.
- Virtual Memory statistics.
- Channel and Channel Control Word tables.
- Channel control blocks.
- Device control blocks.
- File control blocks.
- SCL partition (mix # 0).
- Overlay module analysis (mix #62).
- DATA COMM module analysis, if DCCH was running.
- Shared files Data segment.
- A few other system tables.
- For each user mix, its ICB, PST, DST, FIBs and FPBs.

The copy of the interpreters in memory is compared with their initial version on the system disk and the address of the first discrepancy, if applicable, is displayed.

The Memory Dump file is always retained after MEM.ANALYZER has gone to End of Job. It can be removed with the RM command by the operator.

The default option does not analyze everything in the memory dump file. It is therefore recommended that the file be copied to a backup disk, or dumped to a tape, in case the information not in the analyzed listing is requested later.

NPC1.B1000

Introduction

The NDL Post Compilers are MPL programs which take an NDL S-code file as input and generate one or more DCP microcode files. These files are loaded in the appropriate DCP at Data Comm Sub-System (DCSS) initialization time.

Since three Data Comm processors exist (DCP-1, DCP-3 and DCP-4), the correct NDL Post Compiler must be used to generate the appropriate DCP code files.

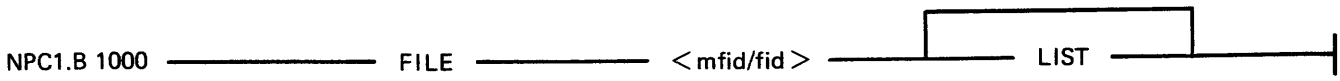
The NPC1.B1000 generates code files to be loaded on DCP-1.

Any mismatch is recognized by DCCH, which issues an error message and terminates the loading of that DCP.

Use of the NPC1.B1000

NPC1.B1000 should be run after each successful NDL compilation.

Syntax



It is strongly recommended that the CODEFILE statement is used in NDL to give a distinctive name to the NDL S-code file.

The LIST option produces a listing of the microcode file(s) produced by NPC1.B1000.

The names of the microcode files produced are taken from the TERMINAL statements in the DCP section of the NDL source.

As each microcode file is created, the message:

<mix>/NPC1.B1000 MICRO CODE FILE <file-id> CREATED

is displayed.

If a file exists with the same name as the DCP file, a "DUPLICATE FILE" message is issued by the MCP. The RM function must be used to remove the old file. (DCP files are "SYSTEM" files so their removal must be "OK'ed").

If the file is not an NDL S-code file, then NPC1.B1000 displays:

NO NDL OBJECT FILE
ENTER NEW NDL OBJECT FILE NAME (NULL ACCEPT TO USE SAME FILE)

NPC34.B1000

Introduction

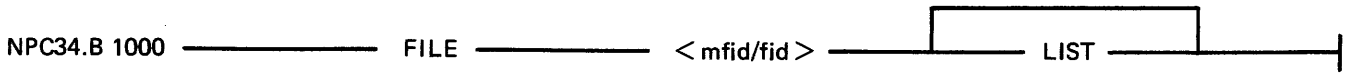
See Introduction for NPC1.B1000.

The NPC34.B1000 generates code files to be loaded on DCP-3 or DCP-4.

Use of the NPC34.B1000

The NPC34.B1000 should be run after each successful NDL compilation.

Syntax



The LIST option creates a printer backup file named PB00000, containing a listing of the microcode file(s) produced by NPC34.B1000. This listing can be obtained by executing "PB 0".

See NPC1.B1000 for the names of the microcode files and for the utility messages.

PATCH.MAKER

This utility is released in order to implement required patches to released firmware. This utility either reads a file of patches from cassette or generates such a file from SPO input.

It is recommended that an adequate backup system is employed.

The easiest way to patch the MCP is to use two System disks. However, a single System disk can be used to perform the patching of the BIL interpreter (BILINTERP), the COBOL interpreter (COBOLINT), the DCP-3 or the DCP-4 NDL Post Compiler (NPC34.B1000) as well as of the MCP when two System disks are not available (see Operating Instructions later in this section).

Remember that ALL disks containing system software should be updated with the patched software.

Levels of patches are successive. If a user requires patch level 3, for example, then both previously released patches must have been implemented. The utility employs a number of checksum routines which prevent patching of the wrong file and implementation of unauthorized patches.

The utility first asks the user, with displays and accepts, for the identity of the firmware file to be patched in this run (MCP, BILINTERP, COBOLINT or NPC34.B1000) and of the new file to be created.

The user is then asked to specify if the patches are to be entered via the SPO or via a cassette. If the response specifies cassette, a search for a tape file named "PATCHES" is initiated. When this file is found, the patching is carried out. If the response specifies SPO, the patches are to be entered via the keyboard from the hard-copy supplied. The characters entered must be exactly as supplied. The series of characters can be divided into several series of any length up to 254 characters. The operation is terminated by a null AX.

In order to minimize the risks involved in transmitting patches by TELEX, and possible errors in operator action, an initial sumcheck has been inserted in the patch string itself. Each 12 bytes of patch information are sumchecked. If a mismatch is detected, the wrong 12-byte group will be displayed and these 12 bytes must be re-entered via the SPO.

A null "AX" terminates input and allows the patching to begin.

If any message is displayed, one of the checks has failed. The input must be examined and re-entered when the error is found.

Error Messages Displayed

<string of 24 bits> AT OFFSET <nn> IS INVALID: RESUBMIT ON SPO

Initial sumcheck has detected an error within the string of digits entered via the SPO or cassette.

PATCH LEVEL DISCREPANCY

The specified file does not have the patch level expected. All previous patches must have been entered.

INITIAL SUMCHECK DISCREPANCY

The sumcheck of the original file to be patched does not match the value contained in the patch entry. All previous patches must have been entered.

FINAL SUMCHECK DISCREPANCY

The sumcheck of the patched file does not match the value contained in the patch entry. Try again on another disk drive. If the error persists, the patch may be in error. Contact your local Burroughs representative.

ADDRESS ERROR

The address given for a micro is out of range for the file specified. Try again on another disk drive. If the error persists, the patch may be in error. Contact your local Burroughs representative.

OLD/NEW MICRO DISCREPANCY

The old micro contained in the patch entry does not match the old micro in the file to be patched. All previous patches must have been entered.

Example

Assume that a pack with all the released software is available. It is called BKP3.04.

The system is CLEARSTARTed with BKP3.04 as system disk. The patch consists of a hard-copy (flash with "333355554444", for example).

When CLEARSTART has finished:

```
    PATCH.MAKER
02/PATCH.MAKER BOJ PR=A TIME: hh.mm.ss
02/PATCH.MAKER [DISP] ENTER NAME OF FILE TO BE PATCHED
02/PATCH.MAKER [DISP] [<PACK-ID>/] <FILE-ID>
02/PATCH.MAKER [ACPT]
    AX 02 BKP3.04/MCP
02/PATCH.MAKER [DISP] ENTER NAME OF NEW FILE
02/PATCH.MAKER [DISP] [<PACK-ID>/] <FILE-ID>
02/PATCH.MAKER [ACPT]
    AX 02 BKP3.04/MCP001
02/PATCH.MAKER [DISP] ENTER INPUT DEVICE: SPO OR CASSETTE
02/PATCH.MAKER [ACPT]
    AX 02 SPO
02/PATCH.MAKER [ACPT]
    AX 02 33335555
02/PATCH.MAKER [ACPT]
    AX 02 4444
02/PATCH.MAKER [ACPT]
    AX 02
02/PATCH.MAKER EOJ
```

When this has finished, CLEARSTART the system with BKP3.04 as system disk, with 000001 loaded in X register between TAPE mode and RUN mode (see CLEARSTART later in this section).

Operating Instructions for Patching the MCP File

The easiest and safest way to proceed is to use two system packs. However, it is possible to work with one system pack only.

IMPORTANT

The MCP file used must not have had a DCP dump stored in it by the SDA utility. To be sure of this a "virgin" copy of the MCP file should be kept and used as the input MCP file to PATCH MAKER.

Two System Packs Are Available (SYS1 and SYS2)

1. CLEARSTART with SYS1 as System disk
   SYS2 as User disk.
2. Run PATCH.MAKER with the file MCP on the System disk as input file
   a file MCP on the user disk as output file.
   Thus, the "name of the file to be patched" is : SYS1/MCP, and
   the "name of the new file" is : SYS2/MCP.
3. CLEARSTART with SYS2 as System disk
   SYS1 as User disk.
4. Copy MCP to SYS1/MCP.

One System Pack Only Is Available

1. Copy the MCP file to MCPxxx.
2. Run PATCH.MAKER with MCPxxx as input file
   MCPyyy as output file.
   (xxx and yyy being three-digit integers)
3. There are now two ways of using the new MCP file MCPyyy:
   A. CLEARSTART directly with MCPyyy.
   B. Copy MCPyyy to MCP and CLEARSTART.
   It is strongly advisable to use procedure A, and to avoid procedure B when possible.
   A. CLEARSTART with 000yyy loaded in the X register between TAPE mode and RUN mode (see CLEARSTART later in this section).
   B. 1. Copy MCPyyy to MCP
      2. Ensure that two conditions are satisfied while copying MCPyyy to MCP:
         1) A NULL MIX (even SYS-SUPERUTL must be DS'ed).
         2) NO DEVICE STATUS CHANGE (No device hardware interrupt).
      3. Set the Interrupt switch (Maintenance panel interrupt toggle).
      4. CLEARSTART.
   If any problem arises during these operations, it is always possible to re-run with the old MCP file by CLEARSTARTing with the file named MCPxxx.
   A COLDSTART selecting only the file MCP on the release tape is another means of loading a valid MCP when a patching operation has failed.

Example

To apply patch 45 to the MCP file, having one system disk only:

1. Copy MCP to MCP045.
2. Run PATCH.MAKER with
   "name of the file to be patched": MCP045
   "name of the new file": MCP046
3. Then, either
   1) Perform a CLEARSTART with 000046 loaded in X register between TAPE mode and RUN mode.
   or
   1) Copy MCP046 to MCP, after having DS'ed all programs in the MIX.
   2) Interrupt and CLEARSTART.

Operating Instructions For Patching the Interpreters or NPC34.B1000

Ensure that the file to be patched is not in use:

– No COBOL or RPG program running for patching COBOLINT. – No MPL program running for patching BILINTERP (all utilities are written in MPL).

Example:

To patch the BILINTERP:

1. Copy BILINTERP to BILINTERP1.
2. Run PATCHMAKER with BILINTERP1 as input file,
   BILINTERP2 as output file.
3. Copy BILINTERP2 to BILINTERP.

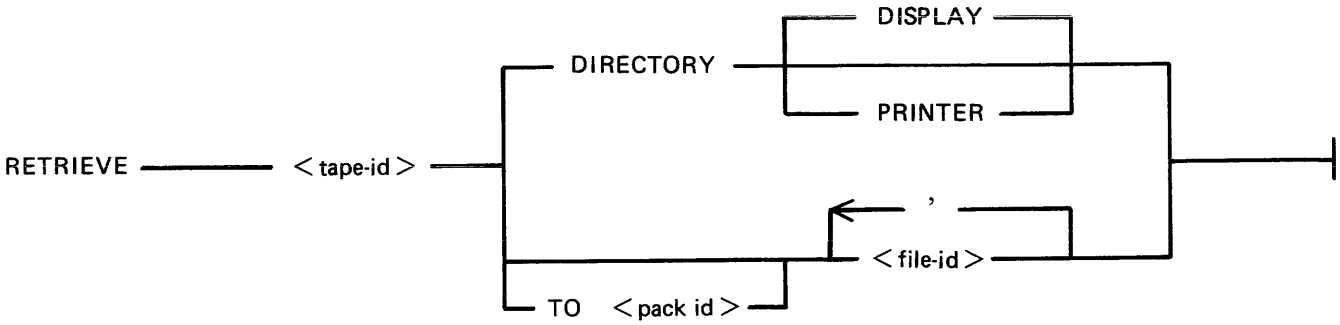
RETRIEVE

This utility provides the user with a means of copying to disk one or more single files from a tape created by the DISKDUMP program.

Syntax

The following syntax diagram is displayed by entering:

"RETRIEVE"



DIRECTORY	Gives to the user the list of the files which are present on the tape, directed to the device specified by one of the following options: DISPLAY causes a single list to be displayed on the screen. PRINTER provides the user with detailed information about each file residing on the tape, that is, the pack-id (of the originating disk), the file-name, the actual and the maximum file size, the creation date and the last access date, the file type.
TO <Pack-id>	Gives the identification of the pack to which the files are to be loaded. The default is the system disk.
<file-id>	Up to 20 files can be loaded in a single run of RETRIEVE utility. The file names are separated by a blank, or a comma or a semi-colon surrounded by blanks.

Messages

Most of the messages that are displayed are self explanatory. The following ones need some explanation.

If a file to be loaded is already on disk, the messages:

FILE <file-name> IS ALREADY PRESENT ON DISK

PLEASE REMOVE IT AND ANSWER TO THE ACCEPT

are displayed on the screen.

The utility resumes as soon as an answer has been given to the Accept, whatever this answer might be (a blank or any word).

SDA (Only Relevant To B 1990 Systems)

This utility provides the user with the means of transferring SPO DCP dump information from a reserved area of the MCP to a file which is accessible by the DCP ANALYZER utility.

If the SPO DUMP option in SYSCONFIG is SET, the SPO DCP memory is stored in the MCP file on disk at CLEARSTART. This avoids the problems of no disk space or directory full.

Syntax

The initiating message for the program is "SDA".

The program creates a dump file which has an identifier of the following format:

DMFILjjjjjnn

where jjjjj = julian date and nn = dump number.

Completion Message

After the dump has been created successfully, the following message is displayed:

DMFILjjjjjnn CREATED

The program now zips DCP.ANALYZER to provide an analyzed listing of the SPO DCP dump.

STAND-ALONE UTILITIES

The stand-alone utilities perform specific functions which are not under MCP control. The following are the stand-alone utilities available with this release:

Name	Function
COLD.START	Loads system software to disk from a CMS library magnetic tape.
CLEAR.START	Determines the system configuration, builds tables and places the system under MCP control.
PACK.INIT	Initializes a disk pack in CMS format.
MEMORY.DUMP	Creates a disk file (DMFIL00) containing a complete image of the system memory.

All these stand-alone utilities are initiated with cassettes.

Creation Of Cassettes

The COPY utility has the facility to enable B 1000 series stand-alone cassettes to be generated under CMS MCP control. Creating cassettes requires an I/O cassette drive on the system. The inbuilt cassette drive cannot be used, because it is used only for loading the stand-alone utilities.

Syntax

COPY <sau-file-name> TO XXX MTP NO.LABEL

where <sau-file-name> is one of:

CLEAR.START	PACK.INIT
COLD.START	MEMORY.DUMP

Operating Instructions

1. Place blank cassette in CTA.
2. Enter "PG CTA".
3. System displays: "CTA PURGED".
4. Enter COPY message.
5. System displays:
   "<mix>/COPY WAITING SPURIOUS/XXX AT NO FILE"
6. Enter "AD <mix> CTA".
7. Cassette is created and the following message displayed:
   "<sau-file-name> TO XXX COPIED"
8. To write on the other end of the cassette, remove it from the drive, wind on by hand until tape is exposed in the opening and insert it the other way round in the cassette drive. Go back to step 2.

Initiation of the Stand-Alone Utilities

For All Systems Except B 1825/30, B 1990

1. Place the cassette in the cassette reader on the control panel. Rewind if necessary so that the BOT light is illuminated.
2. Set the register select switch so that it points to the L register. Set the cassette switch to SYSTEM, if applicable. Ensure that the SINGLE MIC/CONT switch is set to CONT and MICRO SELECT to NORMAL (B 1860), or SINGLE MICRO to NORMAL (B 1900).

3. Set the MODE switch to the MTR position. Press the CLEAR and then the START buttons. If the cassette does not start, press HALT and CLEAR simultaneously and then START. After reading the cassette, the system will halt with @AAAAAA@ in the L register and the release level (for example, 03.04.00) in the T register.
4. Set the MODE switch to the NORMAL position and press the START button. This will load the program and begin execution.

B 1825/30 System

1. Place the cassette in the cassette reader. If the cassette is not at BOT, it automatically rewinds.
2. Enter "ST".
3. Press the CLEAR and START buttons.
4. Enter "CR23" (this is the L register). "0101 0101 0101 0101 0101 0101 AAAAAA" is displayed.
5. Enter "SR" and press the START button. The program will load and begin execution.

B 1990 System

1. Place the cassette in the cassette reader. If the cassette is not at BOT, it automatically rewinds.
2. Ensure the system is halted, one of the maintenance screens is displayed and the terminal "FORMS" light is on.
3. Enter "MTR GO" to begin the CLEAR and START functions or "TAPE GO" to begin the START function.

All Systems

NOTE

The CLEAR function causes the contents of the DCP memories to be reset. If the DCP memory dumps are required following a CLEARSTART, do not action the CLEAR function.

The following halts are common to all stand-alone utilities:

Contents of L	Description
@000004@	Cassette parity error.
@111111@	No channel found with SPO attached.
@222222@	Invalid response after a test service request.
@333333@	Invalid status.
@444444@	Memory parity error.
@555555@	Invalid result descriptor.
@666666@	SPO not ready.

Cassette Error

If, during the loading of a stand alone cassette, the system halts with "CASSETTE ERROR" displayed corresponding to the register PERP = 1, it indicates incompatibility between the cassette and the drive. This may be due to:

1. Bad cassette
   - try other end of cassette
   - try another cassette
   - also see following note.
2. Cassette drive heads misaligned – call Burroughs Customer Service Engineer.
3. Bad cassette drive – call Burroughs Customer Service Engineer

NOTE

Although the problem is primarily due to incompatibility developing between cassette and drive, switching the machine off then on again may resolve the problem temporarily.

COLD.START

COLD.START loads system software from a CMS library tape (a single reel tape) to disk (either disk pack or disk cartridge).

It can be either 800 BPI NRZ or 1600 BPI PE, and any tape unit may be used.

NOTE

If a file on disk is found with the same name as one on the tape, the tape file is loaded and the disk file is removed. However, it is advisable to initialize the disk before using the COLD.START program, or to reinitialize the disk if COLD.START aborts.

COLD.START is only necessary for new installations. Users of the 3.04.00 release need only initialize a user disk and use "LD LOAD FROM <tape-id> TO <pack-id> =" to create a new 3.05 system disk.

Operation

Before loading the cassette, ensure that the output disk drive and the input tape drive are both ready. This applies particularly to the B 1990 where the system will halt if the device is not ready, resulting in the MTR screen being displayed. If the device is then made ready and GO entered to restart the system, the utility output messages will appear on page 1. Ensuring that devices are ready before starting avoids the spread of messages over page 1 and page 2.

"**** B1800/B1900 CMS COLDSTART **** MARK 3.05.01" is displayed and the cartridge/pack configuration is displayed.

Disk Pack and Cartridge Units

CHANNEL x Dyy Dyy ...

The value x of the channel is given in hexadecimal (2, B,...). The different values of yy correspond to the identification of all the units connected to disk controls (DKA, DPB,...)

Parameters have to be entered by the operator, in answer to the following questions:

QUESTION: CMS DISK DRIVE -<DPX> OR <DKX>

RESPONSE: Enter disk drive on which the system software is to be loaded, for example "DPA". Any bad specification causes the program to ask for the same information again.

The magnetic tape configuration is then displayed.

Magnetic Tape Units

CHANNEL x MTy MTy ...

The value x of the channel is given in hexadecimal (2, B,...). The different values of y correspond to the identification of all the devices connected to magnetic tape controls (A, B,...).

QUESTION: CMS MAG TAPE UNIT -<MTX>

RESPONSE: Enter the mnemonic for the tape unit on which the CMS library tape is loaded. Any bad specification causes the program to ask for the same information again.

The tape label is read and the following messages are displayed:

"CREATION DATE OF TAPE: MM/DD/YY"

QUESTION : DISPLAY LOADED FILES ? <YES OR NO>.

RESPONSE: Enter "YES" if a list of all files loaded is required, "NO" if no list is required.

The following information is then displayed:

"RELEASE LEVEL : N"

where:
N = 3 if the tape was created with a 3.02 version (or any previous version) of LD.
N = 4 if the tape was created with a 3.03 or later version of LD.

This information is obtained from the label of the tape.

QUESTION: "LOAD ALL FILES <YES OR NO>"

A "YES" response directs COLDSTART to load all the files from the tape.

A "NO" response causes a display of the following message:

"ENTER FILE NAME <12 CHAR OR ETX>"

The operator must enter the name of the file or the family of files (a character string appended with an "=") required. This message is displayed again until a blank entry (ETX only) is made or the limit of 20 entries has been reached.

NOTE

Introducing a file name or a family name with no entry in the directory of the tape will display the error message:

"THIS FILE IS NOT ON TAPE"

The operator is asked to enter another file name or family name.

The message:

"LOADED FILES :"

is displayed, followed (if requested) by the names of the files, five per line, loaded to disk. When all the requested files have been loaded, the message:

"END OF COLDSTART"

is displayed.

NOTE

The message "MISSING ETX, TRY AGAIN" is displayed in the case of transmission without an ETX and the user is allowed to re-enter the information.

It is highly recommended to check that the displayed configuration of disk and magnetic tape devices corresponds to the desired configuration, to avoid overwriting files in error.

Error Messages

The following error messages may be printed. The action to take is noted except where the message is self-explanatory.

"NOT LIBRARY TAPE"

Attempt to COLDSTART with an incompatible tape. Check that the tape unit was specified correctly and that the tape is a CMS library tape.

"MAG TAPE NOT READY. CORRECT AND START".
"DISK NOT READY. CORRECT AND START".

Problem with the indicated device. When corrected, on B 1800/B 1900 systems press the START button on the console panel, on B 1990 systems, enter "GO".

"PROBLEM WITH MAG TAPE COLDSTART ABORTS"
"DATA ERROR ON READING TAPE LABEL COLDSTART ABORTS"

Problem encountered while reading the tape. Verify its density and try on another device if possible.

"PROBLEM WITH Dxx COLDSTART ABORTS" (where Dxx is pack or cartridge mnemonic)
"DIRECTORY FULL COLDSTART ABORTS"
"NOT ENOUGH SPACE ON CMS DISK COLDSTART ABORTS"
"NOT A CMS DISK COLDSTART ABORTS"

These errors are fatal to COLDSTART and it will terminate. For the DIRECTORY FULL or NOT ENOUGH SPACE errors, if the minimum files for system operation have been loaded, the system may be CLEARSTARTed and the files on disk reorganized under MCP control. The minimum files for system operation are:

MCP	SYSLANGUAGE
SYSCONFIG	BILINTERP
SYS-SUPERUTL	TL

For the "PROBLEM WITH Dxx" and "NOT A CMS DISK" errors, the disk should be reinitialized (on this system or reformatted under MCP control on another system) before attempting to COLDSTART again.

"CANNOT REMOVE <FILE-NAME> PART OF DUAL PACK FILE"

COLDSTART continues with loading the other files. Only this file is not loaded.

"DISK NOT FOUND"
"MAG TAPE NOT FOUND"

Warning message displayed when a unit outside the controller's range has been specified.

"NOT STANDARD LABEL"

Warning message. COLDSTART will try to resume processing as if the tape had a standard label.

"TAPE NO LONGER READY COLDSTART ABORTS"

"NOT A DISK FILE HEADER COLDSTART ABORTS"
"TAPE MARK NOT FOUND COLDSTART ABORTS"

Internal messages indicating that COLDSTART is lost somewhere within a file. This can be caused by a hardware problem on the magnetic tape device or a corrupted copy of the System tape.

"COLDSTART INTERRUPTED" (flashing)

The interrupt switch has been set during the load process.

CLEAR.START

Function

CLEAR.START determines the exact configuration of the system, builds various tables for the MCP, initializes memory links, executes the AVR of the system disk, loads MCP and finally, passes control to the Operating System.

Selection of the MCP File

At CLEARSTART time, the user has the ability to specify the MCP file he wants to start with. This MCP file must have the name MCPnnn, where nnn is a three-digit integer.

For all systems except B 1990 the procedure is as follows:

1. With the MODE switch set to MTR, press the CLEAR and the START buttons.

2. Load 000xxx in the X register:
   1) Set the register select switch so that it points to the X register.
   2) Set the maintenance panel toggles to 000xxx (for example, to start with MCP123, set the panel toggles so that they represent the binary number: @000000000000000100100011@)
   3) Press the LOAD button.

3. Set the MODE switch to NORMAL and press the START button.

For B 1990 systems the procedure is as follows:

1. Enter "TEXT MCPxxx"
2. Enter "MTR GO" or "TAPE GO"

Channel Selection

The channels to be handled by the MCP can also be selected at CLEARSTART time via the maintenance panel toggles which must be set before RUN mode, using the CSW register on B 1800 systems, referred to as the CNS register on B 1900 systems.

The toggles are represented here from left to right by six one digit values (A, B, C, D, E and F); the first two digits (A and B) must be set to the value @FA@ and the last four digits (C to F) must be set to a value representing, from right to left, the numbers of the channels which have to be taken into consideration.

For example, setting the toggles to @FA0203@ corresponds to the binary setting @111110100000001000000011@. This enables the MCP to handle only channels 0, 1 and 9. All the other channels are disregarded.

On B 1990 systems set the CNS register to the appropriate @FAxxxx@ value.

Channel Identification

The system configuration can be interrogated using the maintenance panel at CLEARSTART time: if the system disk is not ready or off-line, the system halts with L = @C00009@. At this point,

1. Load @000000@ in the L register.
2. Press START (enter GO on B 1990). The L register contains information about channel 0.

3. Information about each subsequent channel is obtained from the L register by pressing START again.

Format of the L Register

LA = @0@
LB = channel number
LC, LD = CMS device kind
LE, LF = Hardware identification

Hardware ID	CMS ID	Device Type
@01@	@CF@	DSC-2
@1E@	@CF@	DISK PACK
@1A@	@CB@	DCC-1
@1C@	@CB@	DCC-2
@08@	@73@	ICMD
@30@	@A7@	MT(NRZ)
@34@	@97@	MT(PE)
@3C@	@AB@	CASSETTE
@09@	@A7@	MTC-6
@2C@	@33@	TTY SPO
@2E@	@33@	CRT SPO
@10@	@0A@	PC-2-3-4
@3E@	@0A@	PC-5-6
@03@	@0A@	PC-7
@0B@	@0A@	GEM-PC
@2A@	@15@	READER-80
@26@	@19@	READER-96
@04@	@16@	PUNCH-80
@06@	@1A@	PUNCH-96
@02@	@17@	MFCU-80
@0A@	@1B@	MFCU-96
@5E@	@51@	DCP-1
@41@	@53@	DCP-3
@42@	@54@	DCP-4

DISK INITIALIZER (Pack.init)

General Information

All disks must be initialized in CMS format before they can be used under MCP control.

The initializer performs several passes on the disk to initialize and verify it. The number and type of passes depends on the disk type.

If a sector fails, it is relocated or removed if relocation is not possible. That is, an available table entry is made which indicates that the corresponding allocation unit is not available. At most, five sectors per track can be relocated.

Known marginal sector numbers can be entered at the start of the initialize. These sectors are then relocated or removed during the initialize. (See "Maximum Number of Removed Sectors" later in this section.) The disk label and disk directory are built after initialization.

Operating Instructions

Cassette Loading

On cassettes the utility does not stop with L = AAAAAA as other cassette utilities do, but continues loading until approximately half the cassette has been spooled.

After the cassette has been loaded, a title is displayed:

CMS DISK PACK INITIALIZER MARK 3.05.01 (<compile date>)

If the terminal is configured to display "ETX", the following message is displayed:

PLEASE PUSH SPECIFY IF YOU SEE ETX DISPLAYED ON THIS SCREEN

This is needed to allow the initializer to format its output correctly. If the initializer is running on a B 1990 and the terminal has ETX displayed, all input should be terminated with an ETX before pressing XMT. Alternatively, the terminal firmware may be reprogrammed so the ETX is not displayed.

If a card controller is present, the following message appears:

IS CARD INPUT DESIRED (YES or NO)

If the response is NO, the initializer operates in an interactive mode (see "SPO Input" later in this section).

If the response is YES, the input parameters are read from cards, one card for each disk.

Card Format

Each input is in free format, separated by a space, with the parameters in this order:

Unit mnemonic	A through H
Serial number	6 digits
Packid	Up to 7 characters
Date	MM/DD/YY

Maximum number of files	Up to 2805
Owner's name	Up to 14 characters.
Marginal sector numbers	As many as required – decimal numbers separated by a space.

The last two fields are optional. All other fields are checked for validity. If an error is discovered, the card is rejected.

NOTE

The cards must be terminated by a "?END" card.

SPO Input

If SPO input is used, parameters are entered interactively. Each field is checked for validity as it is input. If it is invalid, a suitable message is displayed and the field can be re-entered.

QUESTION: ENTER UNIT MNEMONIC DP <x> or BLANK FOR EOJ
RESPONSE: DPA, DPB and so on or BLANK

QUESTION: ENTER 6 DIGIT SERIAL NUMBER
RESPONSE: 6 digit serial number

QUESTION: ENTER PACKID PR2
RESPONSE: packid, up to 7 characters from the set "A" through "Z", "0" through "9" or "_"

QUESTION: ENTER DATE AS MM/DD/YY
RESPONSE: date as MM/DD/YY

QUESTION: ENTER MAXIMUM NUMBER OF FILES (UP TO 2805) 2805
RESPONSE: A decimal number between 1 and 2805

NOTE

See Available Table Size Example later in this section.

QUESTION: ENTER OWNER'S NAME TBC
RESPONSE: up to 14 characters (this is optional – a single blank is acceptable).

If a printer is available, the operator is asked

QUESTION: DO YOU WANT A PRINTOUT OF REMOVED SECTORS? (LPx or NO)
RESPONSE: LPA, LPB and so on, or NO

Finally:
QUESTION: ENTER MARGINAL SECTORS (DECIMAL SECTOR NUMBERS)
RESPONSE: Decimal sector numbers of known marginal sectors.

Up to three lines of numbers may be input at a time. The order of sector numbers input is not important; the numbers must be separated by at least one space.

If a printout has been requested, a heading is printed.

The disk will now be initialized. While initialization is in progress, pressing the XMT button displays information about the initialize. The initializer processes each cylinder completely before starting the next cylinder.

If a printout of removed sectors is requested, a list of sector numbers removed from a cylinder is printed when that cylinder has been completely processed.

If the printer becomes not ready, the initializer is suspended when it tries to write the printout and remains suspended until the printer is ready. The following message is displayed:

DISPATCH TO PRINTER LPx FAILED – SUSPENDED UNTIL PRINTER READY

When the initialize is complete, a message is displayed on the screen indicating the number of bad sectors. The message is in the form:

DPx : ID = <packid>, SER # <serial number>
<error count> BAD SECTORS (<relocate count> RELOCATED <remove count> REMOVED)
INITIALIZATION COMPLETED

A similar message is printed if the printer option was selected.

The initializer then displays the original prompt. This allows another pack to be initialized or, if a blank is entered, allows the initializer to be sent to End Of Job (EOJ).

Time Needed

The time taken to initialize packs depends on the type of pack and the number of sectors that need to be relocated or removed. Times for various packs with around 20 bad sectors are approximately:

25 minutes for a 205
35 minutes for a 206
150 minutes for a 207
60 minutes for a 225

For a 207 with 2000 bad sectors, the time will be around 190 minutes.

Error Messages

All error messages are self explanatory and most are related to the detection of bad syntax.

If the error is not related to bad syntax, the following values are used when no message can be displayed:

Halt Value (T Register)	Meaning
FC0000	No local SPO found.
FCID <port and channel>	Dispatch to local SPO failed (B 1800/B 1900)
FCIE <port and channel>	SPO error
FC20 <port and channel>	Dispatch to local SPO failed (B1990)

Available Table Size – Example

The size of the Available Table is given by the following algorithm

(number of files requested/32) + 3

The Available Table size is extended so that the whole directory ends on an allocation unit boundary.

Therefore, on a 206 pack where the allocation unit is 8 sectors, 2805 files will give:

2805 sectors for headers

255 sectors for name list
91 sectors for Available Table (2805/32) + 3

3151 sectors for the whole directory.

The Available Table is extended to 92 sectors to have a total directory length of 3152 sectors (the next integer multiple of 8 sectors).

Maximum Number of Removed Sectors

The maximum number of removes varies with the disk type:

223 and 225 : 2030
205 and 206 : 4070
207 no explicit limit : limited by available table

MEMORY.DUMP

The function of the MEMORY.DUMP utility is to get the contents of the memory written into a memory dump file, named DMFIL00 (refer to the MEM.ANALYZER utility for the listing of the dump). If DMFIL00 already exists when a memory dump is initiated, the contents of the memory are written into file DMFIL0001, DMFIL0002 or the first free file up to DMFIL0099. If all these files are used, the message dump will overwrite the contents of DMFIL00.

Operating Instructions

The MEMORY DUMP cassette must be executed in Stand-Alone mode. If the system is still running at the time a memory dump is desired, it must be stopped with the INTERRUPT button. If INTERRUPT does not halt the system, it is in a loop. Therefore, before taking a memory dump, the loop addresses should be recorded. (See "How To Trace A System Loop" later in this section.)

When the system has been halted, the MEMORY DUMP cassette should be loaded as described in the following paragraphs.

NOTE Before taking the dump, check the following:

1. For all dumps, record the values in PERM and PERP as these are not stored in the dump file.
2. If it is known that the dump will be re-directed to a pack other than the system pack, record the values of the following registers as they will be overwritten by the utility: L, T, X, Y, A, CC, CD, PERM, PERP, LR, FA, FB, CP, MAXS, SOA to S15A, TAS and the first ten values on STK.

Loading Instructions

1. If running on a B 1825/30, note the value of the A register and press the CLEAR button.
   On B 1990 systems enter "TAPE GO"
   On other systems, do not press the CLEAR button, only the START button.
2. Mount the appropriate MEMORY DUMP cassette.
3. Switch to TAPE mode (with B 1825/30 enter ST). (Automatic on B 1990 systems.)
4. If running on a dual processor, disable the slave processor. (Automatic on B 1990 systems.)
5. Press START. (Automatic on B 1990 systems.)
6. If the cassette does not start, note the value of all relevant registers. In this case only, press the CLEAR button before attempting to take the dump again. (with B 1825/30 enter SR).
7. When register L = @AAAAAA@, switch to RUN mode and press START. (T register contains the release level).
8. Rewind the cassette.
9. Perform CLEARSTART.
   - do not press CLEAR on B 1800/B 1900, except for B 1825/30.
   - on B 1990 systems enter "TAPE GO"
10. Print the dump file by executing MEM.ANALYZER.

If datacomm was running (always the case on B 1990 systems), dumps of the DCPs should be obtained by setting the DCPDUMP option in SYSCONFIG, then executing the MCS. On B 1990, the SPO DCP is dumped automatically on CLEARSTART, if the SPO DUMP option in SYSCONFIG has been set. A SPO DCP cannot be dumped by setting DCPDUMP in SYSCONFIG.

How To Trace A System Loop

A loop is detectable by the fact that the system does not respond to SPO input or the interrupt switch.

If this occurs, execute the following procedure:

1. Press the HALT button.
2. Display the A register (CR24 on the B 1830).
3. Change the system to single microinstruction mode.
   In this mode, each time the system is started it executes one microinstruction, then halts. Recording the A register gives the address of the current microinstruction. When the values repeat, the loop has been identified.
4. Press the START button and note the A register value. Repeat until the A register displays the same value as first displayed. If that value does not reappear after pressing START about 20 times, the loop is not a small one. In this case, proceed to the next step.
5. Take a system dump.

When submitting an FCF for a system loop problem, please indicate on the Memory Dump listing the contents of the A register for the extent of the loop.

List of Possible Halts (in L Register)

c represents the channel number and d the drive number of the disk to which the dump file is directed.

@Dc000d@ Dump is complete (dump file DMFIL00)

@Dcnn0d@ Dump is complete (dump file DMFIL00nn)

@Fc001d@ Dump file size error.
The size of the dump file can be deduced from the contents of the LR register, according to the following list:

@FFFFF0@	11746 sectors	(2 M-bytes)
@800000@	5872	(1 M-byte)
@700000@	5138	(896 K-bytes)
@600000@	4404	(768 K-bytes)
@500000@	3670	(640 K-bytes)
@400000@	2936	(512 K-bytes)
@300000@	2202	(384 K-bytes)
@200000@	1468	(256 K-bytes)
@100000@	734	(128 K-bytes)
@0F0000@	689	(64 K-bytes)

@Fc002d@ Not enough space in the disk directory. Enter in the L register the parameters of the drive on which the dump file is to be created.

@Fc003d@ Not enough space on the designated disk. Enter in the L register the parameters of the drive on which the dump file is to be created.

@Fc004d@ An error occurred while reading or writing the dump file to disk. Pressing START causes 10 retries to be performed. Enter in the L register the parameters of the drive on which the dump file is to be created.

If any one of the above halts occurs, the memory dump procedure can be restarted by entering:
LB the channel number
LF the drive number
of the pack on which it is desired to create the dump file.

@FF0000@	A cassette parity error occurred.
@AAAAAA@	Switch to RUN mode and press START to continue (automatic on B 1990). Information given at the time of the halt is: – Release level in register T. – Julian compile date in register Y. – Dump file size in register X. – Channel number in register S2A. – Unit number in register S3A.

Dump Failures

If the dump cannot be taken because of "CASSETTE ERROR" refer to "Cassette Error" earlier in this section. If the dump cannot be taken for some other reason and this occurs after more than one attempt, the only way to get a dump is to press CLEAR and START (MTR GO on B 1990) and reload the cassette. However, this means that all status, registers, STACK and scratchpad values are lost and subsequently DCP dump cannot be obtained.

One of the reasons why a dump file cannot be created is if the system pack was seeking when the interrupt occurred.

SYSTEM HALTS DOCUMENTATION

System Halts

Most of these halts are traps for conditions that should not occur. When a halt occurs, a value is displayed in the L register as listed below.

WHEN THE SYSTEM HALTS TAKE A MEMORY DUMP AND SUBMIT AN FCF (SEE MEMORY.DUMP EARLIER IN THIS SECTION). IF DATACOMM WAS RUNNING WHEN HALT OCCURRED, DCP DUMP(S) SHOULD ALSO BE OBTAINED (SEE DCP.ANALYZER EARLIER IN THIS SECTION).

THE EXCEPTIONS TO THESE HALTS ARE DENOTED BY †. THESE ARE CONDITIONS WHICH ARE RECOVERABLE BY USER ACTION. THE ACTION REQUIRED IS INDICATED BY THE NATURE OF THE HALT.

The system cannot be restarted, unless specified.

Where "PRESS START" is advised, the command on the B 1990 is to enter 'GO" and press XMT.

CLEARSTART Halts

"* ALL CLEARSTART HALTS SHOULD BE RECOVERABLE BY THE USER RESOLVING THE EXCEPTION CONDITION"

C00001	SYSMEM not found.
C00002	SYSMEM DFH corruption.
C00003	BILINTERP not found.
C00004	SYS-SUPERUTL not found.
C00005	TL not found.
C00006	SYSCONFIG not found.
C00007	SYSCONFIG File type not = @22@ or SYSCONFIG machine type not in range @51@ – @60@ NOTE @5118@ = B 1800, @5119@ = B 1900, @511A@ = B 1990
C00008	TRACE not found (internal, press START to resume).
C00009	MCP not found.
C0000A	MCP file type not = @1C@
C0000B	System disk not write enable.
C0000C	System disk not ready.
C0000D	Other system disk exception. Press START to start again.
C0000F	CRT SPO exception.
C00010	Soft console exception (B 1830).
C00011	Invalid Data Comm configuration. (both DCP-1 and DCP-3/4 present). The system can be restarted with a correct setting of the channel mask using the maintenance panel toggles.
C00012	Memory parity error.
C00013	System disk AVR failure (Available Table full).
C00014	Cassette parity. (See "Cassette Errors"earlier in this section.)
C00015	Table error. One of the tables, for example Device Control Block or Channel Control Block contains an invalid value. Check system hardware configuration.
C00016	Bad GEM configuration.
C00017	Bad Disk Header (DFH of system file not of expected format).

CxOOFF	An I/O interrupt has been waited from a channel for 25.6 sec. The value of x is the Channel number. Check peripheral status and if good, press START to resume.

MCP Halts

SCL Resident

4A0000	Cannot open SYSLANGUAGE file.
4A0001	Invalid fetch value on read (in X).

VM

4B0002	† Recoverable error, press START to continue.
4B000F	Fatal error, memory corrupted.

Global

4D0000	† Console interrupt, press START.
4D0001	† Memory parity error. T contains: card and chip address decode and syndrome.
4D0002	READ or WRITE out of bounds occurred. See CD register. This is detected in the SCHEDULER.
4D0003	Attempt to schedule a job while its ICB.ADDRESS is zero. This is detected in the SCHEDULER. Mix number is in S10A.
4D0004	† Attempt to schedule a job with an empty STACK.LIST (probably an interpreter error). Replace interpreter. This is detected by the scheduler. Mix number is in S10A and ICB.ADDR in S09A.
4D0005	Attempt to schedule a job with an FIB.REQUEST.ATTENTION bit which does not correspond to an FIB segment. This is detected by the LOGICAL-IO selector. Mix number is in S10A and ICB.ADDR in S09A.
4D0006	Attempt to schedule a job with the FIB.REQUEST.ATTENTION bit set. The FIB segment type is valid (01), but is not in memory. This is detected by the LOGICAL.IO selector. Mix number is in S10A and ICB.ADDR in S09A.
4D0007	Attempt to schedule a job with the FIB.REQUEST.ATTENTION bit set. The FIB type is valid and present in memory, but PHYSIO did not put a fetch value in the FIB.REPLY word. Early return – Wait bit has disappeared. This is detected by the LOGICAL.IO selector. Mix number is in S10A and ICB.ADDR in S09A.
4D0008	ICB.STACK overflow. The error is detected by COMMUNICATE.SWITCH. Mix number is in S10A and ICB.ADDR in S09A.
4D0009	READ or WRITE out of bounds occurred (see CD register).

	As this error is detected by COMMUNICATE.SWITCH, the faulty module can be identified when analyzing the dump. Mix number is in S10A and ICB.ADDR is in S09A.
4D000A	Same as 4D0009.
4D0010	Attempt to communicate with a non-existent resident module. The X register contains the desired module number. The T register contains: module number – displacement. The faulty module can be identified when analyzing the dump. Mix number is in S10A and ICB.ADDR is in S09A.
4D0011	Attempt to communicate with a non-existent overlay module. The X register contains the module number multiplied by 12. The T register contains: module number – displacement. The faulty module can be identified when analyzing the dump. Mix number is in S10A and ICB.ADDR in S09A.
4D0012	BOOTSTRAP does not find an empty slot in the Mix table. SCL is the faulty module.
4D0013	Master tries to execute Slave reserved micro code.

SCL Handler

630002	Previous op-code sent to the SPO was TEST.WAIT.RECEIVE.
630006	Invalid fetch value after a Load Data Segment (in X).
631000	Pack error (System Pack Lockout).

SCL Input

640002	Invalid fetch value after a load Data Segment (in X).

PHYSIO Resident

F0000D	FCB queued twice.
F0000E	FCB not queued while in FCB delink routine.
F0000F	FCB queue link error.
F00010	DCB not queued while in DCB delink routine.
F00011	DCB queue link error.
F00012	System disk not ready.
F00013	Invalid status count while in disk Test.Op.
F00021	Invalid status count detected while in Handle Data Transfer.
F00024	FCB error in NOT-READY handling.
F00025	FCB error in restart on NOT-READY routine.
F00026	Status is INPUT while in OUTPUT.
F00027	Status is OUTPUT while in INPUT.

Physical I/O Overlay Loader

These halts are caused by a READ or a SEARCH problem on the following overlay modules. The system may be restarted after these halts.

FF0001	Disk open old file overlay.
FF0002	Disk open new file overlay.
FF0003	Disk close overlay.

FF0004	Disk areas allocation/change overlay.
FF0005	Disk A.V.R overlay.
FF0006	ICMD AVR/(R/W)/OPEN/CLOSE overlay.
FF0007	Disk Purge overlay.
FF0008	Disk crunch overlay.
FF0009	Single device open overlay.
FF000A	Single device close overlay.
FF000B	Reserved for future use.
FF000D	Disk RL OPEN unlabelled disk overlay.
FF000E	Single device A.V.R overlay.

Data Communication Halts

DC0014	Neither Buffer 1 nor Buffer 2 can be transferred. A Read is requested.

System Dependent Fetch Values

The I/O system failures mentioned in this section should not occur. However, if they do, instead of leading to a System Halt condition, they cause a fetch value to be returned to the running program, as listed below. A "COMMUNICATE ERROR , DS/DP" message is displayed. A Dump of the program supplies the Fetch value in the Communicate Parameter Area.

However, it may be impossible to Dump the program if it remains in Terminating status. In this case, the only way to access the Fetch value is to get a Memory Dump when the minimum of programs remain in the Mix.

When submitting an FCF however, a program dump is insufficient to determine the cause of the problem within the system. Therefore, at the error state press the INTERRUPT button and obtain a memory dump. The exception is when the problem is easily reproduced, in which case the SPO log, program(s), data files and other supporting evidence may be submitted without a memory dump.

Physical I/O

PHYSIO Resident

2010FF	Indexed sequential SEARCH error.
80F014	Invalid verb. Take a dump to find this verb in the FIB.
80F015	Mix waiting I/O bit reset before the end of a communicate with wait variant.
80F016	FIB terminating mask flag not reset after completion of previous I/O.
80F018	Duplicate Read/Write/Close communicate on an active file.
80F01E	Open communicate issued on an already opened file.
80F01F	Disk close communicate issued on a non-opened file.
80F020	Single device Read/Write device error.
80F021	Sectors/Block = 0 while locating FIB Block Number in file.
80F024	Communicate on disk FCB queued to an invalid device.
80F033	No local SPO on system.

Disk Open Old File

80F120	Otheruse error. Take a program dump to find it in the FCB.
80F122	End of file error: MAX.AREA.IN.USE is greater than the number of areas allocated. A program dump will provide the DFH in the buffers.
80F12A	SYSMEM lock error: the device status is locked. Take a Memory dump to get the device status in the DCB.

Disk Open New File

80F201	Directory index error: a SEARCH of an entry of all "8080.." is made but the entry does not contain "8080..".
80F221	FIB size overflow: (number of buffers * buffer size) exceeds (65535 – 26).
80F222	Exceeding file size. See the FIB in a program dump.
80F22A	SYSMEM lock error. Same as 80F12A.

Disk Close

80F313	Invalid user count: take a program dump to get the value in the buffers.
80F31E	FCB structure error: for a CLOSE REMOVE or CLOSE LOCK, the file is no longer in use when its FCB is searched for.
80F321	Buffer size is not multiple of record size.
80F322	EOF error: file size computed from the EOF pointer exceeds the maximum file size.
80F326	Attempt to purge or change name of MCP.
80F32A	SYSMEM lock error.
80F3FF	Close Merge Restart.

Area Allocation/Change

80F40C	Area allocation out of the available table limits.
80F414	Verb error. Same as 80F014.
80F415	Program status error: the program is not in WAIT status while an ALLOCATION or a CHANGE needs a WAIT status.
80F42A	SYSMEM lock error: same as 80F12A.

Disk Purge

80F701	Area has been allocated with size 0.
80F702	Area allocation error. Same as 80F804.
80F707	Available Table size error.
80F71E	FCB structure error. Same as 80F31E.
80F72A	SYSMEM lock error. Same as 80F12A.

Disk Crunch

80F801	Last area size computed from the EOF pointer exceeds the size allocated for the last area.

80F802	Area has not been allocated.
80F803	Area length error.
80F804	Attempt to release an area comprised in the available table.
80F807	Available Table size error.
80F813	User count invalid. Same as 80F313.
80F81E	FCB structure error. Same as 80F31E.
80F820	CLOSEMODE error: CRUNCH flag is reset.
80F821	Record size error. Same as 80F321.
80F822	EOF error. Same as 80F322.
80F82A	SYSMEM lock error. Same as 80F12A.

Single Device Open

20F9FF	Device cleared or LT'ed.
40XX13	LT device in use by MIX # XX.
80F91E	Duplicate Single Device Open.

Single Device Close

80FA20	Invalid MYUSE. See FCB in the program dump listing.

Logical I/O

Note that if bits CD(2) or CD(3) are set, a READ/WRITE has previously occurred.

80FF10	Bit CD(2) or CD(3) is set when entering in the resident.
80FF20	Buffer status is invalid in the Can I Read again routine.
80FF30	Buffer status is invalid in the Sequential Write routine.
80FF40	Buffer status is invalid in the Get routine.
80FF50	Buffer status is invalid in the Put routine.
80FF70	Bit CD(2) or CD(3) is set when entering in the Open Segment.
80FF80	FIB Data Segment is not found in the IFNB Segment.
80FF90	Bit CD(2) or CD(3) is set when entering in the Logical Close Segment.
80FFA0	Attempt to divide by zero due to an invalid value of 0 in an FPB or a KFPB field.
80FFF0	Bit CD(2) or CD(3) is set when leaving logical I/O module.

Data Communication Errors

If one of the following Data Communication errors occurs, the message:

<285> DC ERROR xx ON DC y

is issued, xx being the decimal value of the L register last byte, and y the DCP number.

These values are for information only. They indicate a system problem which should be communicated to the Burroughs Service Department.

xx

01	Memory corruption problem. A message is to be added into the input queue. The message link cleared at dequeue available space time is no longer cleared.
02	Cannot lock the Host to send a Nack message after a B 1000 No Space Condition.
03	The message length given by the Liege header is less than the number of bytes contained in the Host.
04	The message length to empty is still greater than 255 bytes, and the transfer in count supplied by the Host is less than 255 bytes.
05	During a Read sequence (B 1000 in receiving mode) a service request is issued by the Host with the corresponding status: HTC.READ = TRUE HTC.BUSY = FALSE HTC.BUF1 = FALSE HTC.BUF2 = FALSE
06	During a Write sequence (B 1000 in sending mode) a service request is issued by the Host with the corresponding status: HTC.WRITE = TRUE HTC.BUSY = TRUE HTC.BUF1 = TRUE HTC.BUF2 = TRUE Therefore there are no free buffers to continue the Write.
07	When the DCP is in sleeping mode, the only valid sequence is READ. The HTC status is located in S6A.
08	Invalid DCP status.
09	For a new input message, the transfer-in count of the first hardware buffer is less than 6 bytes.
10	The sequence byte is wrong. A buffer may be lost.
11	Invalid HTC status (see S6A).
12	Cannot find MTRINTERP.
13	DCP is in No Space condition and the next interrupt gives an invalid HTC status (see S6A).
14	Channel control word corrupted.
15	Transfer-in count value set to zero during a Nack read phase.
16	Invalid Liege Header during the Nack Read phase.
17	During a Read phase, the transfer-in count is less than 255 bytes while Busy bit is true.
18	During a Read phase, the transfer-in count is less than 255 bytes and two buffers are filled.
19	The HTC CNTL register is null after a test and lock command.
21	Protocol error.
26	Firmware halt (take DCP Dump).
27	Unexpected no. of bytes received by HOST DMA.
28	Unexpected status received while in ROM mode.
29	DCP Memory parity error.
30	Unexpected Jump in ROM

APPENDIX A
GLOSSARY OF TECHNICAL TERMS

ADDRESS
A disk is divided physically into tracks and sectors, both numbered sequentially from zero upwards. These 'numbers' are referred to as 'addresses'. The MCP uses this address scheme to quickly locate data on disk.

ALPHANUMERIC
Consisting only of letters of the alphabet plus the ten numeric digits; that is, not containing any other special characters.

APPLICATION PROGRAM
User program that performs day-to-day functions such as invoicing, printing, inventory reports, etc.

ATTRIBUTE
Characteristic or quality.

BACK-UP
Term used to describe the method of insuring that copies of files exist to standby as alternatives.

BINARY-CODED DECIMAL (BCD)
A method of coding numeric information in 4-bit units representing 0 as bits 0000, 1 as bits 0001, 2 as bits 0010, up to 9 as bits 1001. For example, the number 1607 in BCD would take four 4-bit units (2 bytes), coded as 0001 0110 0000 0111.

BOJ
'Beginning of Job'. The term used to notify the operator that a program has entered the 'mix' and has just started running.

BSMD
Abbreviation for 'Burroughs Super Mini Disk'.

BYTE
One alphanumeric character of data.

CHECKERBOARDED
Term applied to any disk having available spaces of varying sizes scattered about the disk among files. The term can also be applied to memory in a virtual memory system where 'locked' or 'save' areas are scattered through the memory in such a way as to impede getting overlayable memory areas of sufficient size for optimum throughput.

CMS
Computer Management System. A set of interrelated specifications for system software, including level language compilers, object-code formats, operator interface and data communications, which Burroughs has implemented on machines of different hardware characteristics.

COMPILATION DATE
The date on which a programmer's source code was compiled: that is, the creation date of the executable object program.

COMPILERS
Group of system programs that convert instructions written by a programmer, in a language such as COBOL or RPG, into a form which can be run or interpreted by the hardware or system software.

CONFIGURATION
Term used to describe the arrangement of various hardware devices in a particular system.

DATA FILE
A set of information usually on disk, which is used as data to be input.

DEFAULT VALUE
Usually a meaning that a program will assume if not instructed otherwise.

DESTINATION
Disk to which information is being transferred.

DISK DIRECTORY
List, on Track 0, of file names, locations on disk, and sizes. Similar to a table of contents.

DISK FILE
Set of information residing on a disk medium, collectively referred to by its name, 'file-name' and the name of the disk on which it resides ('disk-name').

DISK NAME
Name by which a disk is known to MCP. Every disk medium has a 'label' of information written to it during disk initialization, and the disk name is part of the 'label'.

DUAL-PACK FILE (MULTI-VOLUME FILE)
A file that resides on two separate disks or logically defined disks (for example, DKA, DKB).

EOJ
'End of Job'. The term used to notify the operator that a program has terminated. 'Abnormal' end-of-job occurs when a program is terminated prematurely due to an error condition.

EXECUTION
The running of a program is termed 'program execution'. The operator can execute (or start) a program by entering the name of the program desired (or disk-name/program if program resides on user disk). When a program is 'executed', it enters the 'mix' and is assigned a 'mix number' by the MCP.

FAMILY (GROUP) OF FILES
Two or more disk files having at least the first letter of their names in common. For example, 'PR020', 'PRFILE' and 'PASM1' are members of a family of files that could be referred to as 'P–'.

HARDWARE
Term referring to all equipment on the system. Line printers and disk cabinets are examples.

HEXADECIMAL ('HEX')
A number system based on root 16, in contrast to common 'decimal' system based on root 10. To provide additional symbols, the letters A through F are used, so that counting proceeds thus: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12 ... for decimal numbers 0 through 18 ...

INTERPRETER
A system software item used as an intermediate step in the running of a program. Instead of using a compiler to transform programs directly to machine instructions which can be run by the processor, a compiler may transform the program to an intermediate form (called 'S-code'). The S-code can be 'interpreted' by the interpreter, that is, translated into machine instructions that can be run by the processor.

INTRINSIC
A 'command' used by the operator to direct the activities of the MCP. Intrinsics are actually a part of the MCP and therefore will never by seen on a disk file listing or in the 'mix'. Examples of intrinsics include 'DS' (discontinue the processing of a program) and 'DT' (retrieve or change system date).

KEYFILE
File used by system as an index to a master data file.

LABEL
A small space of disk on tape indicating the medium's contents, name, etc. A disk label may be created during the initialization process, and a tape label is created when the tape is purged.

MAIN MEMORY
Circuit boards inside processor where program code and data in immediate use are held.

MCP ('Master Control Program')
Program which is the central part of the CMS software system. It handles hardware devices, communicates with the operator, and controls processing of programs.

MIX
Term applied to the mixture of programs running in a multi-programming environment. A 'mix-number' is a number which is assigned by the MCP to a program when it enters the 'mix'. A 'null' mix is when no jobs are running. The program's name and mix number can be used by the operator to refer to a particular program in the 'mix'.

MULTIPROGRAMMING
One processor working on more than one program at a time. Processing can be shared on a 'round-robin' basis, and computation can be overlapped with input/output if there is more than one program 'in the mix'.

ON-LINE
Term used for equipment or media currently used as part of the system.

PACK
Synonym for 'disk'.

PERIPHERAL
Hardware device used as input or output. Examples are line printer, disk drive unit, console keyboard.

PURGE
To erase. When disks or tapes are 'purged', their contents are lost.

SECTOR
A disk is divided physically into data storage spaces called sectors, numbered sequentially from zero upwards. Each sector is 180 characters in length.

SOFTWARE
Term referring to programs and files, as distinct from the 'hardware' of the actual machine.

SOURCE DISK
Disk from which information is being transferred.

SOURCE FILE
A disk file containing statements (instructions) written by a programmer in a high-level language such as COBOL or RPG, before it has been transformed into a program which can be run.

STAND-ALONE PROGRAMS
Programs that do not run under control of the MCP. In particular, functions of general use to all B 80 users are held in a disk file called 'SAU' (Stand Alone Utilities). Examples include LS (list disk name and sizes), and RL (relabel a disk). Loading and execution of SAU is done with no need of the MCP. Refer to section 8 for details.

STARFILE
A small disk file optionally used at the start of most CMS-common utilities. The information in the starfile is used to build up the initiating message for the utility, which could also be entered by the operator on the SPO. Starfiles are also called 'macro-files'.

SYSTEM DISK
The disk containing the copy of the MCP that is currently in use. Note that a user disk may also contain MCP code files, but only the disk containing the MCP that is in use since the last warmstart is the system disk. There can be only one system disk at any time during operation. System disks cannot be used as system disks on more than one CMS product (see section 2 for details).

SYSTEM FILE
A disk file which is used by the system software. Special control is placed on these files to minimize the danger of accidental removal from the disk (see RM utility).

SYSTEM SOFTWARE UTILITY
A program of general use to all users, as opposed to an application program which performs a particular user's task using day-to-day tasks, such as invoicing. Examples of utilities include COPY (copy files from one medium to another) and RM (remove files from a disk).

THRASHING
Thrashing is the state where the amount of user work done by the system is very small as compared to that done on its housekeeping.

USER
Any disk available to the system that is not a system disk.

VIRTUAL MEMORY
A software technique, implemented in the MCP, of allowing programs to execute (or several to execute together) when the total program memory requirements exceeds the amount of memory physically available. Some of the executing program's code and data, which is not in immediate use, is stored on disk media and not in main memory. When the code, or data, is required, space is made for it in main memory and the information read back from disk. To make space in memory, it may be necessary for the MCP to re-use some memory which has previously been used by the program and is not required at this moment. Before re-using memory containing data that could have been updated, the MCP writes this segment of memory to the program's 'virtual memory file' on disk. This technique also applies to the code and data of the MCP and other system software.

VOLUME
Synonym for 'disk'.

WRITE INHIBIT
To prevent disk on tape media from being written to by a program. The manner in which this is accomplished depends on the medium (see B 80 or B 800 System Operator manuals for details).

WRITE PERMIT

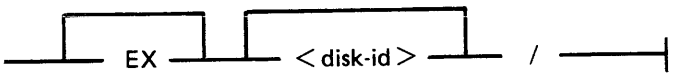
To allow any disk or tape medium to be written to by a program. The manner in which this is accomplished depends on the medium (see B 80 or B 800 System Operator manuals for details).

APPENDIX B
COMPLETE RAILROAD DIAGRAMS

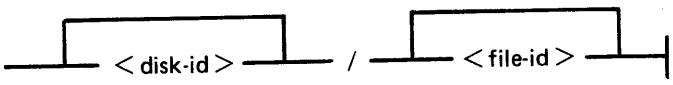
This appendix gives the railroad diagrams for all the CMS-common intrinsics and utilities, including SORT and CO, in alphabetical order. These diagrams give the complete input message formats, for ease of reference.

For details of the meaning of these messages, refer to text.

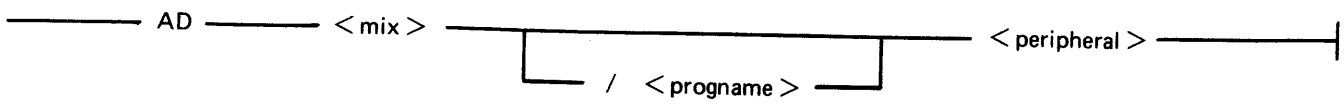
In the following diagrams, the <ex-option> is defined as:



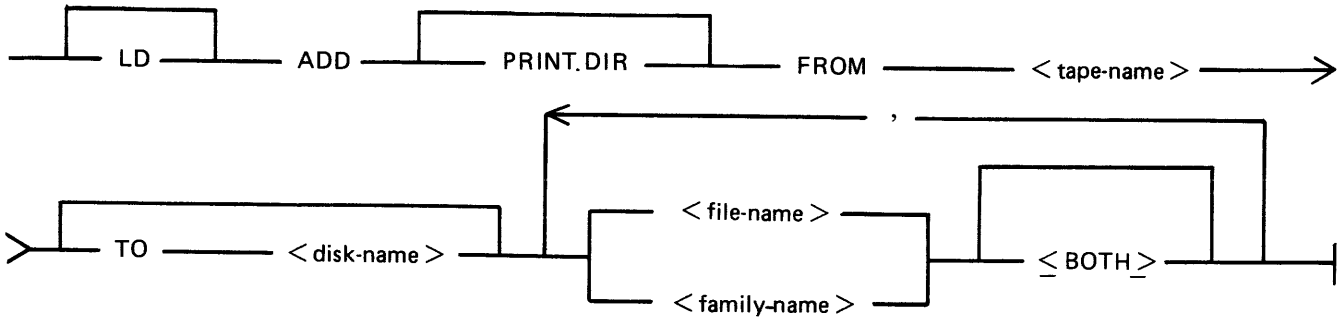
and <file-name> is defined as:



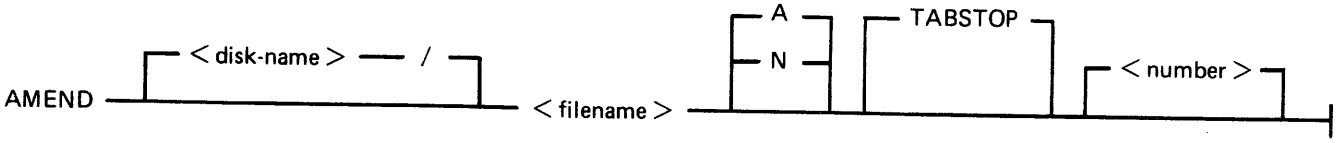
AD Intrinsic



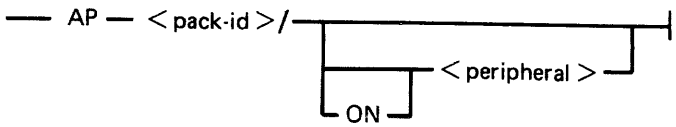
ADD Utility



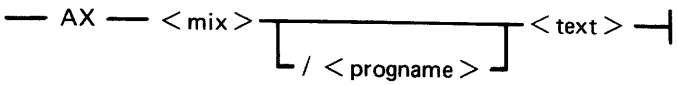
AMEND Utility



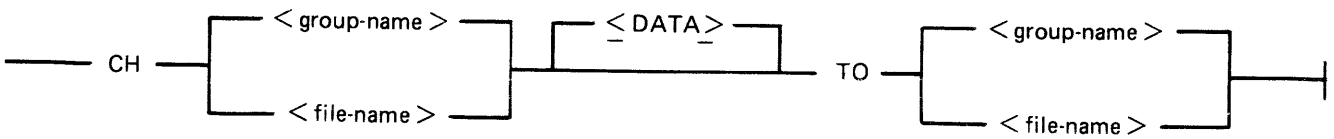
AP Intrinsic



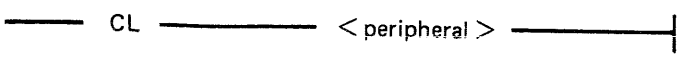
AX Intrinsic



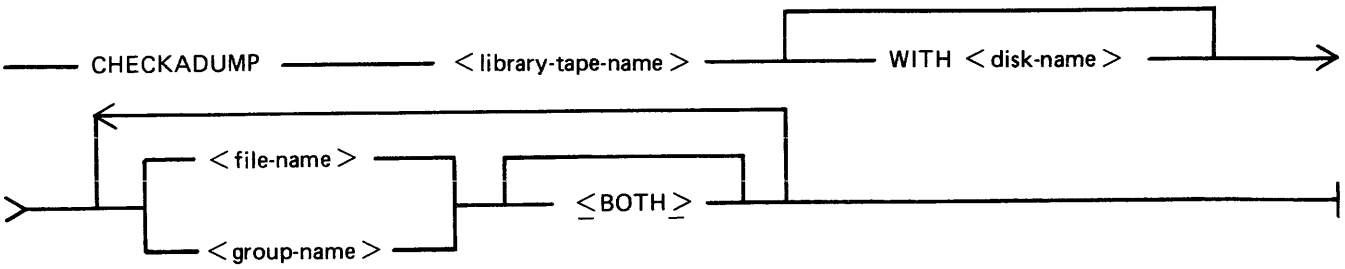
CH Utility



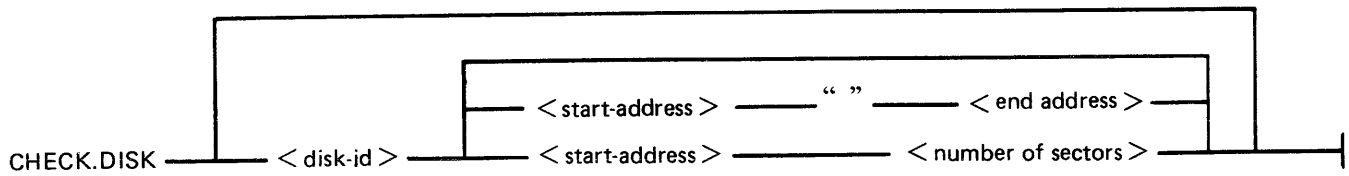
CL Intrinsic



CHECKADUMP Utility

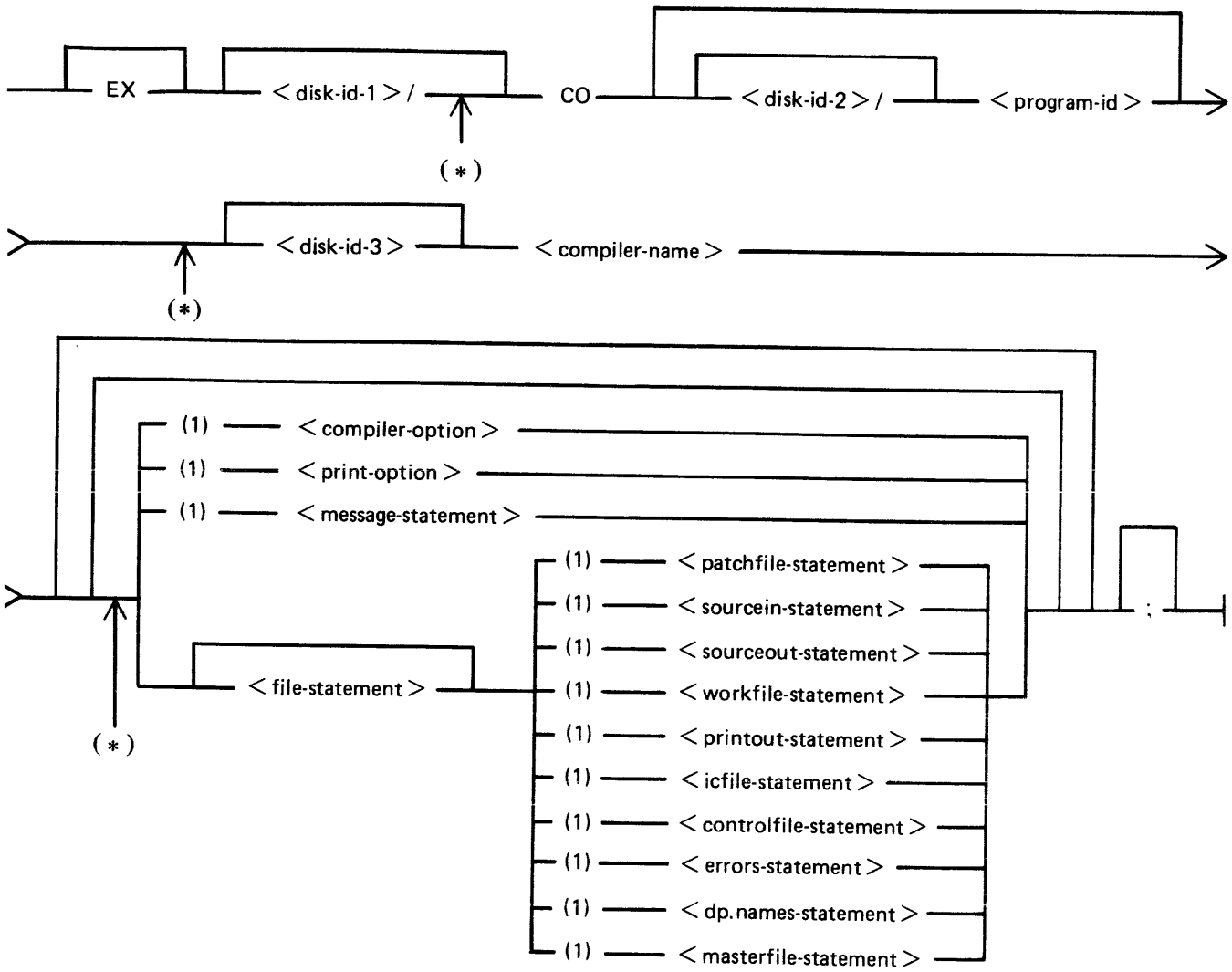


CHECK.DISK Utility

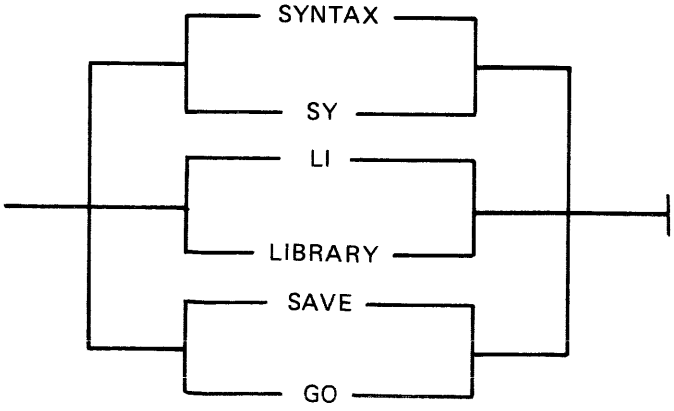


CO Utility

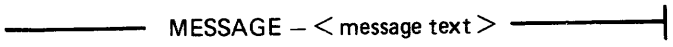
Version 1:

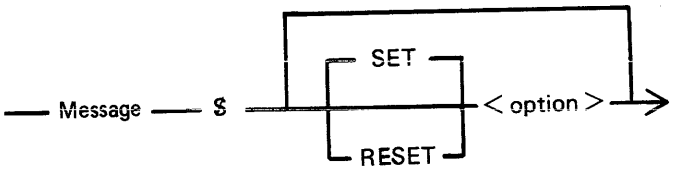


(*) – Indicates allowed position of macro call



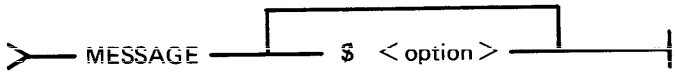
CO Utility (Continued)



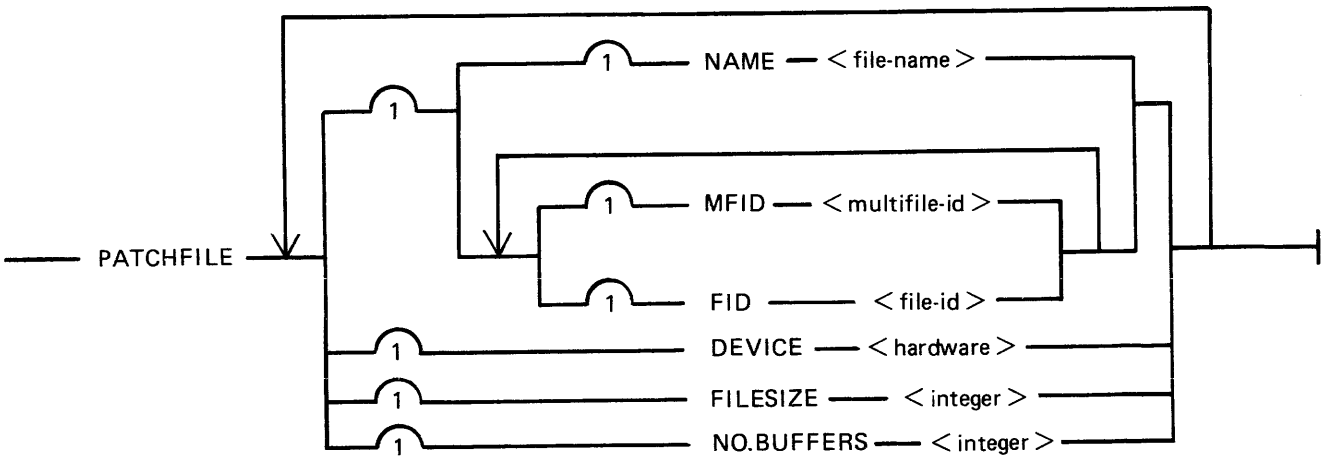


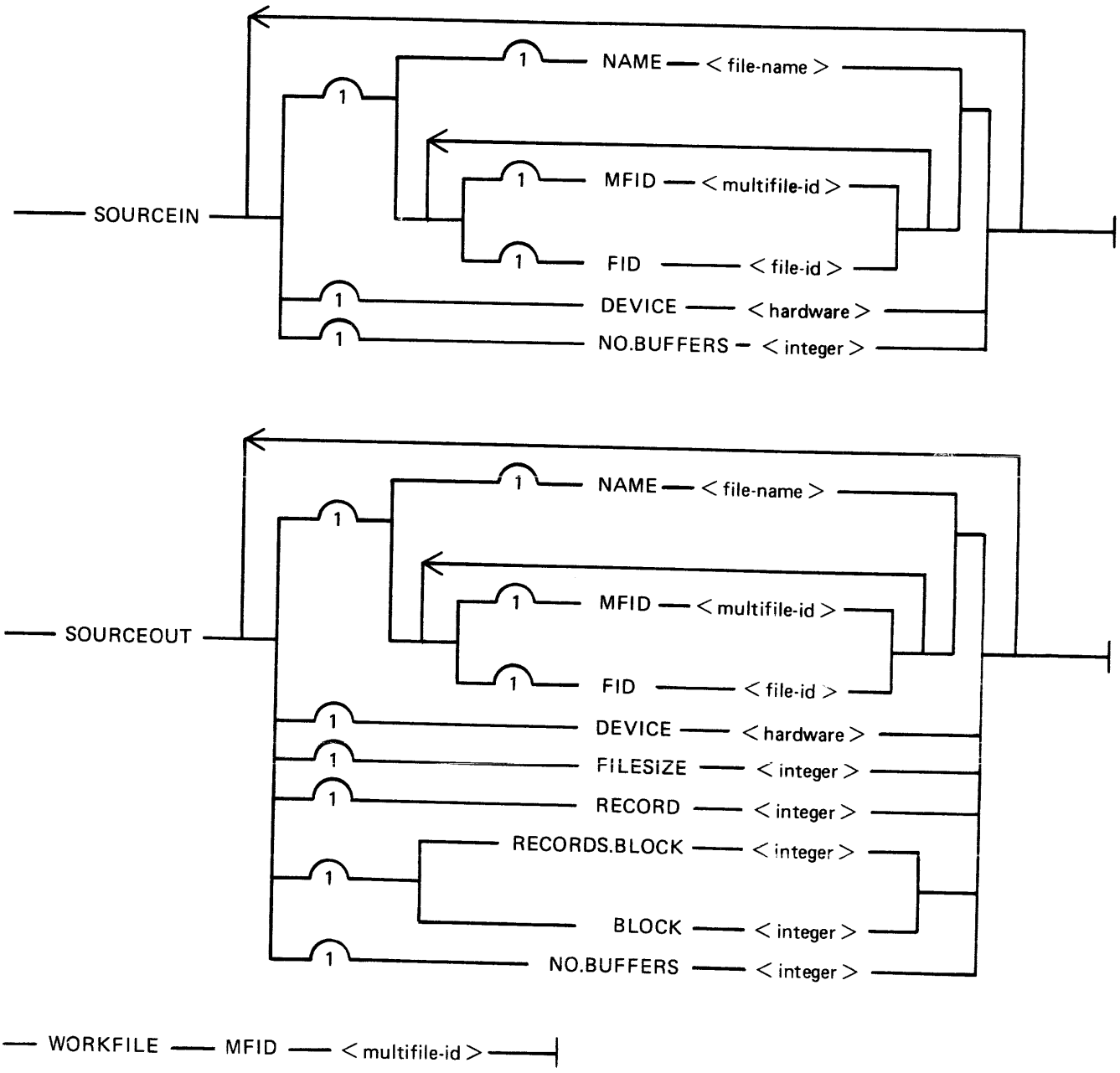
Examples: MESSAGE $ LIST XMAP
sets LIST and XMAP

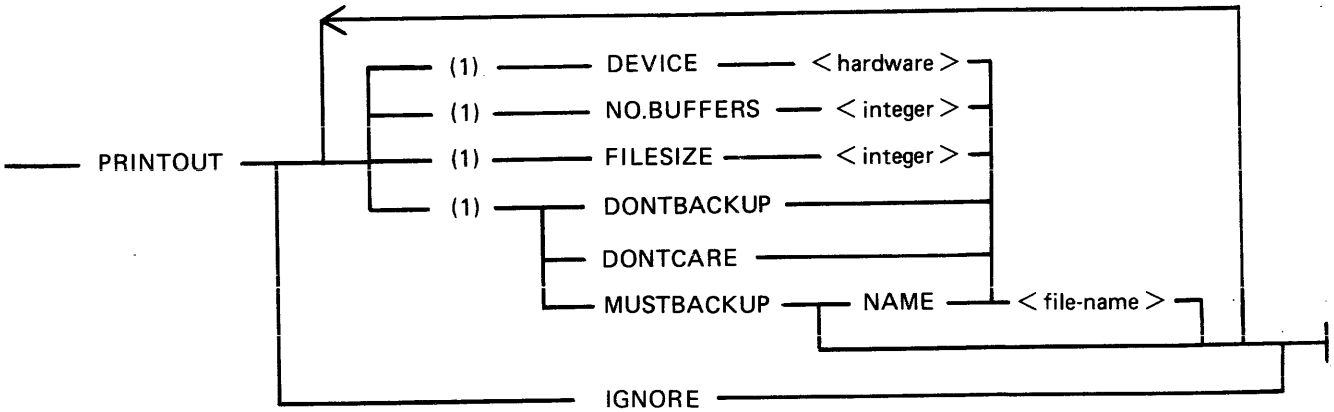
MESSAGE $SET LIST RESET XMAP SET CODE
sets LIST and CODE, resets XMAP.

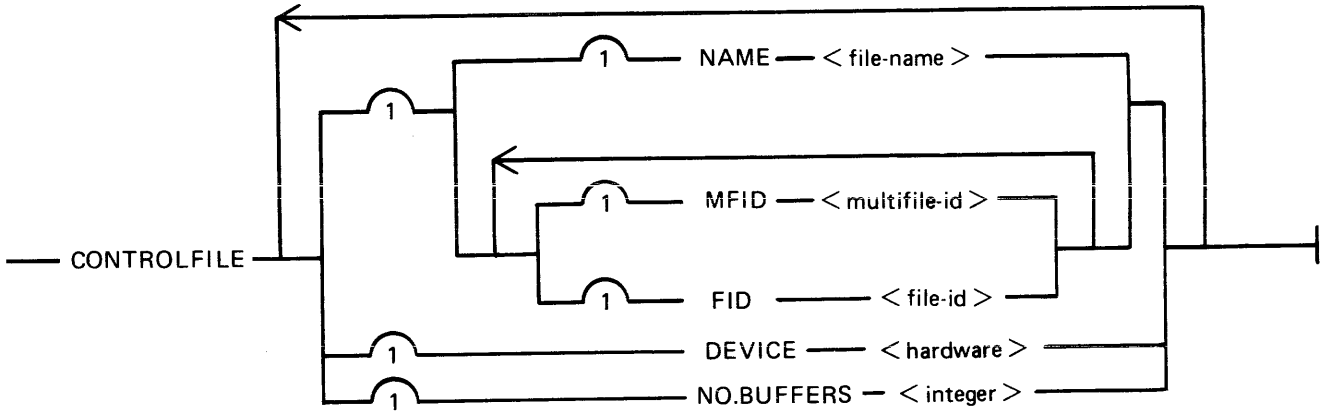


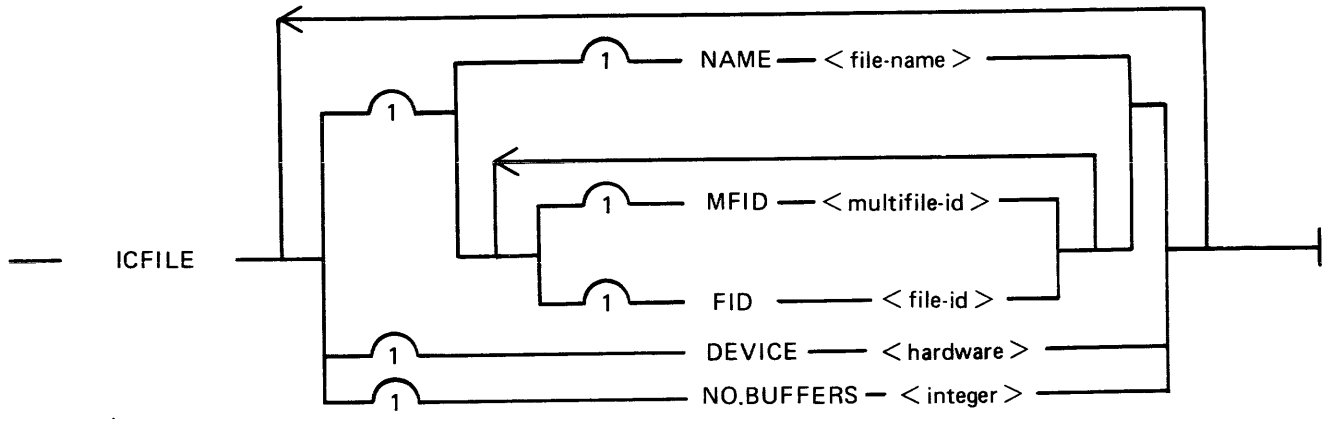
Example: MESSAGE $LIST $XMAP $NOWARNING
sets LIST and XMAP, resets WARNING

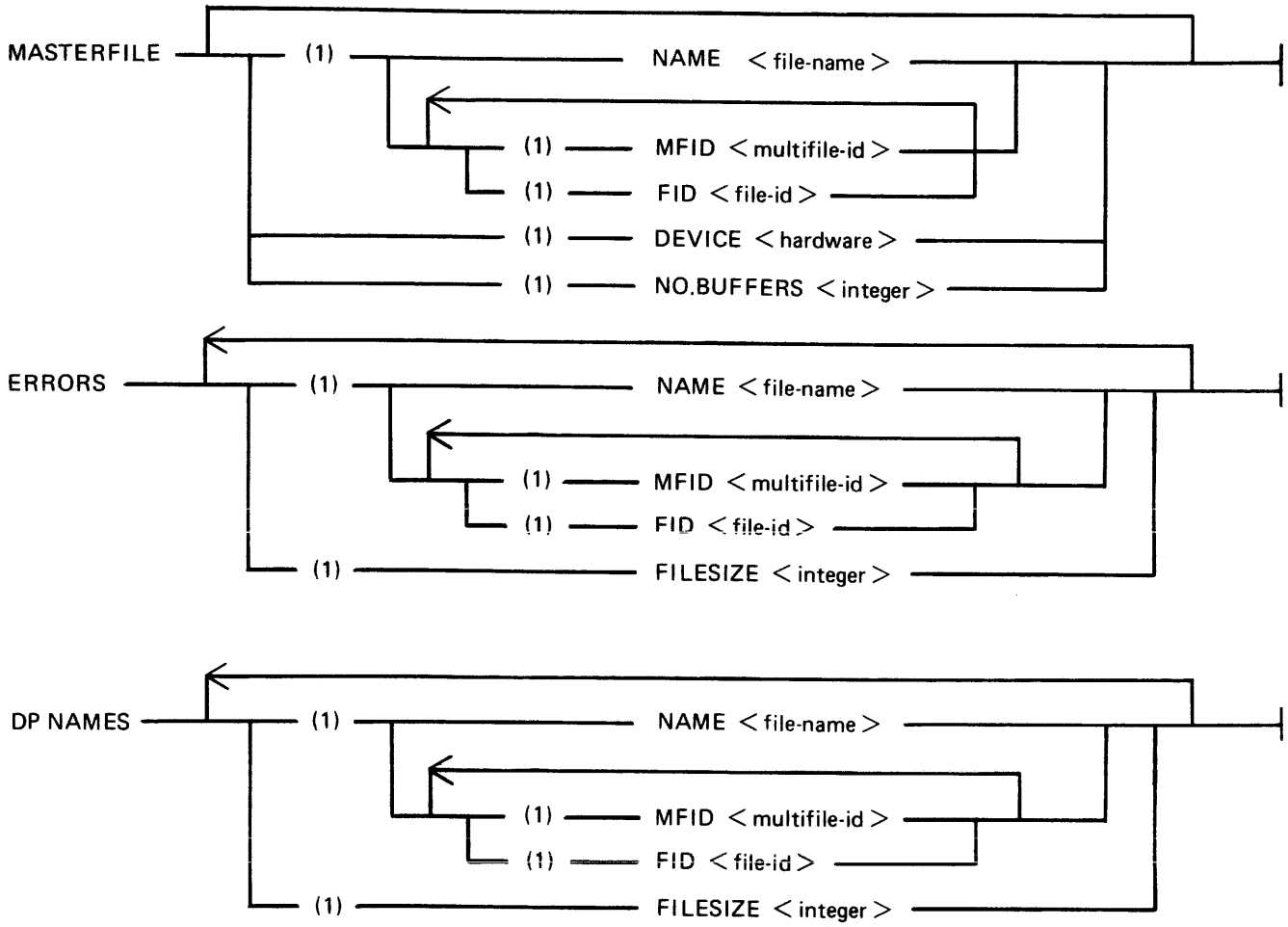



SOURCEIN
NAME < file-name >
MFID < multifile-id >
FID < file-id >
DEVICE < hardware >
NO.BUFFERS < integer >
SOURCEOUT
NAME < file-name >
MFID < multifile-id >
FID < file-id >
DEVICE < hardware >
FILESIZE < integer >
RECORD < integer >
RECORDS.BLOCK < integer >
BLOCK < integer >
NO.BUFFERS < integer >
WORKFILE MFID < multifile-id >

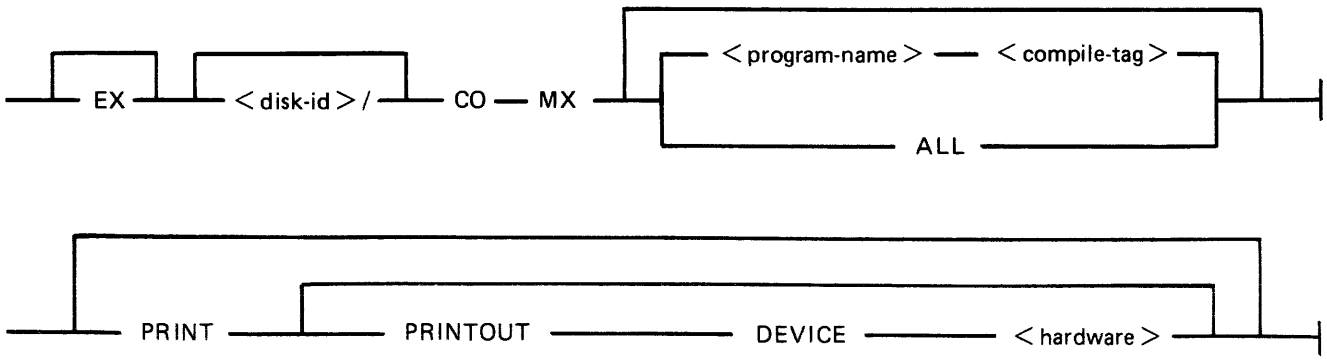

PRINTOUT
(1)
DEVICE
<hardware>
(1)
NO.BUFFERS
<integer>
(1)
FILESIZE
<integer>
(1)
DONTBACKUP
DONTCARE
MUSTBACKUP
NAME
<file-name>
IGNORE


CONTROLFILE
1
NAME
<file-name>
1
MFID
<multifile-id>
1
FID
<file-id>
1
DEVICE
<hardware>
1
NO.BUFFERS
<integer>


ICFILE
1
NAME
<file-name>
1
MFID
<multifile-id>
1
FID
<file-id>
1
DEVICE
<hardware>
1
NO.BUFFERS
<integer>

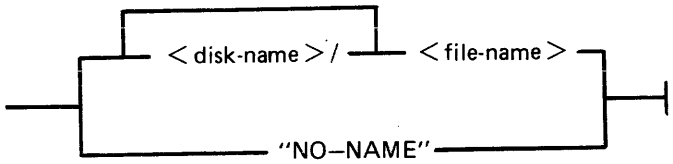

MASTERFILE
(1)
NAME <file-name>
(1)
MFID <multifile-id>
(1)
FID <file-id>
(1)
DEVICE <hardware>
(1)
NO.BUFFERS <integer>
ERRORS
(1)
NAME <file-name>
(1)
MFID <multifile-id>
(1)
FID <file-id>
(1)
FILESIZE <integer>
DP NAMES
(1)
NAME <file-name>
(1)
MFID <multifile-id>
(1)
FID <file-id>
(1)
FILESIZE <integer>

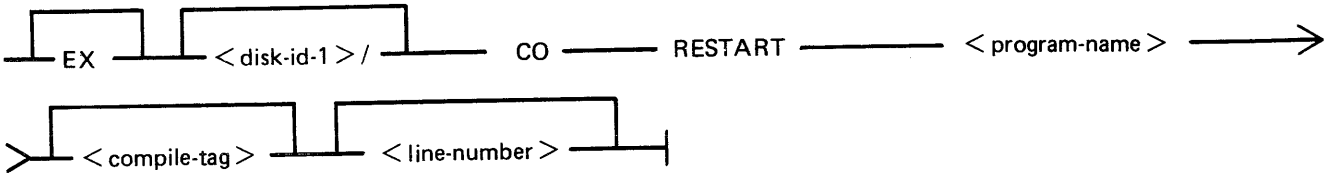
Version 2:


EX
<disk-id>/
CO
MX
<program-name>
<compile-tag>
ALL
PRINT
PRINTOUT
DEVICE
<hardware>

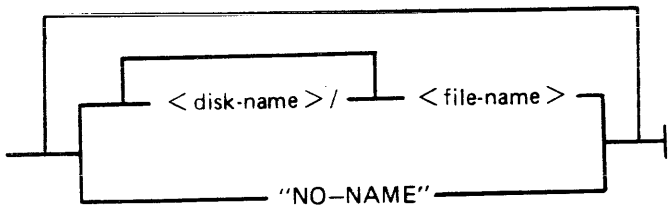
CO Utility (Continued)

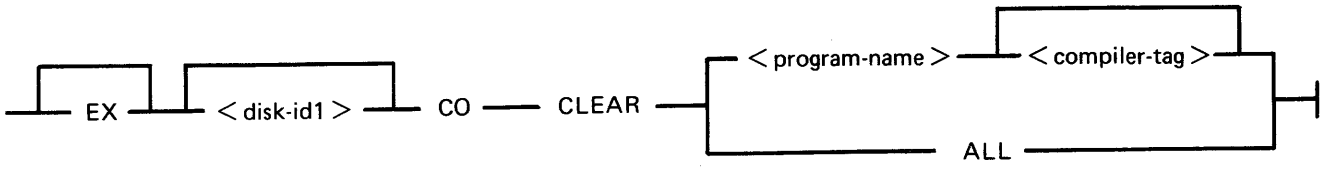
Where <program-name> is defined:



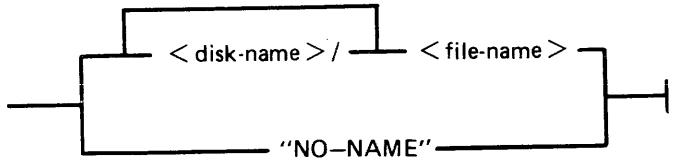


Where <program-name> is defined:

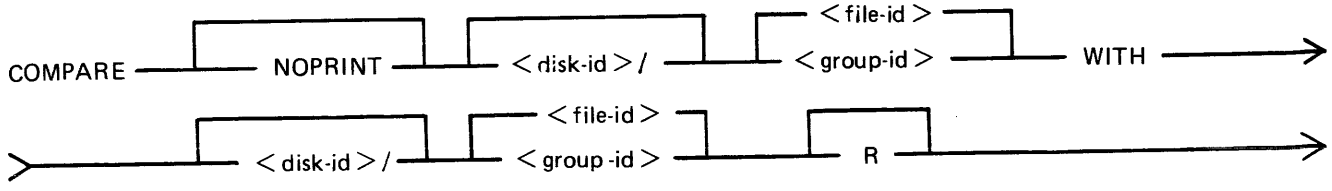


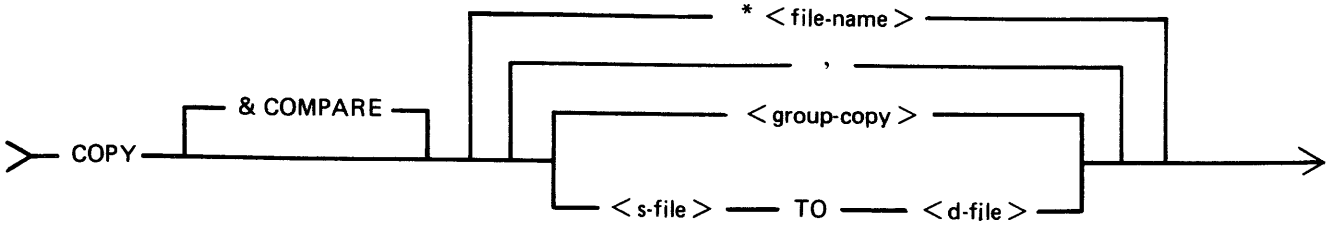


Where <program-name> is defined:

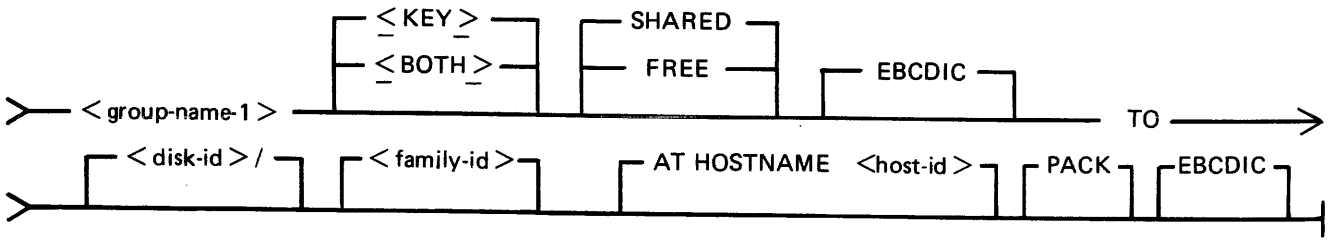


COMPARE Utility

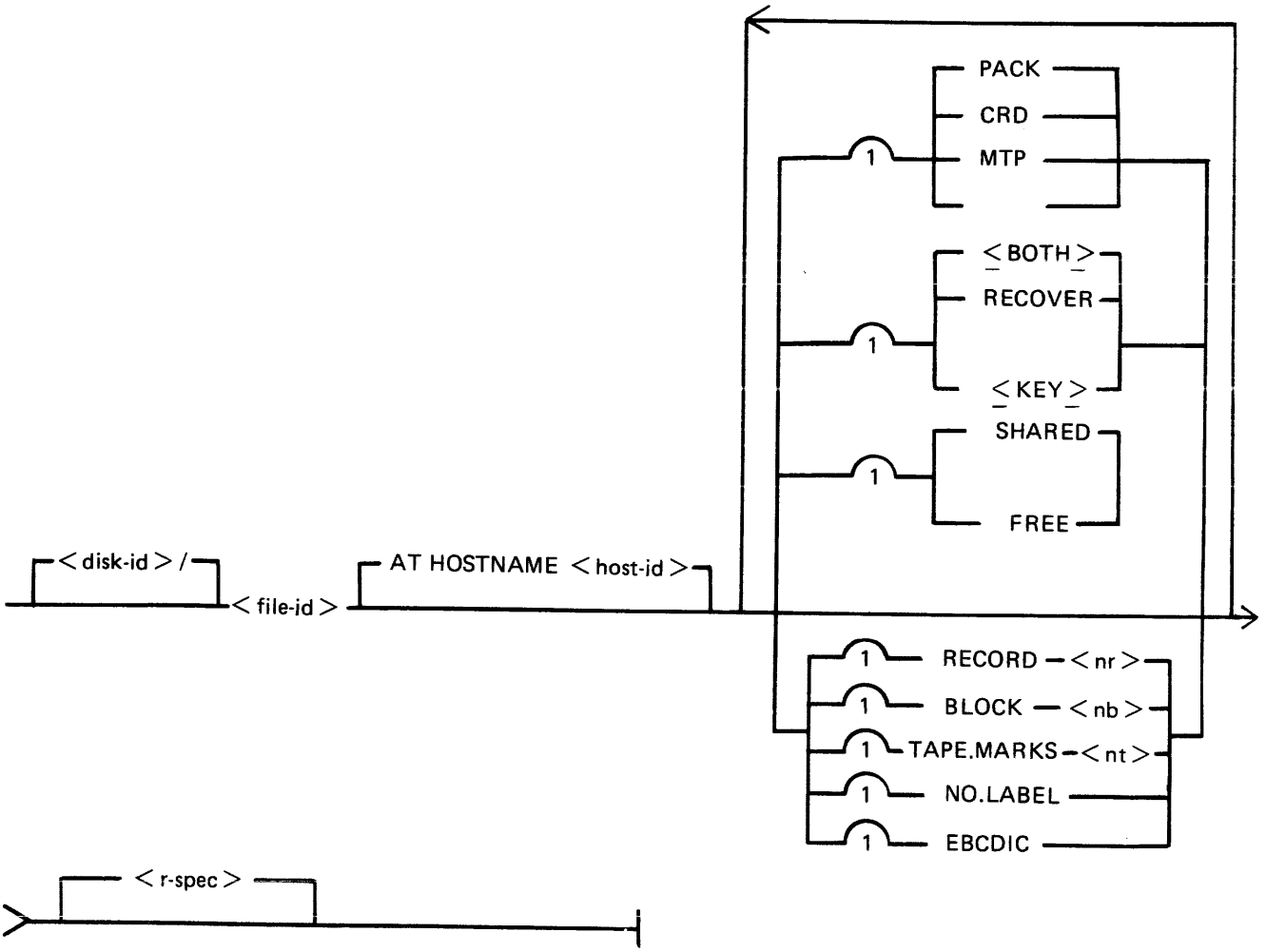




<group-copy> is defined as :

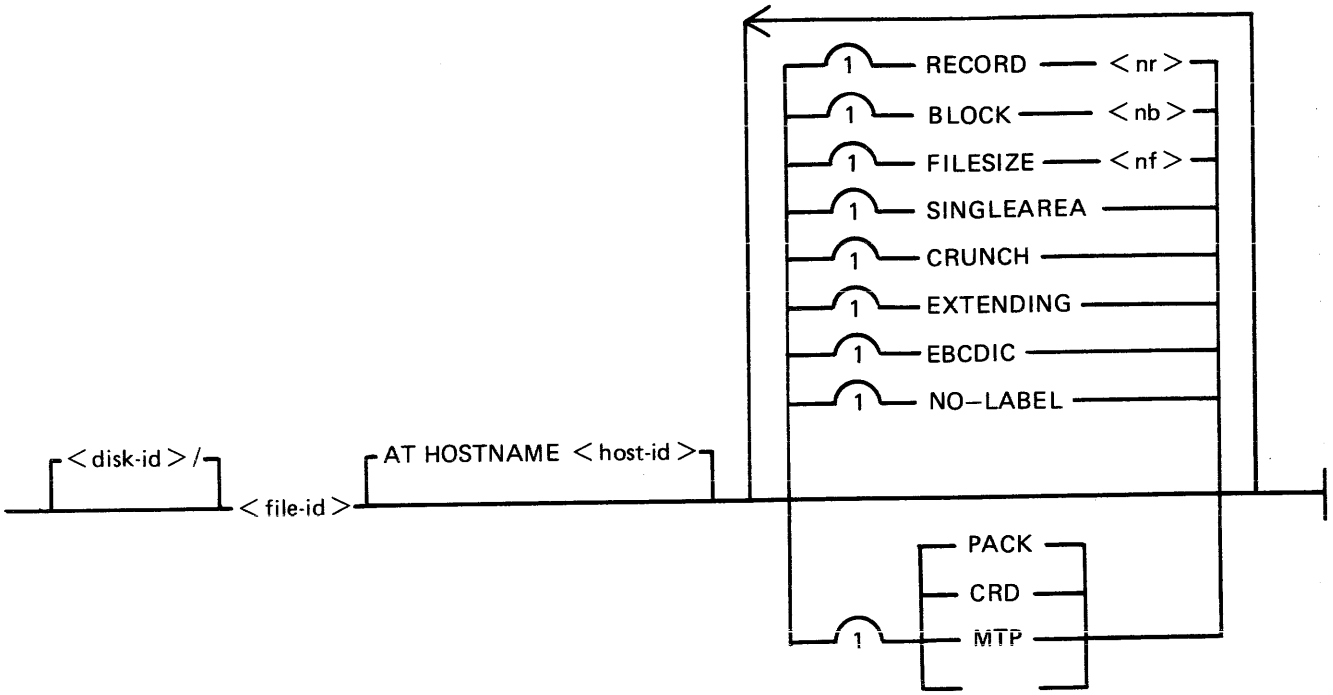


<s-file> is defined as :

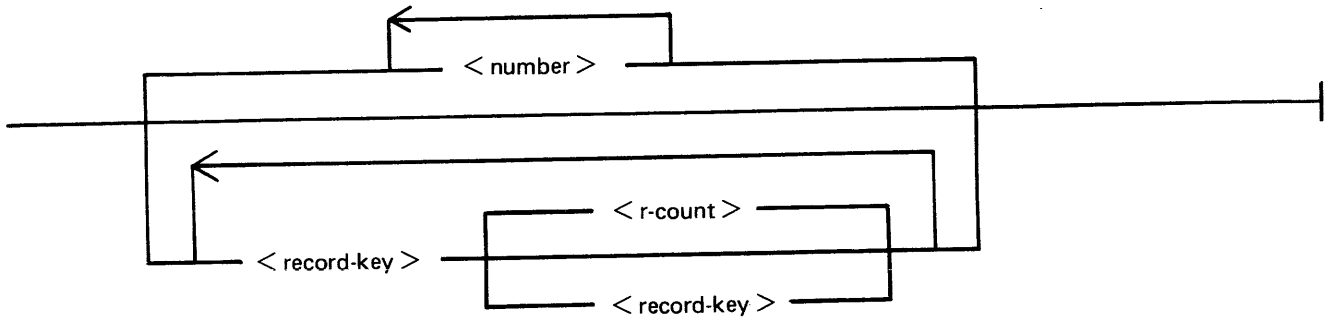


COPY Utility (Continued)

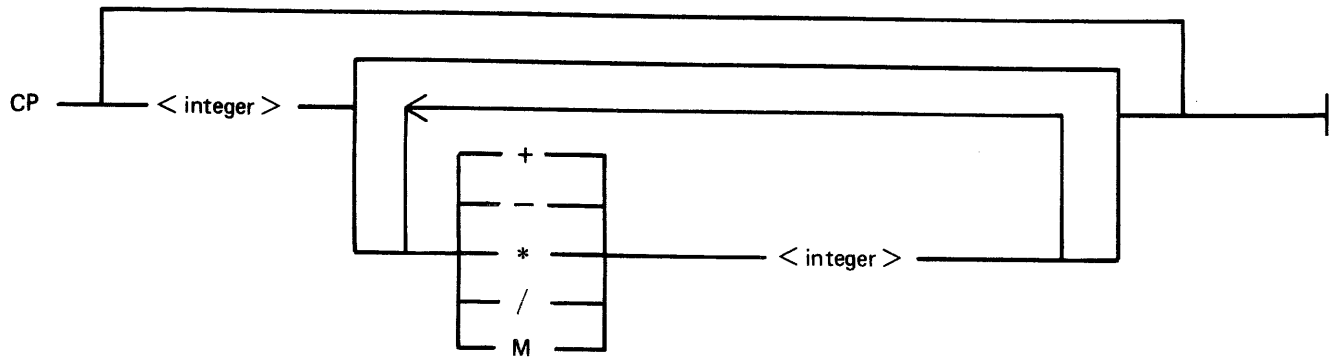
< d-file > is defined as :



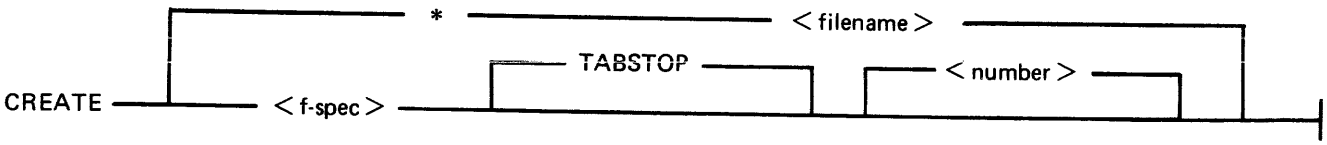
< r-spec > is defined as :



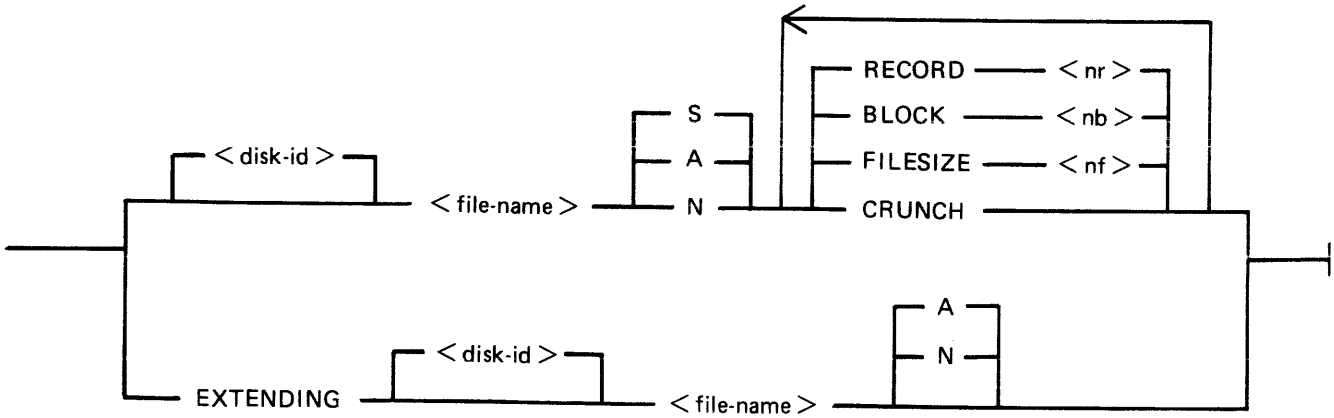
CP Utility



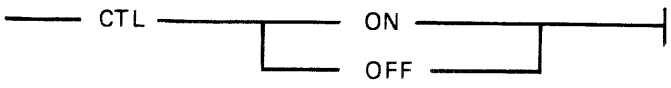
CREATE Utility



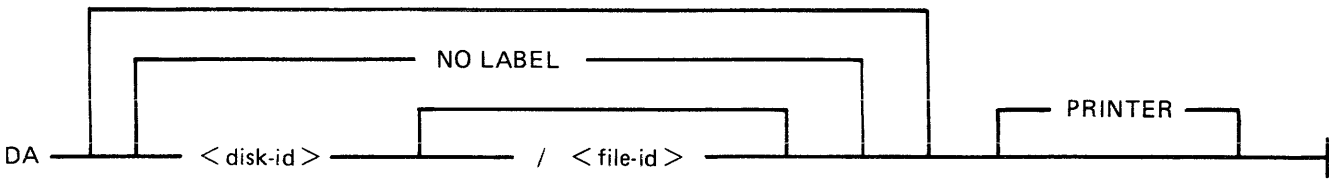
< f-spec > is defined as :

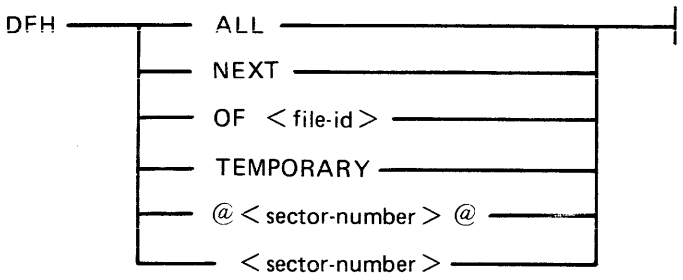


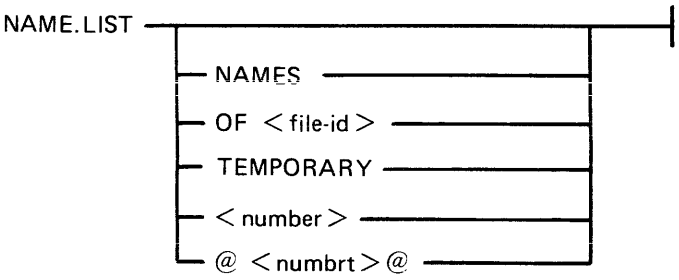
CTL Intrinsic

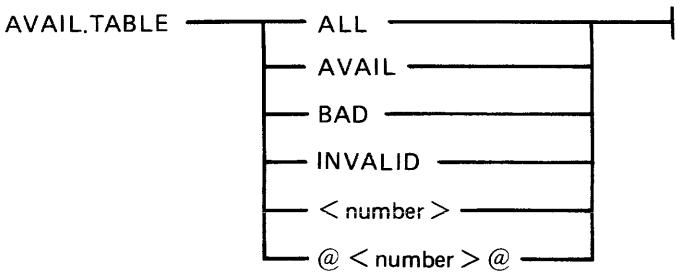


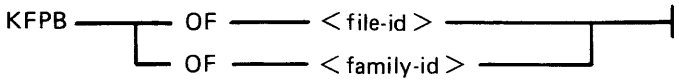
DA Utility


DA
NO LABEL
< disk-id >
/ < file-id >
PRINTER

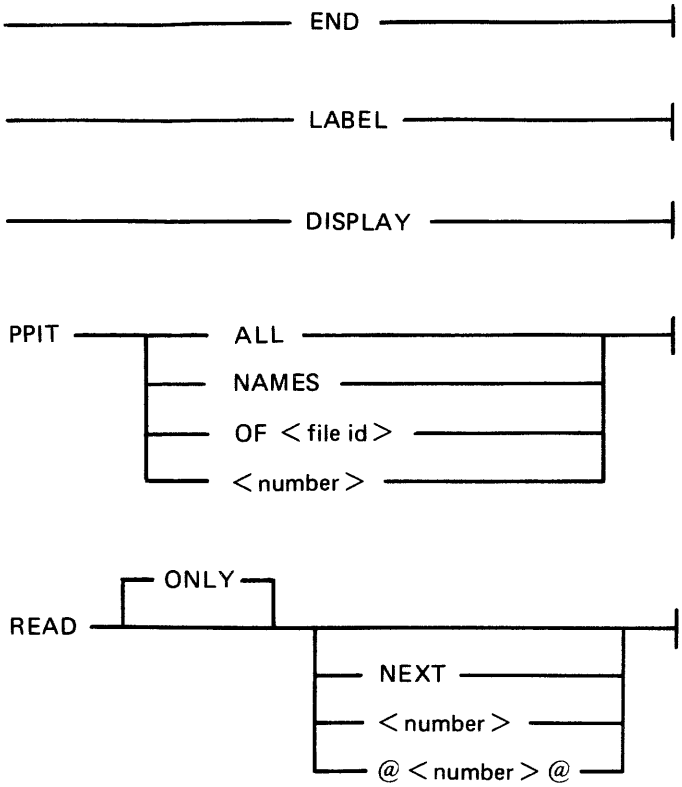

DFH
ALL
NEXT
OF < file-id >
TEMPORARY
@ < sector-number > @
< sector-number >


NAME.LIST
NAMES
OF < file-id >
TEMPORARY
< number >
@ < numbrt > @

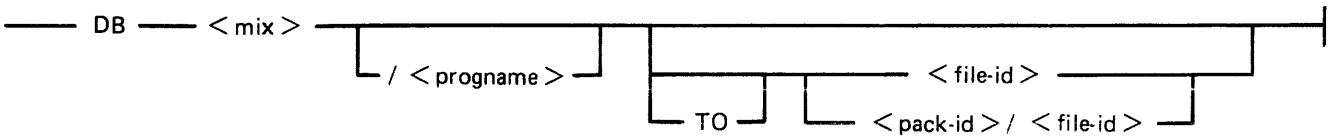

AVAIL.TABLE
ALL
AVAIL
BAD
INVALID
< number >
@ < number > @


KFPB
OF
< file-id >
OF
< family-id >

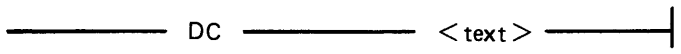
DA Utility (Continued)



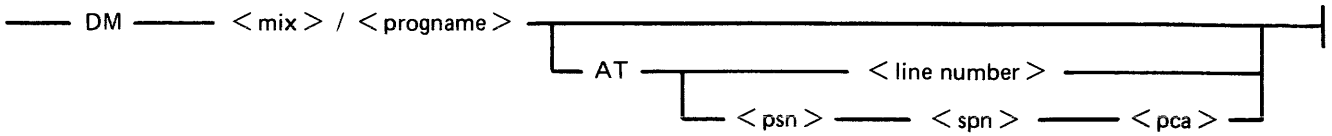
DB Intrinsic (not on B 900)



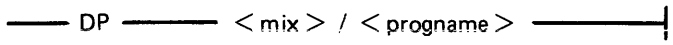
DC Intrinsic



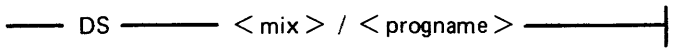
DM Intrinsic (B 1000 only)



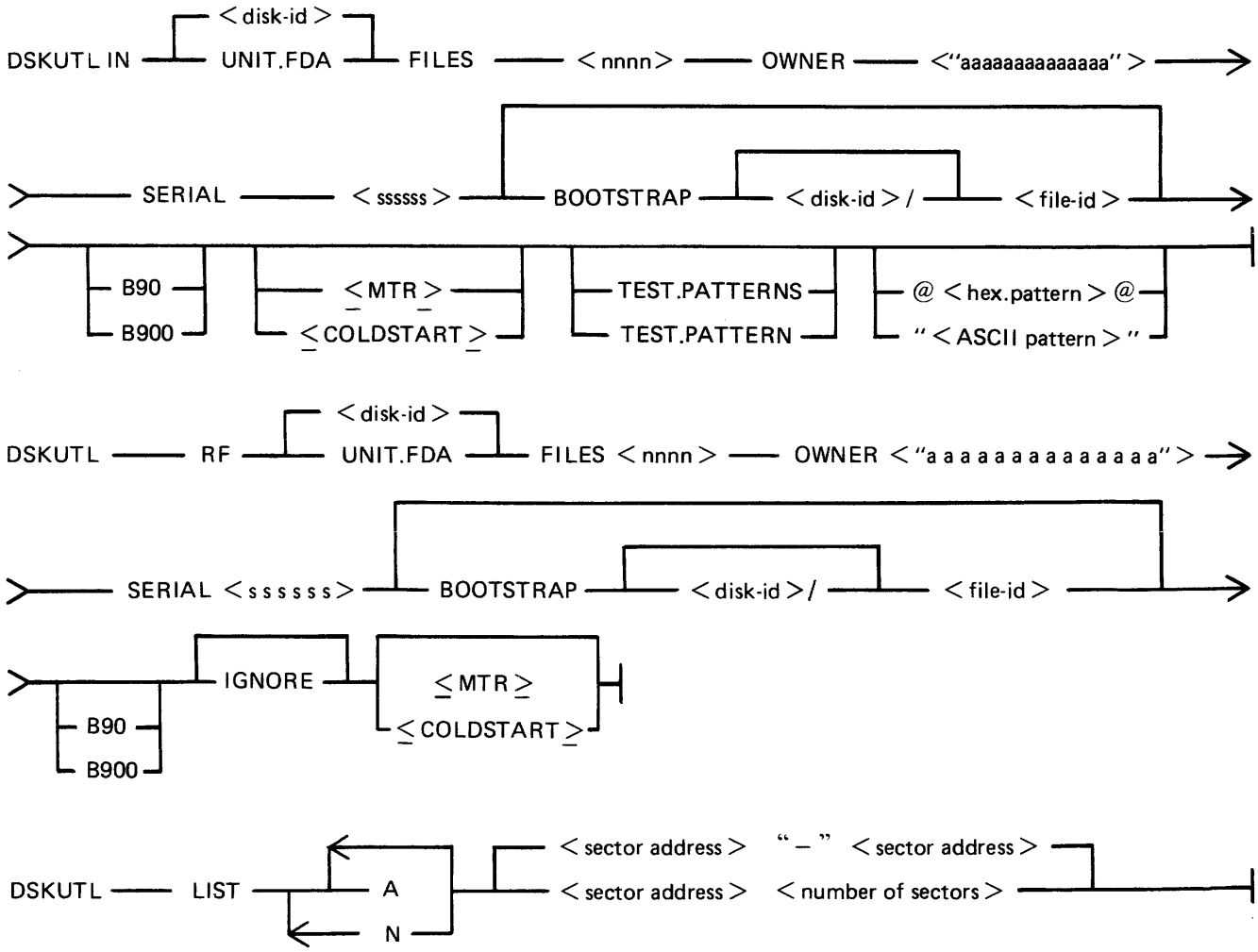
DP Intrinsic



DS Intrinsic

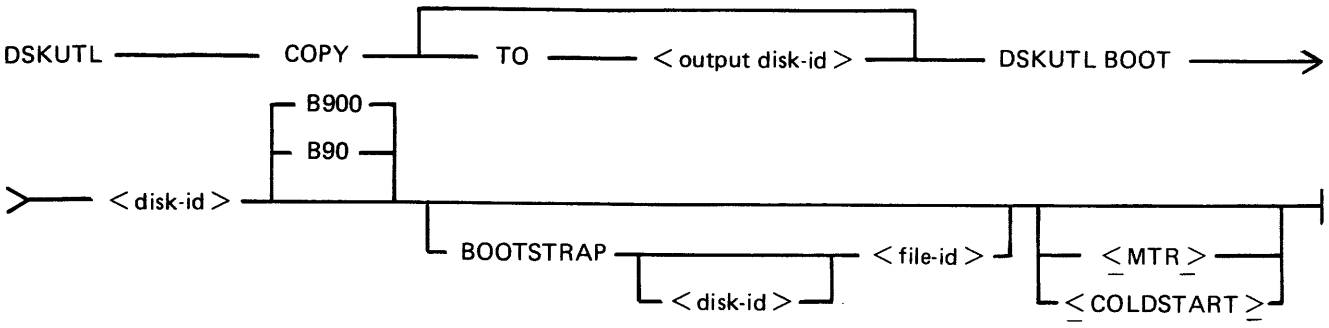


DSKUTL Utility

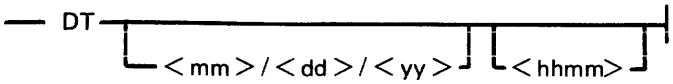


2032801

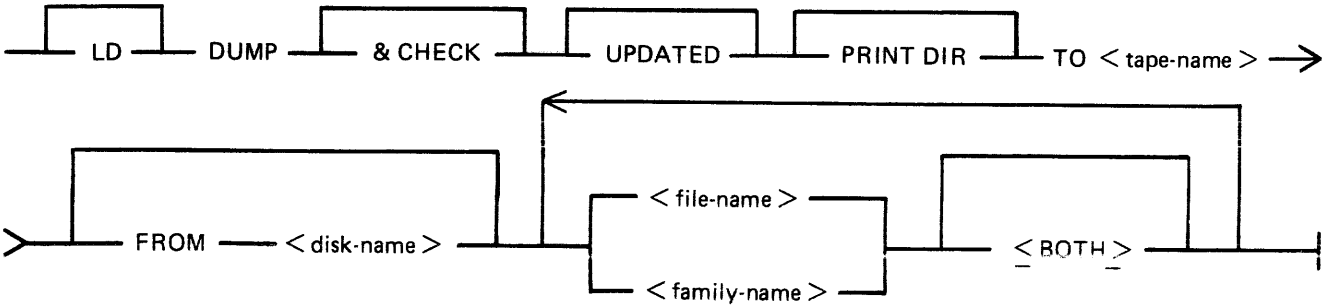
DSKUTL Utility (Continued)



DT Intrinsic



DUMP



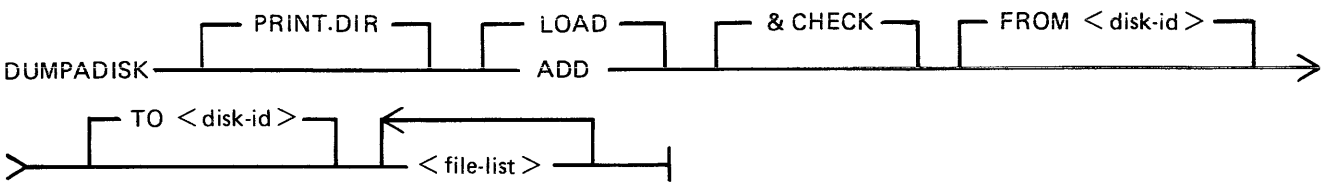
DUMPADISK Utility

Format 1

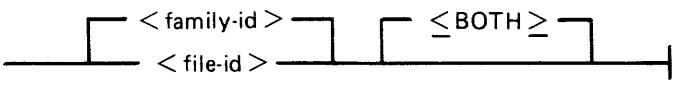
DUMPADISK PRINT.DIR

This function lists the disk directory.

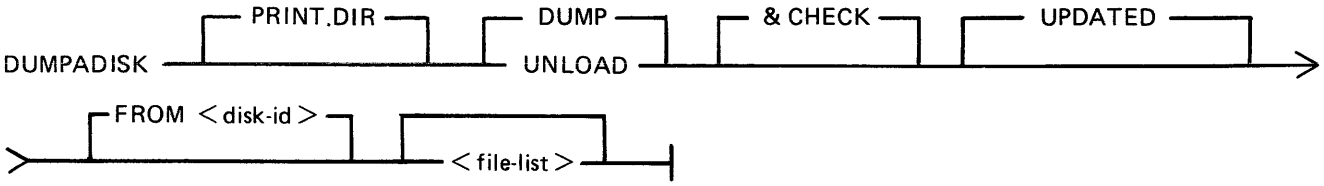
Format 2



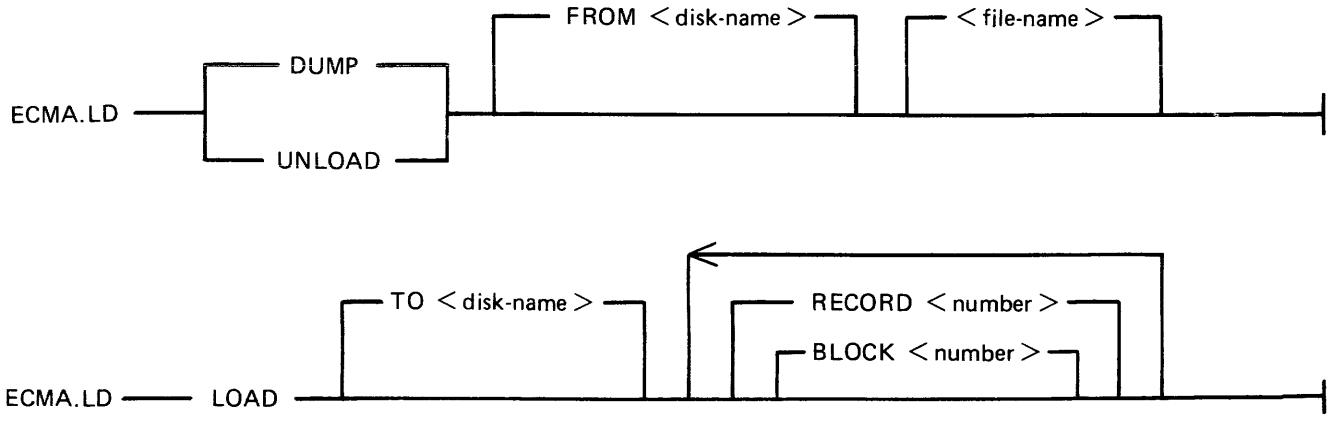
where < file-list > is defined as :



Format 3

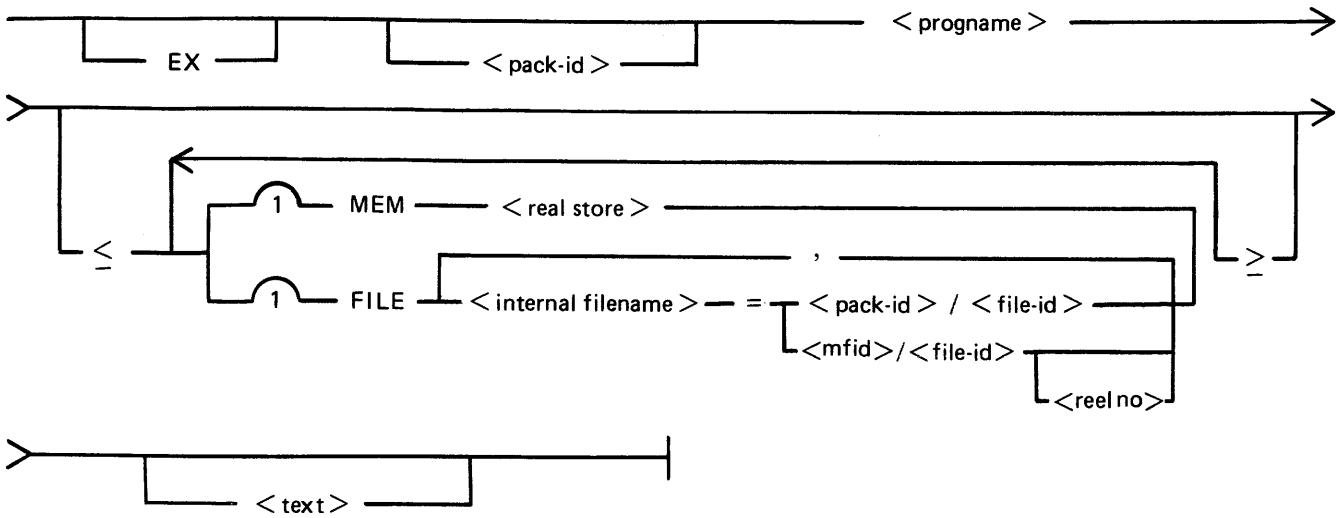


ECMA Utility

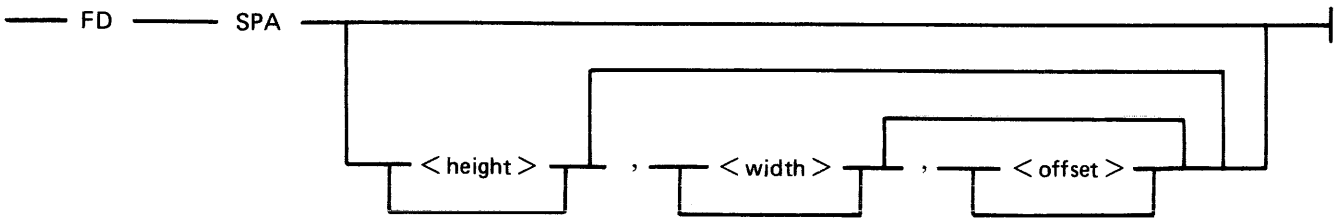


2032801

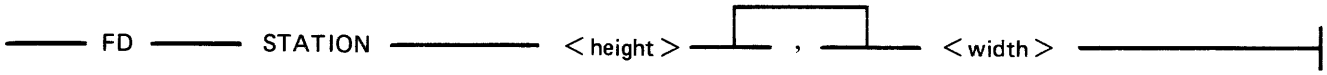
EX Intrinsic



FD Intrinsic (B 90 only)

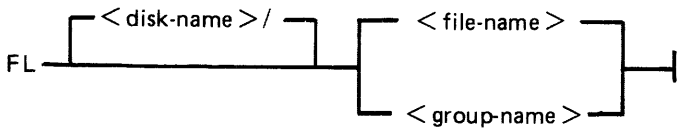


For terminal SPOs running Multiple Terminal SPO (MTS), B 90 systems only:

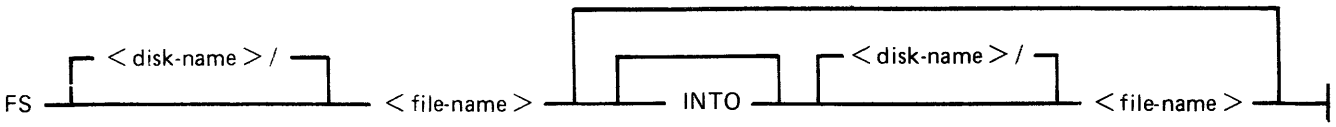


FILEUTL Utility

FL Utility

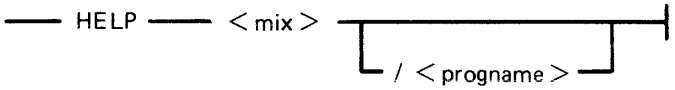


FS Utility



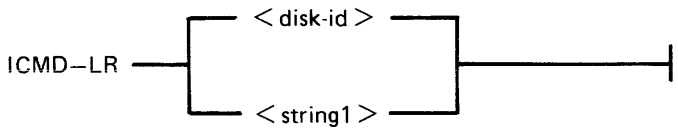
GO Intrinsic

HELP Intrinsic

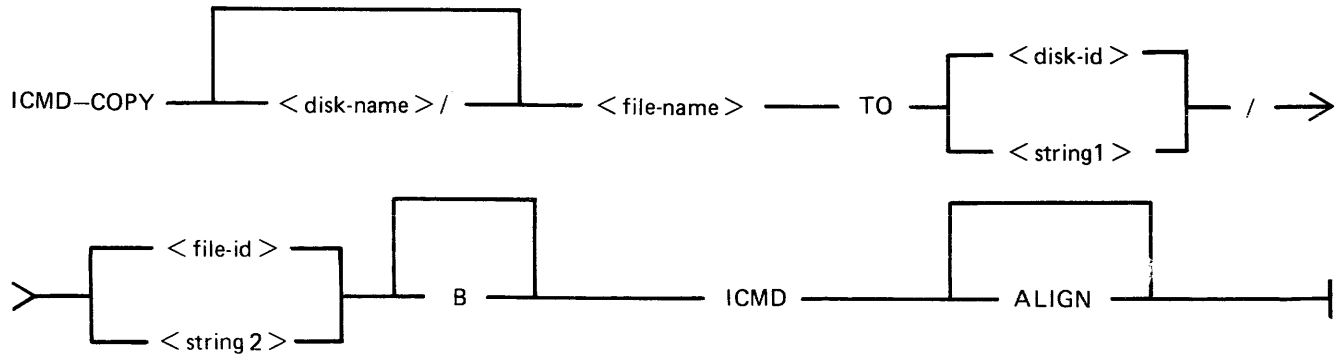


ICMD Utility

Version 1

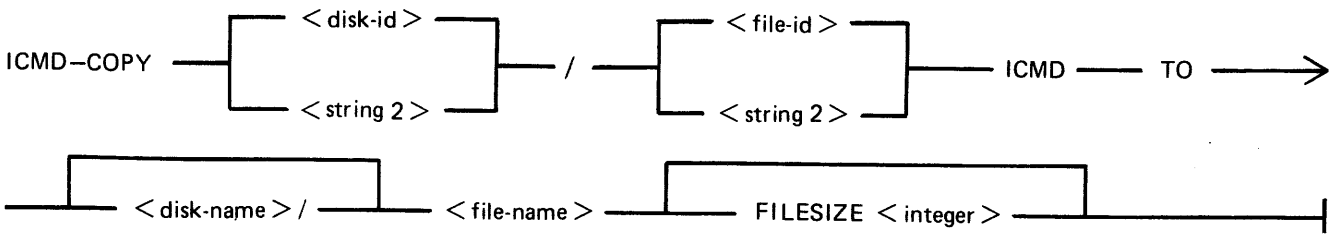


Version 2

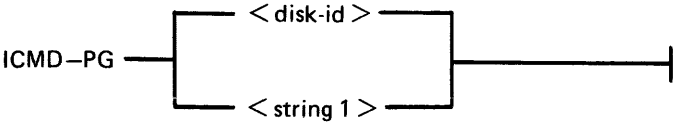


ICMD Utility (Continued)

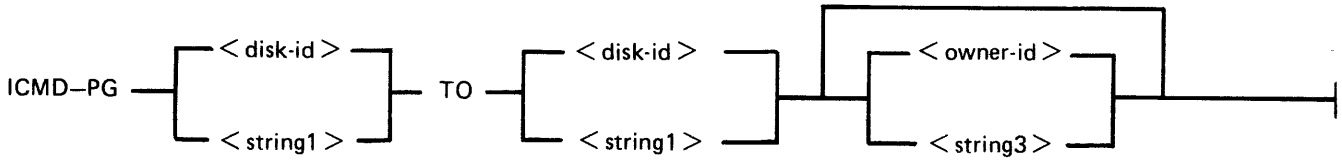
Version 3



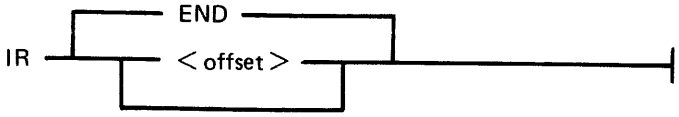
Version 4



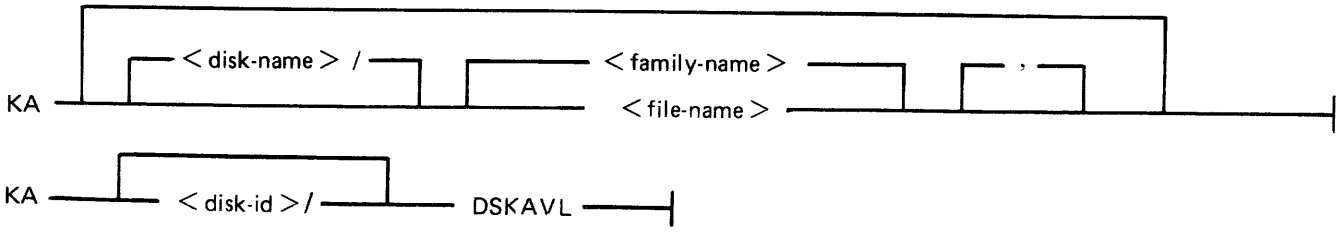
Version 5



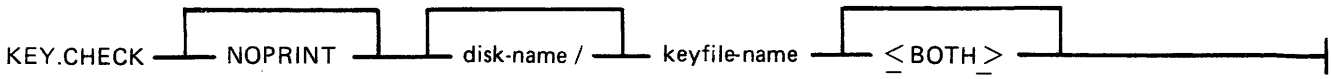
IR Utility



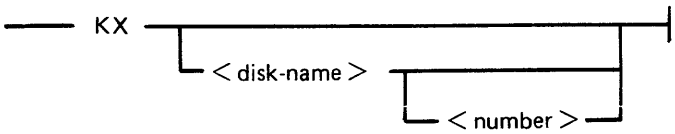
KA Utility



KEY.CHECK Utility



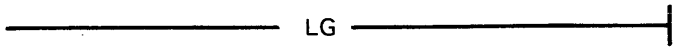
KX Utility



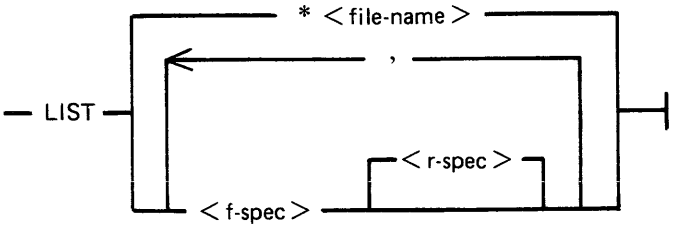
LB Utility

LF Utility

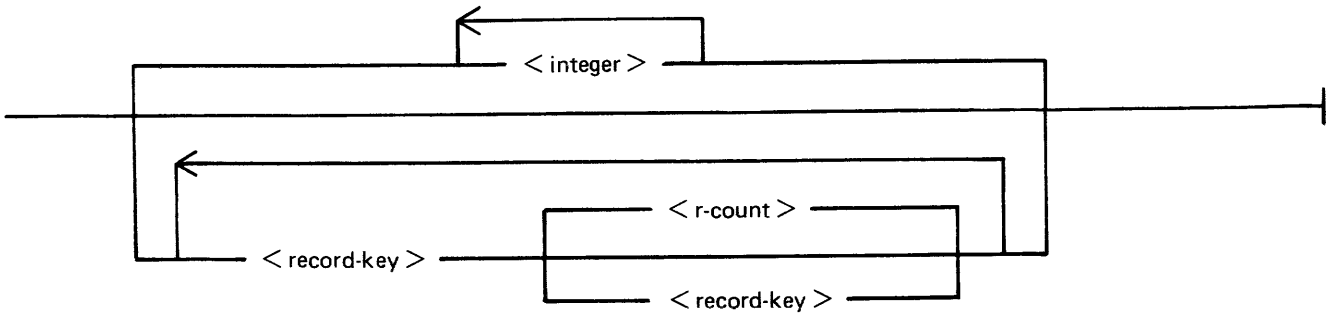
LG Intrinsic



LIST Utility

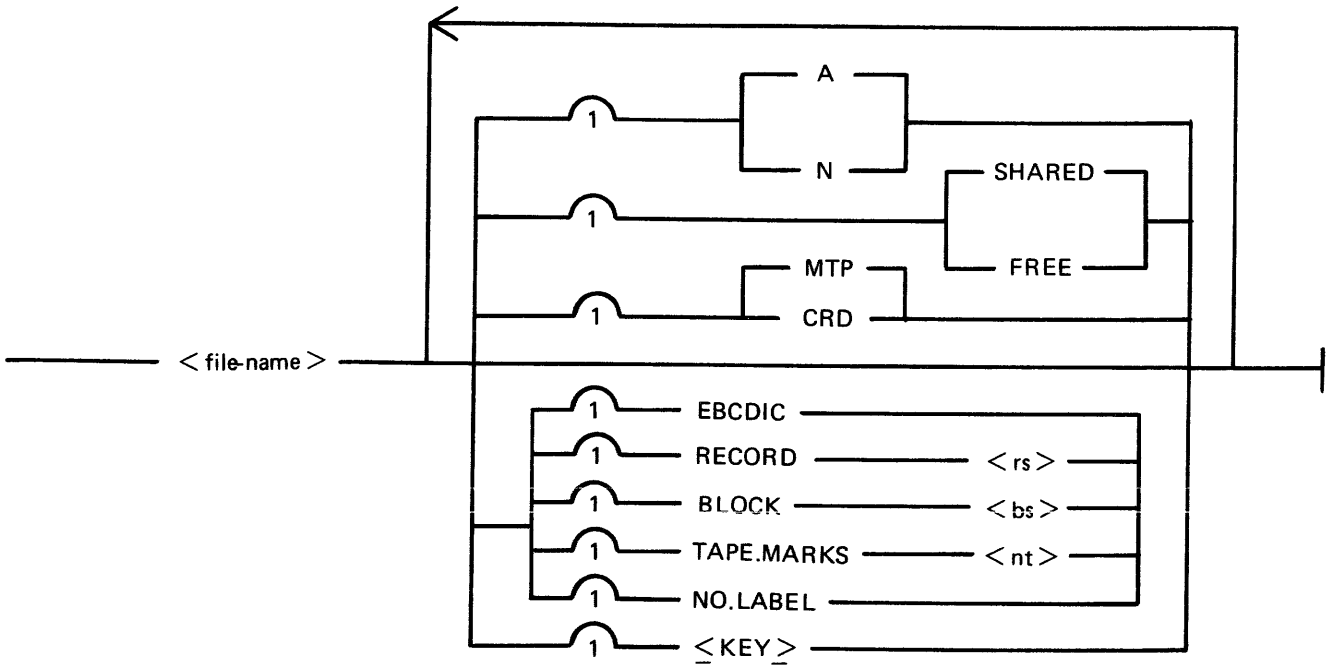


< r-spec > is defined as :

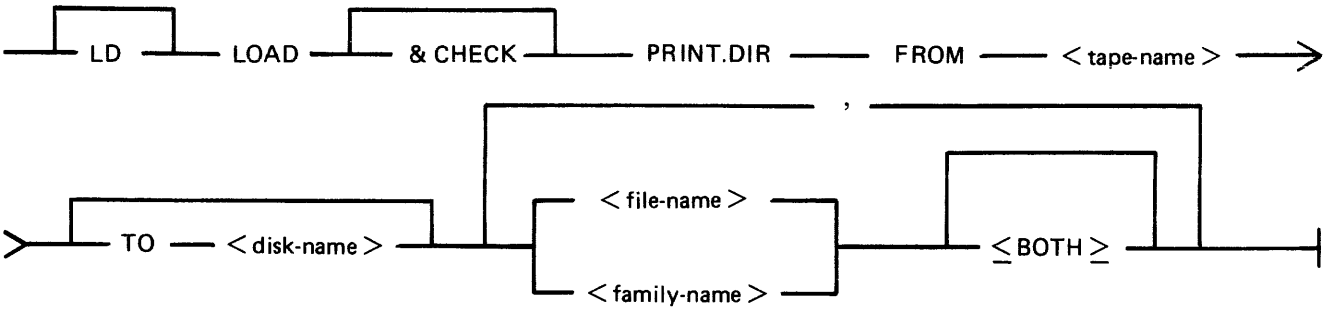


LIST Utility (Continued)

< f-spec > is defined as :

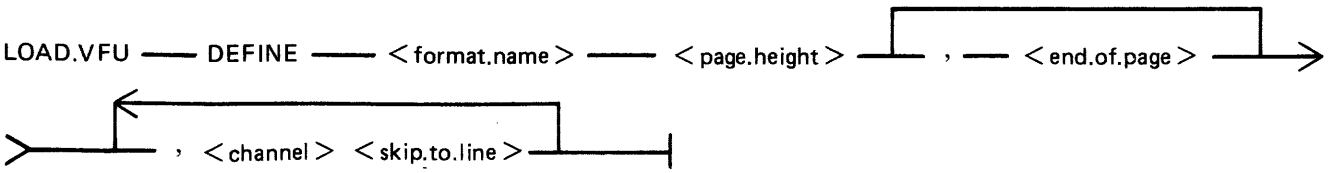


LOAD Utility



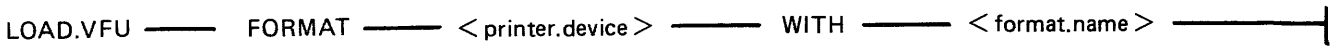
LOAD.VFU Utility

DEFINE

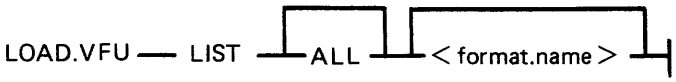


LOAD.VFU Utility (Continued)

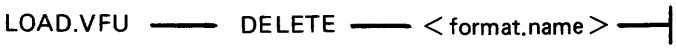
FORMAT



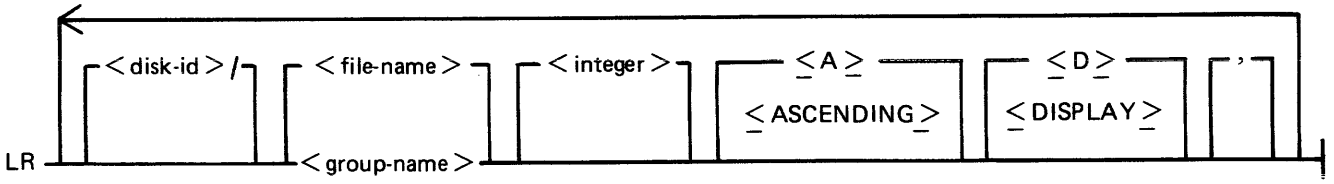
LIST



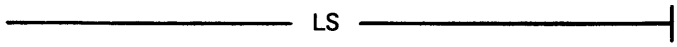
DELETE



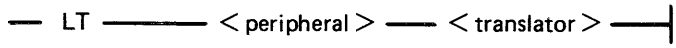
LR Utility



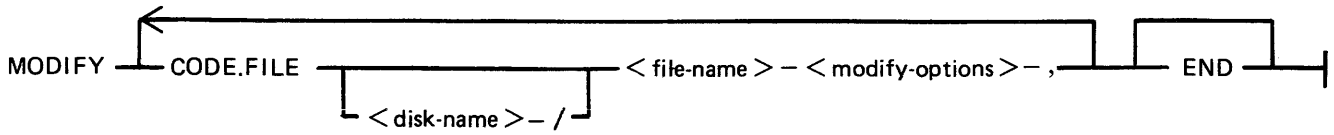
LS Intrinsic



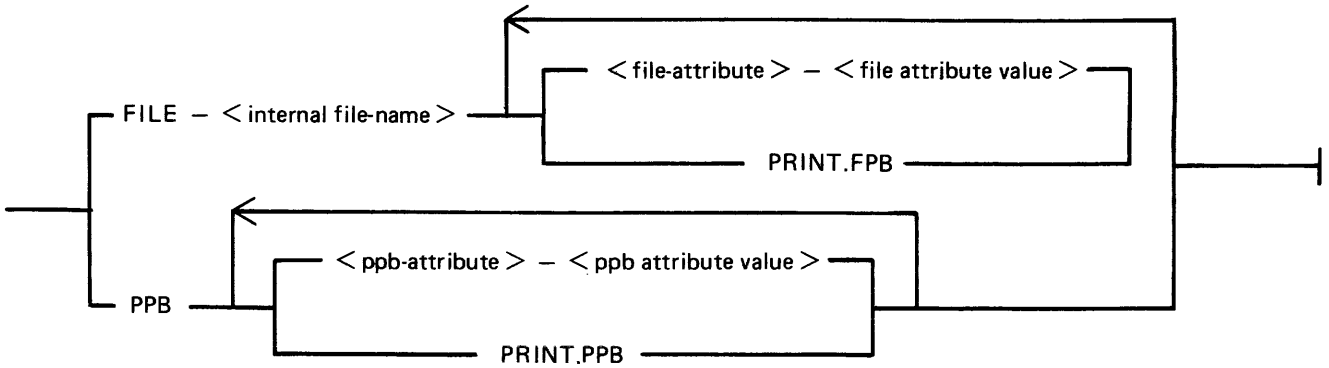
LT Intrinsic



MODIFY Utility

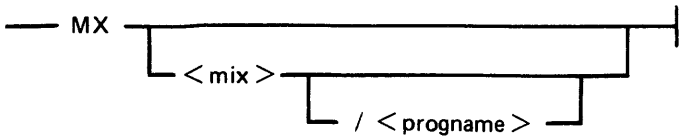


where < modify-options >

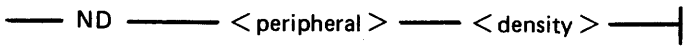


2032801

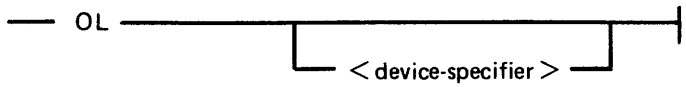
MX Intrinsic



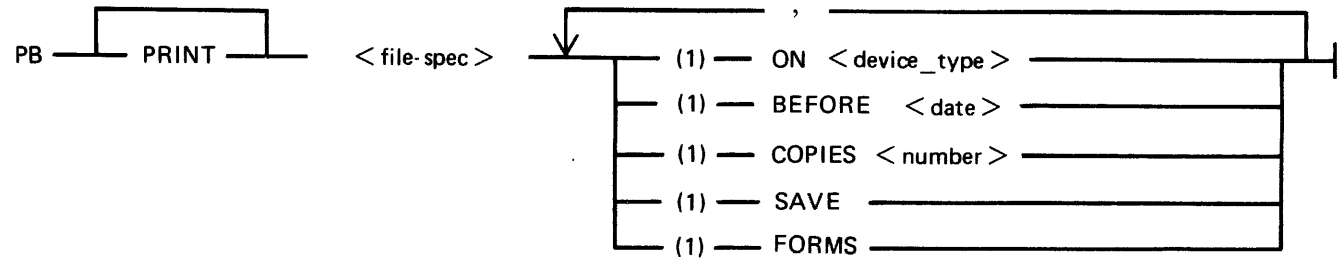
ND Intrinsic

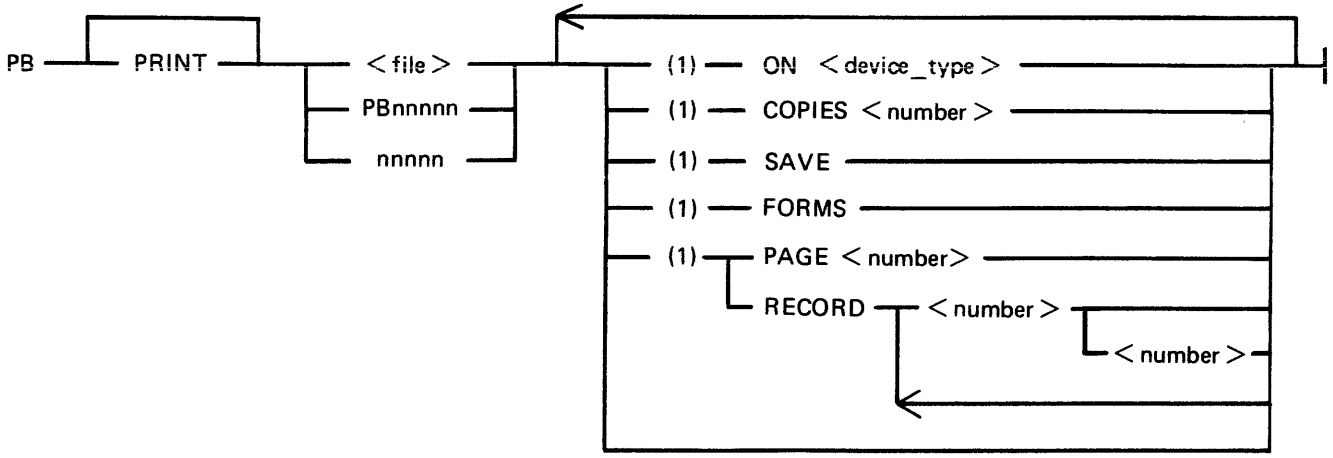


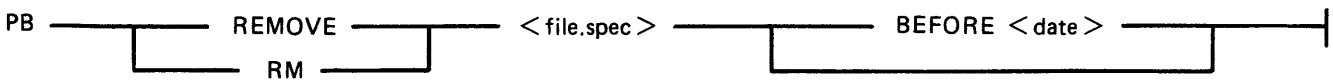
OL Intrinsic

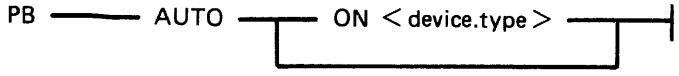


PB Utility

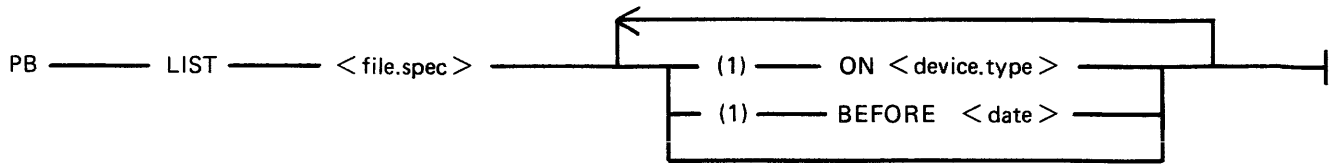




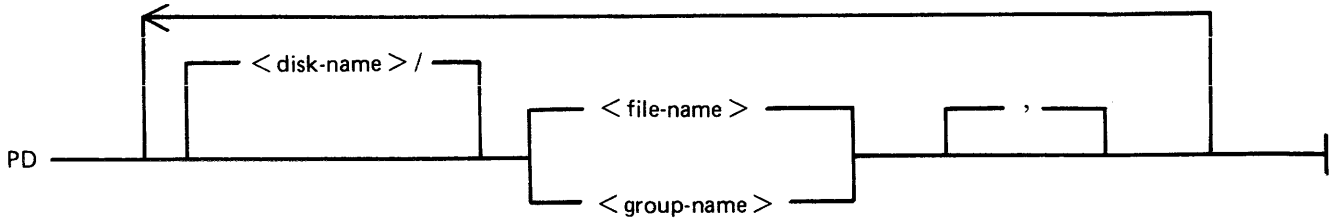




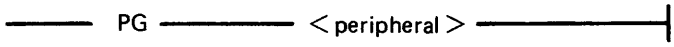
PB Utility (Continued)



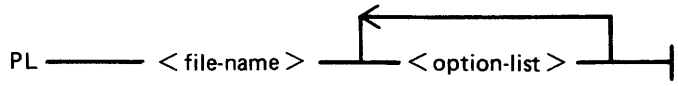
PD Utility



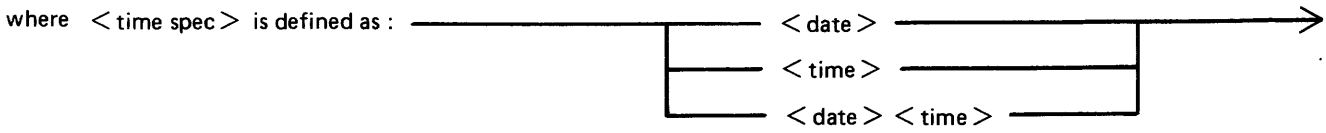
PG Intrinsic



PL Utility



where <option list> – SYSTEM or S
ERROR or E
STATS
DISPLAY
BACKUP
FROM <time spec>
TO <time spec>
ENTRY <number>
MIX <number>
INPUT
OUTPUT
<empty>
DEVICE <device spec>
PURGE

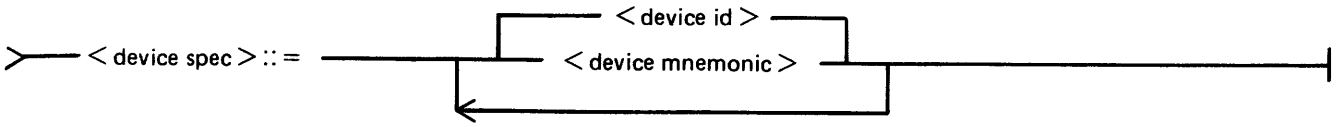


< date > : = { MM/DD/YY }

< time > : = { HH : MM : SS }

2032801

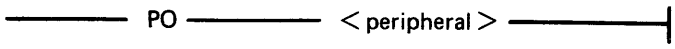
PL Utility (Continued)



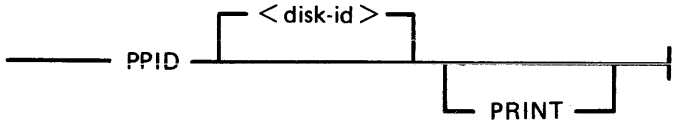
<device-id> ::= form DFA, CTB etc.

<device mnemonic> ::= form DF, CT etc.

PO Intrinsic

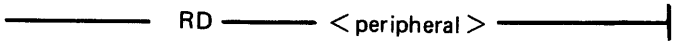


PPID Utility

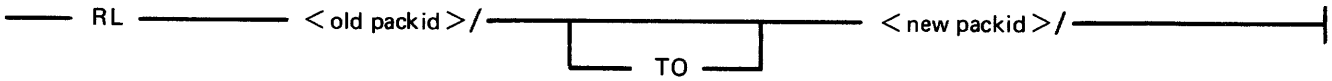


PR Intrinsic

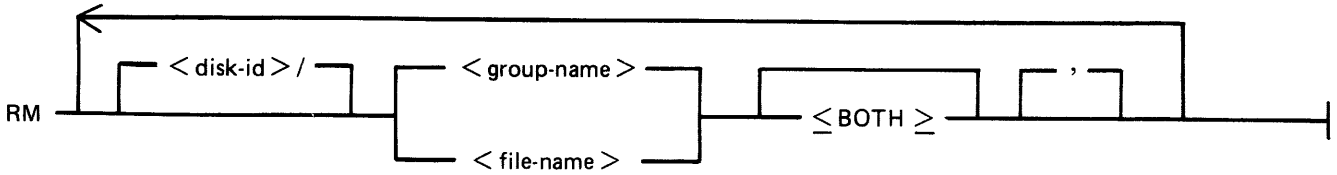
RD Intrinsic



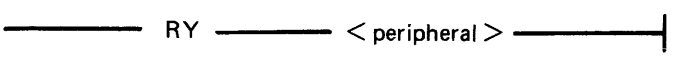
RL Utility



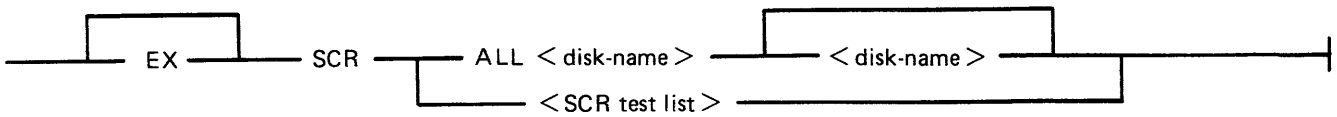
RM Utility



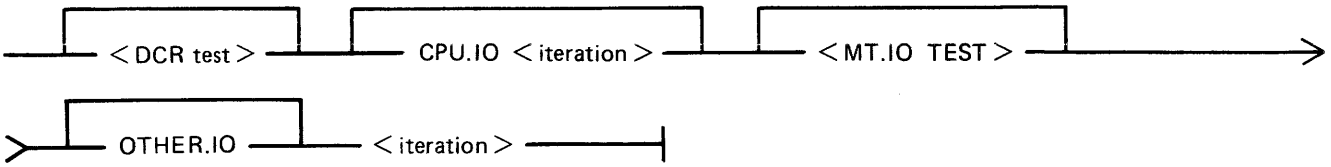
RY Intrinsic



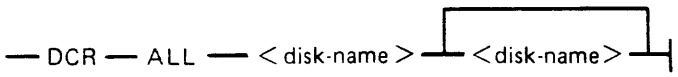
SCR Utility



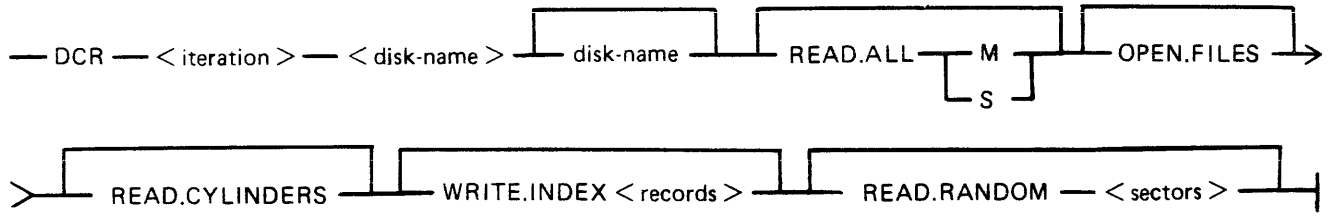
iteration = the number of iterations of the subsection.
Where SCR test list is:



Where DCR test is:



or

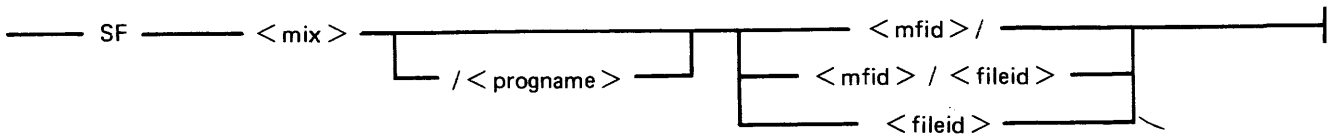


records = the number of (data) records in the index file
sector = the number of sectors to be read randomly.

Where MT.IO test is:

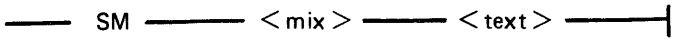
reels = the number of magnetic tape reels to be tested.
drives = the number of cassette drives to be tested.
blocks = the number of records to be written.

SF Intrinsic (not on B 90)

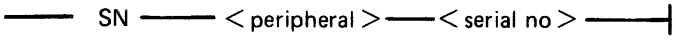


2032801

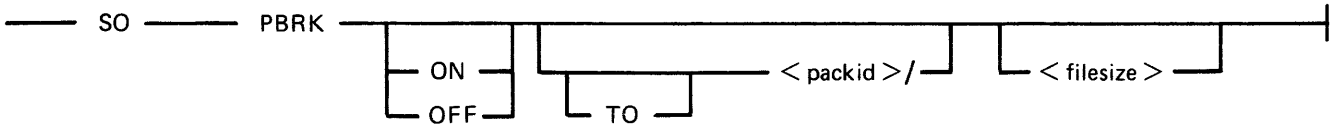
SM Intrinsic



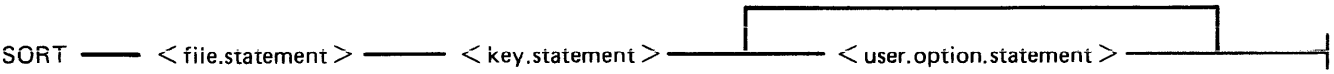
SN Intrinsic



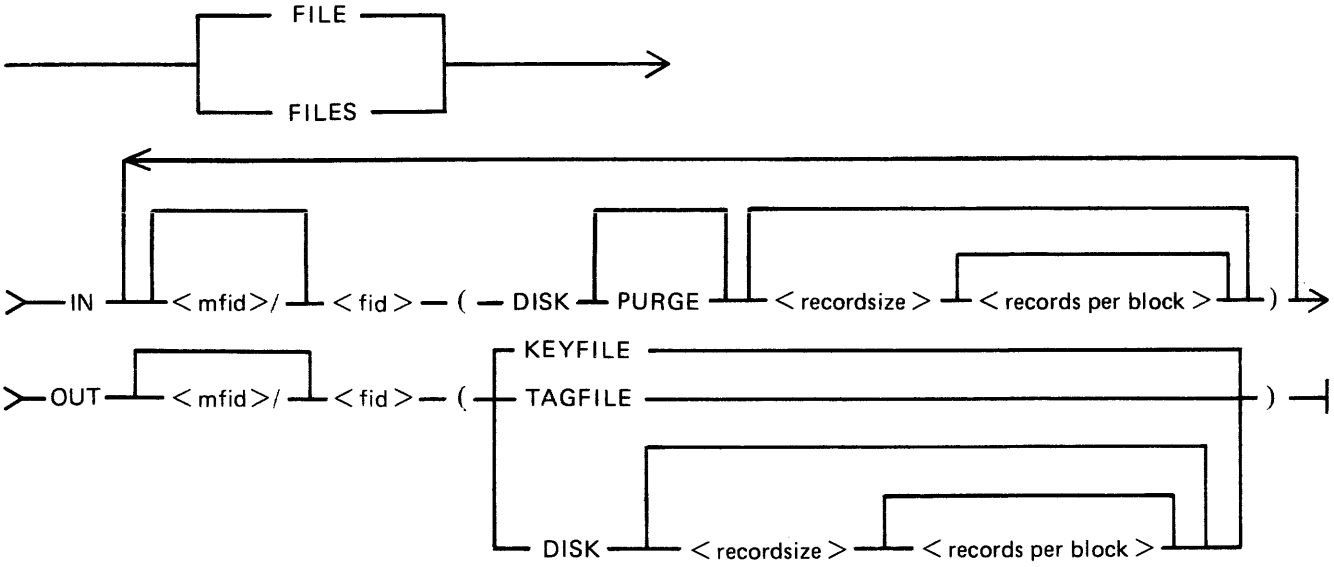
SO PRBK Intrinsic



SORT Utility

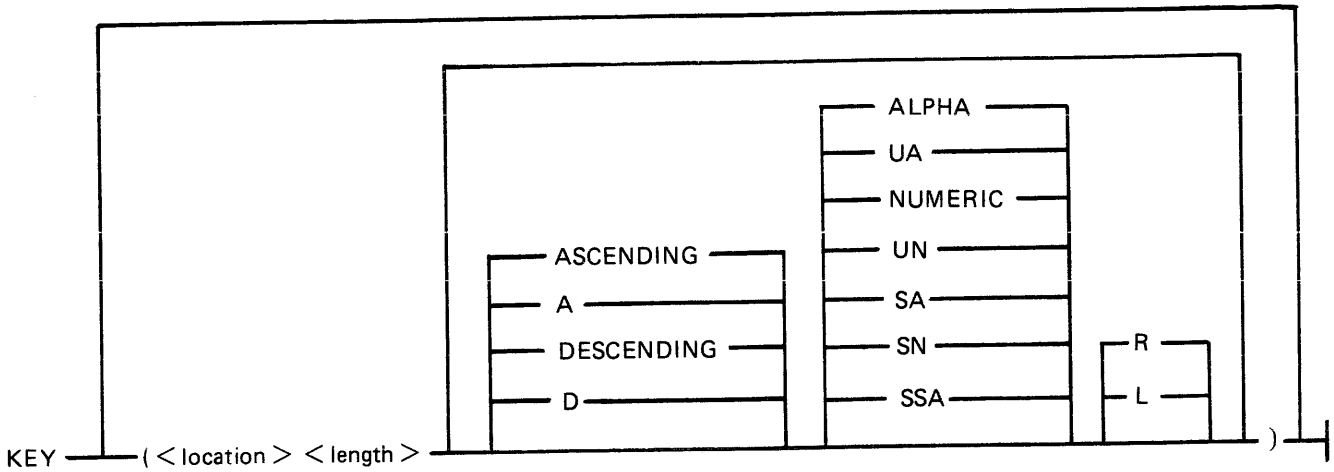


< file.statement >

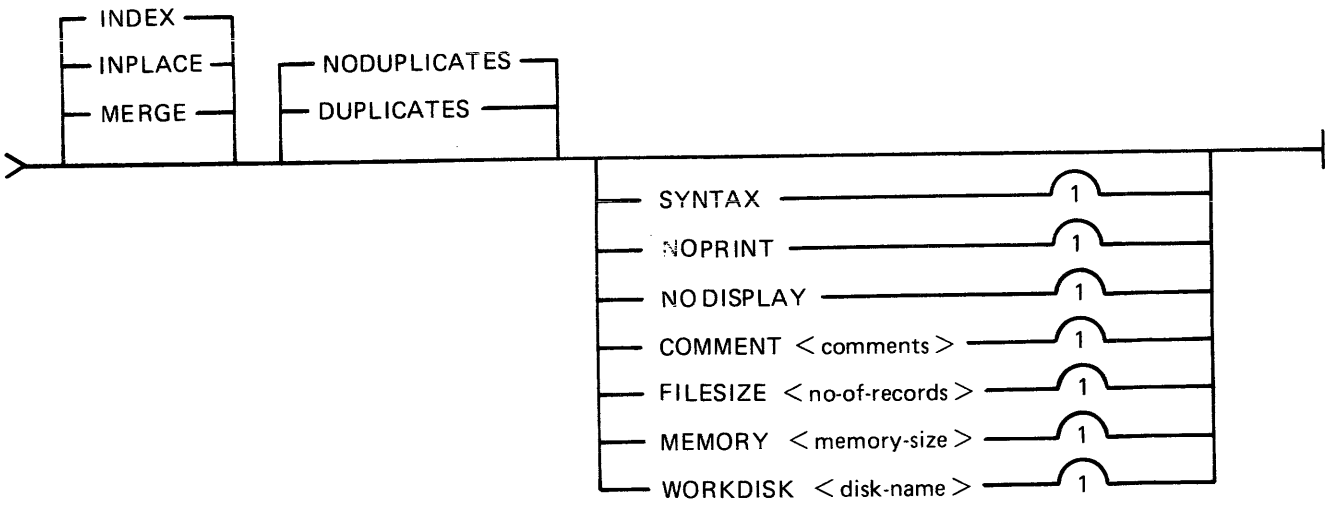


SORT Utility (Continued)

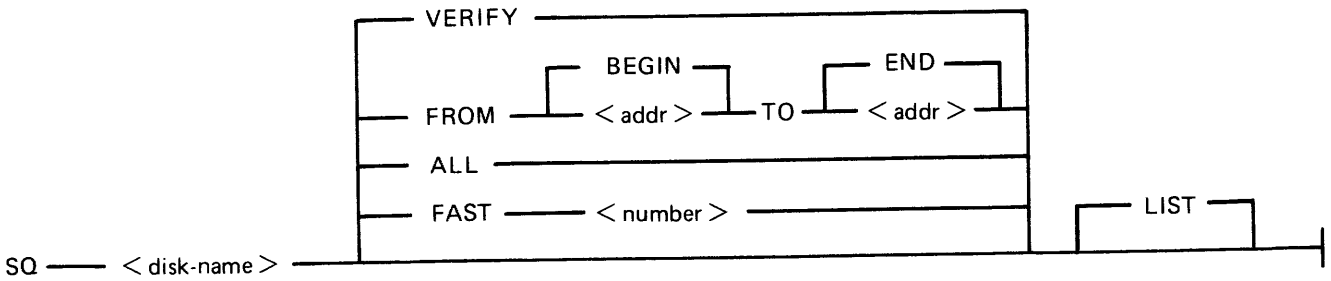
< key.statement >



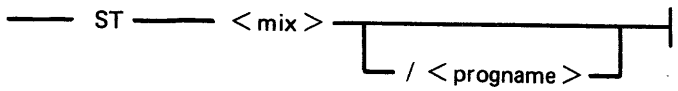
< user.op >



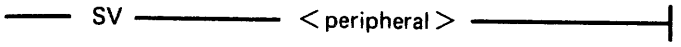
SQ Utility



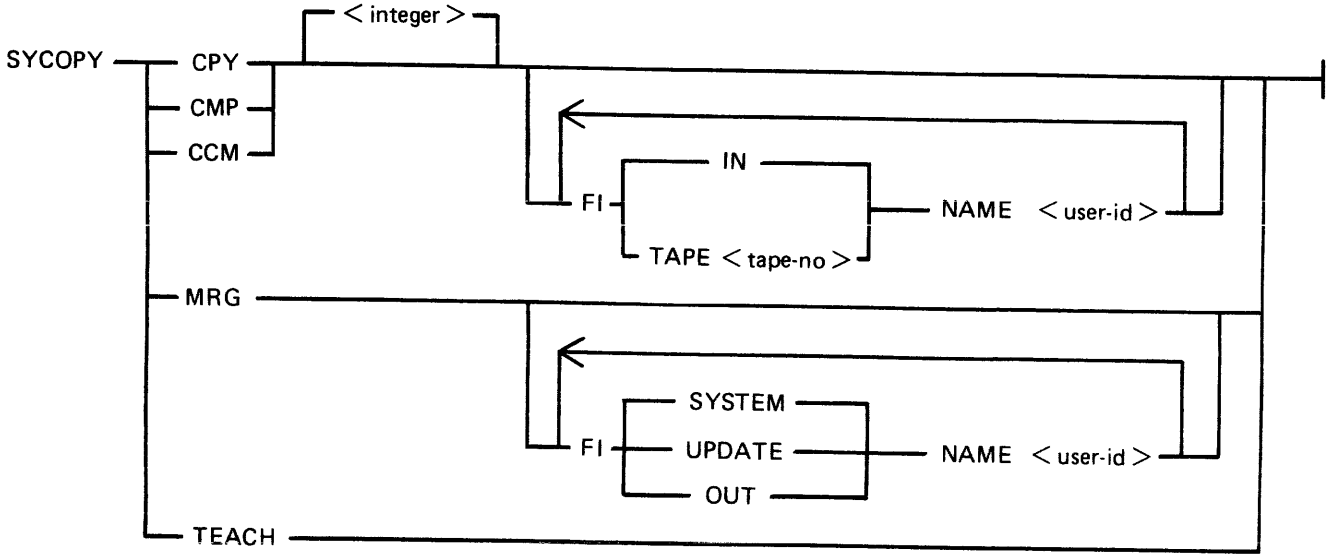
ST Intrinsic



SV Intrinsic



SYCOPY Utility



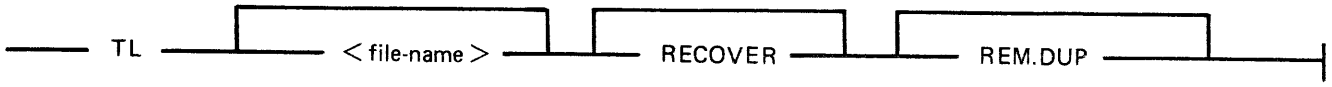
TAPELR Utility

TAPELR < library-tape-name >

TAPEPD Utility

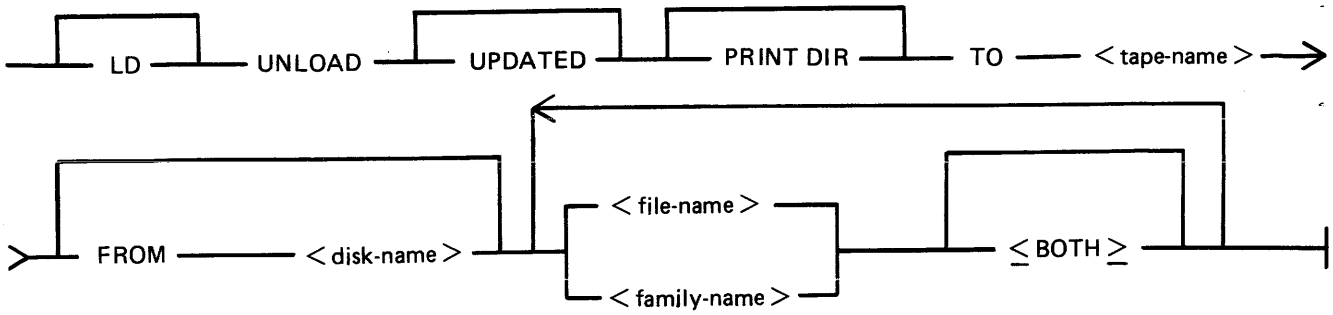
TAPEPD < library-tape-name >

TL Utility

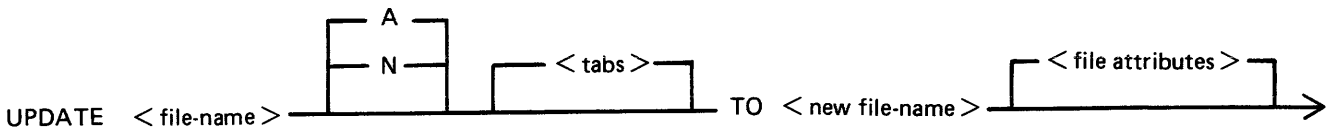


TO PRBK Intrinsic

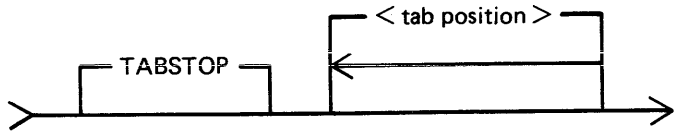
UNLOAD Utility



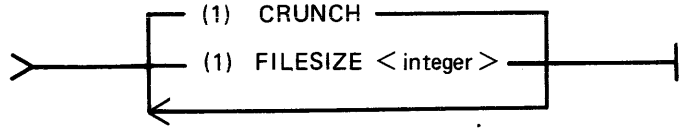
UPDATE Utility



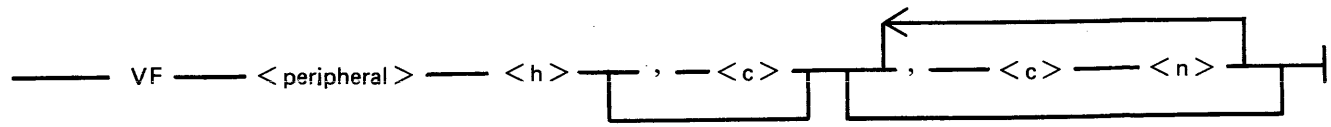
where < tabs > is



and < file attributes > is



VF Intrinsic



WL Utility

WL

XD Utility

XD < disk-name > < address > < length >

APPENDIX C
EXAMPLES OF PRINTED UTILITY OUTPUT

This appendix provides sample output from some of the CMS-common utilities described in section 4.

SPO and console input messages are underlined. Some utilities use SPO display messages for output. Output print listings use a printer if one is available, or (for a B 80 or B 800) a console file. Print files can be either labelled or unlabelled: if file is labelled, the name is printed following

?DATA

at the beginning of the listing, and

?END

at the end of the listing (for example, refer to the PL output listing). In this appendix, print files are shown boxed in: other output is on the SPO.

The meaning of the input messages are given in section 4. The utilities are given here in alphabetical order.

CHECK.DISK

```
I/O  MIX          MESSAGE          TEXT
     NO.

I    30           CHECK.DISK SOGDOC1
0     7           07/CHECK.DISK <2> SOGDOC1/SYSMEM DMA INPUT PARITY ERROR WHILE IN REA
                  D DISK ADDRESS = 001203
0     7           ERROR NOTIFIED ON READING DISK SOGDOC1
0     7           SECTOR 4611/@001203@ NOT DENOTED BAD IN DIRECTORY
0     7           CHECK.DISK ON SOGDOC1 COMPLETED -
0     7           ONE CONSISTENT ERROR NOTIFIED
0     7           1 BAD SECTOR<S> NOT DENOTED IN DIRECTORY
```

```
DIFFERENCE(S) FOUND FROM BYTES @002A@ AND @002A@

RECORD.       1 OF FILE TEST1                  COMPLETE RECORD -
@0000@ 2020 2020 2020 2020 5448 4953 2049 5320 4120 5445 5354 2052 4543 4F52 4420 494E"       THIS IS A TEST RECORD IN"
@0020@ 2046 494C 4520 5445 5354 3120 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020" FILE TEST1                    "
@0040@ 2020 2020 2020 2020 3030 3030 3031 3030                                          "       00000100"

RECORD.       1 OF FILE TEST2                  DIFFERENCE(S) -
@0020@ .... .... .... .... .... 32.. .... .... .... .... .... .... .... .... .... ...."..........2.....................
NO DIFFERENCES IN SECTION(S) NOT PRINTED
```

DA

```
I/O  MIX          MESSAGE          TEXT
     NO.

I    30           DA MYDISK PRINTER
O    15           15/DA  PRINTER BACKUP FILE SYSTEM/PB00018  OPENED
O    15           DA. MINI DISK MYDISK
O    15           15/DA  ACPT
I    30           AX 15 DCL
O    15           DA. ILLEGAL PARAMETER DCL
O    15           15/DA  ACPT
I    30           AX 15 LABEL
O    15           15/DA  ACPT
I    30           AX 15 DFH TEST1
O    15           DA. ILLEGAL PARAMETER TEST1
O    15           15/DA  ACPT
I    30           AX 15 DFH OF TEST1
O    15           15/DA  ACPT
I    30           AX 15 DFH NEXT
O    15           15/DA  ACPT
I    30           AX 15 AVAIL.TABLE AVAIL
O    15           15/DA  ACPT
I    30           AX 15 NAME.LIST NAMES
O    15           15/DA  ACPT
I    30           AX 15 READ @0140E@
O    15           15/DA  ACPT
I    30           AX 15 END
O    15           DA. TERMINATED
O    15           15/DA  PRINTER BACKUP FILE SYSTEM/PB00018  CLOSED
```

```
DA VERSION 3.04.03          ANALYSIS OF MINI DISK MYDISK                08:47:39 ON FRI  6 AUG 82  PAGE     1.

-----------------------------------------------------------------------------------------------------------

SECTOR          CARTRIDGE IDENTIFIER          MYDISK            SERIAL NUMBER                 000003
  0/@000000@    OWNERS IDENTIFICATION         JOHN

  0/@000000@    OWNERS IDENTIFICATION         JOHN

                INITIALIZATION DATE           82147             INITIALIZATION SYSTEM           BDS
                PACK CODE                     0                 ACCESS CODE
                RESTRICTED CARTRIDGE          NO                INTEGRITY FLAG                1
                BAD SECTOR COUNT              000000            ACTUAL ERROR COUNT            000000

                NUMBER OF CYLINDERS             88/@0058@       UNIT OF ALLOCATION (SECTORS)    1/@01@
                NUMBER OF TRACKS/CYLINDER        2/@02@         NUMBER OF SECTORS/TRACK        32/@20@

                NAME LIST ADDRESS               42/@002A@       NAME LIST LENGTH               10/@0A@
                AVAILABLE TABLE ADDRESS         32/@0020@       AVAILABLE TABLE LENGTH         10/@0A@
                ADDRESS OF FIRST DFH            52/@0034@       MAXIMUM NUMBER OF FILES       100/@0064@
                ADDRESS OF PPIT                  0/@000000@     PPIT LENGTH                     0/@00@

                LOGICAL UNIT NUMBER              0/@00@         CMS/WARMSTART DISK
                                                                BOOTSTRAP VERSION   03.04.03

-----------------------------------------------------------------------------------------------------------

DISK FILE HEADERS          ADDRESS:       52/@000034@   LENGTH:       100/@000064@          SPECIFIED FILE.ID

 SECTOR
  53/@000035@   FILE IDENTIFIER     TEST1              @5445535431202020202020202 0@

                PSEUDO PACK TAG     @20@               FILE TYPE         @00@ = NORMAL DATA

                CREATION DATE       82216              LAST ACCESS DATE            82217
                GENERATION NUMBER       0/@0000@       IMPLEMENTATION LEVEL NUMBER     0/@00@

                FLAGS:        FILE CRUNCHED      - 0   (BIT 0)
                              ROUGH TABLE VALID. - 0   (BIT 1)
                              DUAL PACK FILE     - 0   (BIT 2)
                              SINGLE AREA FILE   - 0   (BIT 3)
                  EXTENDABLE PSEUDO PACK FILE    - 0   (BIT 4)
                              FILE UPDATED       - 1   (BIT 9)

                RECORD SIZE            80/@0050@       RECORDS/BLOCK                   9/@0009@
                ACT. FILE SIZE          5/@000005@     SECTORS/BLOCK                   4/@0004@
                MAX. FILE SIZE          5/@000005@     SAVE FACTOR                  (   )

                MAX. AREAS IN USE       1/@01@         OVERFLOW DISK-ID
                RECS IN LAST AREA       5/@0005@       SPARE BYTES IN LAST RECORD      0/@0000@

                USER COUNTS: TOTAL NO. OF USERS              - 0  (BITS 0-2)
                             FILE OPEN WITH OTHERUSE SHARED  - 0  (BIT  3)
                             FILE HAS OUTPUT USER            - 0  (BIT  4)
                             NO. OF SHARED/LOCK ACCESS USERS - 0  (BITS 5-7)

     AREA   BIT MAP  ALLOCATION     ENTRY (ALLOCATION UNITS)   NO. OF RECORDS    ACTUAL DISK LOCATION (SECTORS)
             1 2     OF AREA         ADDRESS       SIZE         IN AREA            ADDRESS            SIZE

       1     1 0 = THIS DISK        152/@0098@    4/@0004@      5/@0005@        152/@000098@       4/@000004@
   OTHER(S)  0 0 = UNUSABLE
```

```
DA VERSION 3.04.03          ANALYSIS OF MINI DISK MYDISK                08:47:39 ON FRI  6 AUG 82  PAGE     2.
-------------------------------------------------------------------------------------------------------------

DISK FILE HEADERS          ADDRESS:       52/@000034@   LENGTH:       100/@000064@          SPECIFIED SECTOR

 SECTOR
  54/@000036@   FILE IDENTIFIER     COBOL4              @434F424F4C342020202020202@

                PSEUDO PACK TAG     @20@                FILE TYPE         @10@ = S-CODE

                CREATION DATE       82096               LAST ACCESS DATE            82217
                GENERATION NUMBER       0/@0000@        IMPLEMENTATION LEVEL NUMBER     0/@00@

                FLAGS:        FILE CRUNCHED     - 0   (BIT 0)
                              ROUGH TABLE VALID - 0   (BIT 1)
                              DUAL PACK FILE    - 0   (BIT 2)
                              SINGLE AREA FILE  - 1   (BIT 3)
                  EXTENDABLE PSEUDO PACK FILE   - 0   (BIT 4)
                              FILE UPDATED      - 1   (BIT 9)

                RECORD SIZE           180/@00B4@        RECORDS/BLOCK                   1/@0001@
                ACT. FILE SIZE        331/@00014B@      SECTORS/BLOCK                   1/@0001@
                MAX. FILE SIZE        331/@00014B@      SAVE FACTOR                 (   )

                MAX. AREAS IN USE       1/@01@          OVERFLOW DISK-ID
                RECS IN LAST AREA     331/@014B@        SPARE BYTES IN LAST RECORD      0/@0000@

                USER COUNTS: TOTAL NO. OF USERS               - 0  (BITS 0-2)
                             FILE OPEN WITH OTHERUSE SHARED   - 0  (BIT  3)
                             FILE HAS OUTPUT USER             - 0  (BIT  4)
                             NO. OF SHARED/LOCK ACCESS USERS  - 0  (BITS 5-7)

     AREA   BIT MAP  ALLOCATION     ENTRY (ALLOCATION UNITS)   NO. OF RECORDS     ACTUAL DISK LOCATION (SECTORS)
              1 2     OF AREA          ADDRESS       SIZE         IN AREA             ADDRESS            SIZE

       1     1 0 = THIS DISK        4252/@109C@   331/@014B@    331/@014B@        4252/@00109C@      331/@00014B@
   OTHER(S)  0 0 = UNUSABLE

-------------------------------------------------------------------------------------------------------------

AVAILABLE TABLE            ADDRESS:       32/@000020@   LENGTH:       10/@00000A@          AVAILABLE ENTRIES.

                      ENTRY IN TABLE (ALLOCATION UNITS)                ACTUAL DISK LOCATION (SECTORS)
 SECTOR          STATUS    LENGTH        START      END (+1)          LENGTH        START ADDRESS    END ADDRESS(+1)
  32/@000020@  1 AVAIL   2562/@0A02@  1690/@069A@  4252/@109C@     2562/@000A02@    1690/@00069A@    4252/@00109C@
  41/@000029@ 29 AVAIL   1049/@0419@  4583/@11E7@  5632/@1600@     1049/@000419@    4583/@0011E7@    5632/@001600@

-------------------------------------------------------------------------------------------------------------

NAME LIST                  ADDRESS:       42/@00002A@   LENGTH:       10/@00000A@              NAMES ENTRIES.

          SECTOR          NAME/STATUS                                PP-TAG  INDEX   DFH ADDRESS
          42/@00002A@   1 SYSMEM       @5359534D454D202020202020@    @20@    00       52/@0034@
                        2 TEST1        @5445535431202020202020202@    @20@    01       53/@0035@
                        3 COBOL4       @434F424F4C342020202020202@    @20@    02       54/@0036@
                        4 COBOL1       @434F424F4C312020202020202@    @20@    03       55/@0037@
                        5 COBOL2       @434F424F4C322020202020202@    @20@    04       56/@0038@
                        6 COBOLXREF    @434F424F4C565245462020202@    @20@    05       57/@0039@
                        7 COBOLINTX    @434F424F4C494E5458202020@    @20@    06       58/@003A@
                        8 COBOL.NOTES  @434F424F4C2E4E4F544553 20@    @20@    07       59/@003B@
```

```
DA VERSION 3.04.03          ANALYSIS OF MINI DISK MYDISK                08:47:39 ON FRI  6 AUG 82  PAGE     3.

                    9  COBOL3       @434F424F4C332020202020202@    @20@    08       60/@003C@
-------------------------------------------------------------------------------------------------------------------

SECTOR          CHARACTER: 0 1  2 3  4 5  6 7  8 9  A B  C D  E F  0 1  2 3  4 5  6 7  8 9  A B  C D  E F
 5134/@00140E@    0/@000@: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

                 32/@020@: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

                 64/@040@: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

                 96/@060@: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

                128/@080@: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF

                160/@0A0@: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

DSKUTL BOOT

I/O	MIX MO.	MESSAGE	TEXT
I	30		DSKUTL BOOT DISK2 <COLDSTART>
O	12		DSKUTL <BOOT> - COLDSTART BOOTSTRAP VERSION 03.04.01 COPIED TO DISK DISK 2

DSKUTL IN

I/O	MIX NO.	MESSAGE TEXT
I	30	DSKUTL IN
O	11	DSKUTL <IN> - ENTER DISK NAME <UP TO 7 LEGAL CHARACTERS>
O	11	11/DSKUTL ACPT
I	30	AX 11 DISK2
O	11	DSKUTL <IN> - ENTER NO. OF FILES <NUMBER LESS THAN 2805>
O	11	11/DSKUTL ACPT
I	30	AX 11 99
O	11	DSKUTL <IN> - ENTER SERIAL NO. <6 DIGITS>
O	11	11/DSKUTL ACPT
I	30	AX 11 304002
O	11	DSKUTL <IN> - ENTER OWNERS NAME <UP TO 14 CHARACTERS. DELIMIT WITH " >
O	11	11/DSKUTL ACPT
I	30	AX 11 "BURROUGHS CORP"
O	11	DSKUTL <IN> - IS THIS DISK FOR MTR USE? - <Y OR N>
O	11	11/DSKUTL ACPT
I	30	AX 11 Y
O	11	DSKUTL <IN> - ENTER REQUIRED BOOTSTRAP FILE-NAME OR NULL IF FA/CMSBOOT I S TO BE USED
O	11	11/DSKUTL ACPT
I	30	AX 11
O	11	DSKUTL <IN> - ENTER NO. OF TEST PATTERNS TO BE SPECIFIED <NUMBER 0-4>
O	11	11/DSKUTL ACPT
I	30	AX 11 1
O	11	DSKUTL <IN> - ENTER 1ST TEST PATTERN <20 CHARACTER MAX. DELIMIT WITH @ I F HEX, " IF ASCII>
O	11	11/DSKUTL ACPT
I	30	AX 11 @FF@
O	11	READY AND RESERVE REQUIRED DRIVE FOR DISK INITIALISE
O	11	11/DSKUTL <28> WAITING INITIALISE DK DEVICE ASSIGNMENT REQUIRED
I	30	RD DMA
O	30	DMA RESERVED O FILES OPEN
I	30	AD 11 DMA
O	11	DMA IS CMS DISK DISK1 - ENTER OK TO CONFIRM <IN> REQUIRED
O	11	11/DSKUTL ACPT
I	30	AX 11 OK
O	11	1ST OF 4 SURFACE CHECK PASSES COMPLETED ON DMA
O	11	2ND OF 4 SURFACE CHECK PASSES COMPLETED ON DMA
O	11	3RD OF 4 SURFACE CHECK PASSES COMPLETED ON DMA
O	11	4TH OF 4 SURFACE CHECK PASSES COMPLETED ON DMA
O	11	MTR BOOTSTRAP VERSION 03.04.00 USED ON DMA DISK DISK2
O	11	O SECTOR(S) ARE NOT AVAILABLE ON DMA DISK DISK2
O	11	DSKUTL <IN> OF DMA DISK <5632> AS DISK2 ON B90 COMPLETED
I	30	RY DMA
O	31	DMA DISK2/ O FILES OPEN

DSKUTL LIST

```
I/O  MIX          MESSAGE          TEXT
     NO.

I    30           DSKUTL LIST 0 3
0    13           PRESENT DISK FOR LISTING PLEASE
0    13           13/DSKUTL  <28>  WAITING LIST  DK  DEVICE ASSIGNMENT REQUIRED
I    30           RD DMA
0    30           DMA    RESERVED 0  FILES OPEN
I    30           AD 13 DMA
```

```
SECTOR ADDRESS   MINI DISK DISK2 ON DRIVE DMA                    THU 12 AUG 82 AT 14:15:16.8  PAGE     1.

0/@000000@ ??????????@??????????DISK2  ????   E5D6D3F1F3F0F4F0F0F240E2D3F9C9D5E3C5D9D5D34449534B322020E2F9F000
           ?????BURROUGHS CORP?????????????    0000000000425552524F554748532043 4F525000000000000000000000000000
           ???????????????@?????????@@@???    0000000000000000000000000000040E5D6D3F2F8F2F2F2F4404040C2C4E220
           ?X? ???)??? ?d???3??????????????    005802200A0000290900002000640100003300000000000F1F0F0F0F0F0F0F0F0
           ??????????????????????????????????  F0F0F0F0000000010000000000000000000000000000000000000000000000000
           ????????????????????                0000000000000000000000000000000000000000
           SECTOR ADDRESS       LENGTH         DIRECTORY
              32/@0020@          9/@0009@   :  AVAILABLE TABLE
              41/@0029@         10/@000A@   :  NAME LIST
              51/@0033@        100/@0064@   :  DISK FILE HEADERS
               0/@0000@          0/@0000@   :  PSEUDO PACK IDENTIFICATION TABLE
1/@0000001@ ????????????????????????????????   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           ????????????????????????????????   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           ????????????????????????????????   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           ????????????????????????????????   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           ????????????????????????????????   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
           ????????????????????                FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
2/@0000002@ 8?:??!????^d????1??}????@??????.   38013A000C21EA1200005E640B1D00943185197D0212110051EFF3CB198A012E
           ?????@???S?+??.??.?????S?x1X????    060412110051EFA2A253CF2B83EB2E06042E0304120E005315783158198E0312
           ?????????/??????.??1????/???????   0200050C84121C00842F01000BEEFF822E0F043100188E032FB4000300181202
           ???.??1????.??1???????SU???X????    0091822E120431FF198E032E0C0431FF198E03081204535503001858D719F101
           ????o???????V?????????.??1??????    BA110B006FE218FE01120C0056040200A2A2821CD3012E0F043100188E03120C
           ?/???/P??????.??1????               002F0100822F50000BF6FF822E120431FF198E03
```

DSKUTL RF

I/O	MIX NO.	MESSAGE TEXT
I	30	DSKUTL RF
O	10	DSKUTL <RF> - ENTER DISK NAME <UP TO 7 LEGAL CHARACTERS>
O	10	10/DSKUTL ACPT
I	30	AX 10 DISK1
O	10	DSKUTL <RF> - ENTER NO. OF FILES <NUMBER LESS THAN 2805>
O	10	10/DSKUTL ACPT
I	30	AX 10 99
O	10	DSKUTL <RF> - ENTER SERIAL NO. <6 DIGITS>
O	10	10/DSKUTL ACPT
I	30	AX 10 304001
O	10	DSKUTL <RF> - ENTER OWNERS NAME <UP TO 14 CHARACTERS. DELIMIT WITH " >
O	10	10/DSKUTL ACPT
I	30	AX 10 "BURROUGHS CORP"
O	10	DSKUTL <RF> - IS THIS DISK FOR MTR USE? - <Y OR N>
O	10	10/DSKUTL ACPT
I	30	AX 10 N
O	10	DSKUTL <RF> - IS THIS DISK FOR COLDSTART USE? - <Y OR N>
O	10	10/DSKUTL ACPT
I	30	AX 10 Y
O	10	DSKUTL <RF> - ENTER REQUIRED BOOTSTRAP FILE-NAME OR NULL IF FA/CMSBOOT I S TO BE USED
O	10	10/DSKUTL ACPT
I	30	AX 10
O	10	DSKUTL <RF> - IS DIRECTORY ON THIS DISK TO BE IGNORED? - <Y OR N>
O	10	10/DSKUTL ACPT
I	30	AX 10 Y
O	10	READY AND RESERVE REQUIRED DRIVE FOR DISK REFORMAT
O	10	10/DSKUTL <28> WAITING REFORMAT DK DEVICE ASSIGNMENT REQUIRED
I	30	RD DMA
O	30	DMA RESERVED 0 FILES OPEN
I	30	AD 10 DMA
O	10	DMA IS CMS DISK FTR - ENTER OK TO CONFIRM <RF> REQUIRED
O	10	10/DSKUTL ACPT
I	30	AX 10 OK
O	10	COLDSTART BOOTSTRAP VERSION 03.04.01 USED ON DMA DISK DISK1
O	10	0 SECTOR(S) ARE NOT AVAILABLE ON DMA DISK DISK1
O	10	DSKUTL <RF> OF DMA DISK <5632> AS DISK1 ON B90 COMPLETED
O	10	PLEASE RUN CHECK.DISK ON DMA DISK DISK1

KA

KA MYDISK/=

```
FRI  6 AUG 82 AT 09:07:13.9        USAGE MAP OF MINI DISK MYDISK   SERIAL NO. 000003  OWNER JOHN              PAGE    1.
 INITIALISED 82147 ON    BDS FOR MAXIMUM OF  100 FILES  ALLOCATION UNIT   1  ERROR COUNT 000000  BAD SECTOR COUNT 000000
   88 CYLINDERS    2 TRACKS PER CYLINDER   32 SECTORS PER TRACK       CMS/WARMSTART DISK   BOOTSTRAP VERSION 03.04.03
    AREA ADDRESS       AREA LENGTH    STATUS         FILE NAME
    ------------       -----------    ------         ---------

        0 @0000000@        32 @0000020@  ASSIGNED      *RESERVED
       32 @0000020@        10 @000000A@  ASSIGNED      *AVAIL.TABLE
       42 @000002A@        10 @000000A@  ASSIGNED      *FILE DIREC.
       52 @0000034@       100 @0000064@  ASSIGNED      *FILE HEADRS
      152 @0000098@         4 @0000004@  ASSIGNED      TEST1
      156 @000009C@       351 @000015F@  ASSIGNED      COBOL1
      507 @0001FB@         98 @0000062@  ASSIGNED      COBOL2
      605 @000025D@       223 @00000DF@  ASSIGNED      COBOLXREF
      605 @000025D@       223 @00000DF@  ASSIGNED      COBOLXREF
      828 @000033C@        64 @0000040@  ASSIGNED      COBOLINTX
      892 @000037C@       256 @0000100@  ASSIGNED      COBOL.NOTES
     1148 @000047C@       256 @0000100@  ASSIGNED      COBOL.NOTES
     1404 @000057C@       176 @00000B0@  ASSIGNED      COBOL.NOTES
     1580 @000062C@       110 @000006E@  ASSIGNED      COBOL3
     1690 @000069A@        24 @0000018@  ASSIGNED      XA
     1714 @0006B2@         50 @0000032@  ASSIGNED      AMEND
     1764 @0006E4@       2488 @0009B3@  AVAILABLE
     4252 @001090@        331 @000014B@  ASSIGNED      COBOL4
     4583 @0011E7@       1049 @0004190  AVAILABLE
```

LD & CHECKADUMP

```
I/O  MIX          MESSAGE          TEXT
     NO.

0    31           MTA  UNLABELLED
I    30           SN MTA 00001
0    31           MTA  PURGED 00001
I    30           DUMP & CHECK PRINT.DIR TO DEMO C=,MPL=,UTIL=
0    28           EX 0000000/CMSCANDE FILE02      CQ02        MX02
0     9           09/CMSCANDE  BOJ PR IS C    1317
0     8           08/LD  <10>  WAITING DEMO/FL00036  MT  NO FILE
0    31           MTA  DEMO/ 001 00001   SAVED
0    31           MTA  DEMO/ 002 00002   SAVED
I    30           RY MTA
0    31           MTA  DEMO/ 002 00002
I    30           SN MTA 00002
0    31           MTA  PURGED 00002     IN USE BY 08/LD
0    31           MTA  DEMO/ 002 00002
0    10           10/CHECKADUMP  <17>  WAITING DEMO/FILE000  AT  NO FILE
0     8           END DUMP
0    11           TRANSFER COMPLETED
0    31           MTA  DEMO/ 001 00001
0    10           10/CHECKADUMP  <17>  WAITING DEMO/FL00036  MT  NO FILE
0    31           MTA  DEMO/ 001 00001
0    31           MTA  DEMO/ 002 00002     IN USE BY 10/CHECKADUMP
0    10           NO DISCREPANCIES BETWEEN
0    10           DUMP TAPE DEMO AND DISK SYSTEM
0    31           MTA  DEMO/ 002 00002
```

```
MON  9 AUG 82 13:28:44.1          TAPE SERIAL NUMBER 00002                PAGE    1
DIRECTORY OF PE TAPE   DEMO      DUMPED ON MON  9 AUG 82 AT 13:17:59.3
```

FILE NUMBER	FILE NAME	ACTUAL SIZE	MAXIMUM SIZE	RECORD SIZE	RECS/ BLOCK	CREATED	ACCESSED	GEN. NO.	FILETYPE
1	COLDSTART	144	144	180	8	82119	82221	0	SYSTEM @15@
2	CMSBOOT	121	121	180	1	82048	82221	0	SYSTEM @14@
3	CHECK.DISK	37	37	180	1	81058	82221	0	CODE @10@
4	CHECKADUMP	44	44	180	1	82014	82221	0	CODE @10@
5	CO	164	164	180	1	82048	82221	0	CODE @10@
6	COMPARE	49	49	180	1	81166	82221	0	CODE @10@
7	CONFIGURER	55	55	180	1	81189	82221	0	CODE @10@
8	CP	25	25	180	1	81058	82221	0	CODE @10@
9	CREATE	58	58	180	1	81058	82221	0	CODE @10@
10	COBOL4	331	331	180	1	82096	82221	0	CODE @10@
11	COBOL1	351	351	180	1	82096	82221	0	CODE @10@
12	COBOL2	98	98	180	1	82096	82221	0	CODE @10@
13	COBERRS	121	121	180	1	81318	82221	0	SYSTEM @43@
14	COBOLXREF	223	223	180	1	81197	82221	0	CODE @10@
15	COBSVERTER	80	80	180	1	81327	82221	0	CODE @10@
16	CONVERSION	66	66	90	20	80323	82221	0	DATA @00@
17	COBOLINTX	64	64	180	1	81261	82221	0	SYSTEM @14@
18	CMSNDL2	350	350	180	1	81317	82221	0	CODE @10@
19	CMSNDL3	211	211	180	1	81317	82221	0	CODE @10@
20	CMSNDL1	46	46	180	1	81317	82221	0	CODE @10@
21	COBOL.NOTES	1542	1542	80	9	82174	82221	0	DATA @00@
22	CMSCANDE	225	225	180	1	79269	82221	0	CODE @10@
23	COBOL3	110	110	180	1	82096	82221	0	CODE @10@
24	CO.MASTER	27	27	180	1	82215	82221	0	SYSTEM @45@
25	CMSBASIC	492	492	180	1	82134	82221	0	CODE @10@
26	COPY	79	79	180	1	81334	82221	0	CODE @10@
27	MPL.BINDER	141	141	180	1	82076	82221	0	CODE @10@
28	MPL.2	190	190	180	1	82077	82221	0	CODE @10@
29	MPL.1	430	430	180	1	82078	82221	0	CODE @10@
30	MPL.SIZE	21	21	180	1	82081	82221	0	CODE @10@
31	MPL.ERRORS	305	305	60	3	82042	82221	0	SYSTEM @40@
32	MPL.NAMES	45	45	128	1	82076	82221	0	SYSTEM @40@
33	MPL.LISTS	34	34	180	1	81057	82221	0	CODE @10@
34	MPL.NOTES	432	432	80	9	82174	82221	0	DATA @00@
35	UTIL.NOTES1	11134	11134	80	9	82186	82221	0	DATA @00@
36	UTIL.NOTES2	7391	7391	80	9	82174	82221	0	DATA @00@

LIST MYDISK/TEST1

```
REC.NO. FILE MYDISK/TEST1 ON DSK                REC/BLK SIZES   80/  720     FRI  6 AUG 82 AT 08:55:37.1

     1  THIS IS RECORD ONE   11111111111    5448495320495320524543 4F5244204F4E4520202031313131313131313131
        1111111111111111111111111111111111  3131313131313131313131313131313131313131313131313131313131313131
        1111111100000100                    3131313131313131303030303030313030
     2  THIS IS RECORD TWO   22222222222    5448495320495320524543 4F5244205457 4F2020203232323232323232323232
        2222222222222222222222222222222222  3232323232323232323232323232323232323232323232323232323232323232
        2222222200000200                    3232323232323232303030303030323030
     3  THIS IS RECORD THREE 33333333333    5448495320495320524543 4F5244205448524545203333333333333333333333
        3333333333333333333333333333333333  3333333333333333333333333333333333333333333333333333333333333333
        3333333300000300                    3333333333333333303030303030333030
     4  THIS IS RECORD FOUR  4444444444     5448495320495320524543 4F5244204644F5552202034343434343434343434
        4444444444444444444444444444444444  3434343434343434343434343434343434343434343434343434343434343434
        4444444400000400                    3434343434343434303030303030343030
     5  THIS IS RECORD FIVE  55555555555    5448495320495320524543 4F5244204649564520203535353535353535353535
        5555555555555555555555555555555555  3535353535353535353535353535353535353535353535353535353535353535
        5555555500000500                    3535353535353535303030303030353030
```

LIST TEST1 A

```
REC.NO. FILE SYSTEM/TEST1 ON DSK                REC/BLK SIZES   80/  720     FRI  6 AUG 82 AT 11:49:24.3

     1  THIS IS RECORD ONE   11111111111111111111111111111111111111111111111100000100
     2  THIS IS RECORD TWO   22222222222222222222222222222222222222222222222200000200
     3  THIS IS RECORD THREE 3333333333 ERRROR 33333333333333333333333333333300000300
     4  THIS IS RECORD FOUR  44444444444444444444444444444444444444444444444400000400
     5  THIS IS RECORD FIVE  55555555555555555555555555555555555555555555555500000500
```

LR

LR MYDISK/= < A >

```
FRI  6 AUG 82 AT 08:58:52.4        DIRECTORY OF MINI DISK MYDISK   SERIAL NO. 000003  OWNER JOHN              PAGE    1.
 INITIALISED 82147 GN    BDS FOR MAXIMUM OF  100 FILES  ALLOCATION UNIT   1  ERROR COUNT 000000  BAD SECTOR COUNT 000000
    88 CYLINDERS    2 TRACKS PER CYLINDER   32 SECTORS PER TRACK       CMS/WARMSTART DISK   BOOTSTRAP VERSION 03.04.03
FILE          ACTUAL MAXIMUM  REC. RECS/ CREATED UPDATED  GEN.    FILE     VERSION   NO.      AREA              AREA
NAME            SIZE    SIZE  SIZE BLOCK      ACCESSED    NO.    TYPE     NO.      AREAS    LOCATIONS         SIZES

*RESERVED         32      32   180    32  82147 82218       0 SYSTEM  @20@              1        0 @0000000@    32 @0000020@
*AVAIL.TABLE      10      10   180    32  82147 82218       0 SYSTEM  @20@              1       32 @0000020@    10 @000000A@
*FILE DIREC.      10      10   180    32  82147 82218       0 SYSTEM  @20@              1       42 @000002A@    10 @000000A@
*FILE HEADRS      100     100  180    32  82147 82218       0 SYSTEM  @20@              1       52 @0000034@   100 @0000064@
AMEND             50      50   180     1  81058 82218 U     0 C810227 @10@ 03.04.00    1     1714 @0006B2@    50 @0000032@
COBOL.NOTES       1542    1542  80     9  82174 82217 U     0  DATA   @00@              3      892 @000037C@   256 @0001000@
                                                                                        1148 @000047C@   256 @0001000@
                                                                                        1404 @000057C@   176 @0000B0@
COBOL1            351     351   180     1  82096 82217 U     0 C820406 @10@ 03.04.05    1      156 @00009C@   351 @00015F@
COBOL2            98      98    180     1  82096 82217 U     0 C820406 @10@ 03.04.05    1      507 @0001FB@    98 @0000062@
COBOL3            110     110   180     1  82096 82217 U     0 C820406 @10@ 03.04.05    1     1580 @000062C@   110 @000006E@
COBOL4            331     331   180     1  82096 82217 U     0 C820406 @10@ 03.04.05    1     4252 @00109C@   331 @00014B@
COBOLINTX         64      64    180     1  81261 82217 U     0 SYSTEM  @14@              1      828 @000033C@    64 @0000040@
COBOLXREF         223     223   180     1  81197 82217 U     0 C810714 @10@ --------    1      605 @000025D@   223 @0000DF@
TEST1             5       5     80     9  82216 82218 U     0  DATA   @00@              1      152 @0000098@     4 @0000004@
XA                24      24    180     1  81058 82218 U     0 C810227 @10@ 03.04.00    1     1690 @000069A@    24 @0000018@
```

LR MYDISK/= 100

```
FRI  6 AUG 82 AT 08:59:39.3        DIRECTORY OF MINI DISK MYDISK   SERIAL NO. 000003  OWNER JOHN              PAGE    1.
 INITIALISED 82147 ON    BDS FOR MAXIMUM OF  100 FILES  ALLOCATION UNIT   1  ERROR COUNT 000000  BAD SECTOR COUNT 000000
    39 CYLINDERS    2 TRACKS PER CYLINDER   32 SECTORS PER TRACK       CMS/WARMSTART DISK   BOOTSTRAP VERSION 03.04.03
FILE          ACTUAL MAXIMUM  REC. RECS/ CREATED UPDATED  GEN.    FILE     VERSION   NO.      AREA            AREA
NAME           SIZE    SIZE  SIZE BLOCK      ACCESSED     NO.    TYPE     NO.      AREAS    LOCATIONS       SIZES

COBOL4          331     331   180     1  82096 82217 U     0 C820406 @10@ 03.04.05   1     4252 @00109C@   331 @00014B@
COBOL1          351     351   180     1  82096 82217 U     0 C820406 @10@ 03.04.05   1      156 @00009C@   351 @00015F@
COBOLXREF       223     223   180     1  81197 82217 U     0 C810714 @10@ --------   1      605 @00025D@   223 @00000F@
COBOL.NOTES    1542    1542    80     9  82174 82217 U     0  DATA   @00@            3      892 @00037C@   256 @0001000
                                                                                      1148 @00047C@   256 @0001000
                                                                                      1404 @00057C@   176 @0000B00
COBOL3          110     110   180     1  82096 82217 U     0 C820406 @10@ 03.04.05   1     1580 @00062C@   110 @00006E@


       AREA            AREA              AREA            AREA              AREA            AREA
      ADDRESSES        SIZES            ADDRESSES        SIZES            ADDRESSES        SIZES

AVAILABLE AREAS
---------------
    1764 @0006E4@    2488 @0009B9@       4583 @0011E7@    1049 @000419@

            TOTAL AVAILABLE SPACE ON DISK          3537 @000DD1@


       AREA            AREA              AREA            AREA              AREA            AREA
      ADDRESSES        SIZES            ADDRESSES        SIZES            ADDRESSES        SIZES

TEMPORARY AREAS
---------------

      TOTAL SPACE ON DISK IN TEMPORARY USE             0 @000000@
```

MODIFY CODE.FILE MYDISK/ COBOL1 PPB EOJ.SUPPRESS OFF PPB PRINT.PPB END

+ MODIFICATIONS SUCCESSFUL +

```
CMS UTILITY: MODIFY [3.04.00]

CODE.FILE MYDISK/COBOL1 PPB EOJ.SUPPRESS OFF PPB PRINT.PPB END




  PPB OF CODE.FILE MYDISK /COBOL1

IMP.LEVEL.NO                      0
PROGRAM NAME          "OBJCBL130405"
HARDWARE                      @51@ (B1800)
MCP                      "03.03.33"
PROG.VERSION               03.04.05
INTERP.PACK               "0000000"
INTERP.PACK               "0000000"
INTERP.NAME           "BILINTERP   "
COMPILER NAME         "MPL 3.3V40  "
COMPILE DATE               "820406"
EOJ.SUPPRESS                      0
CLASS                             4 (A)
INIT.MESS                     @FF@
ENTRY POINT                       0
PST.LENGTH                      180
PST.LOCATION                     70
DST.LENGTH                      450
DST.LOCATION                      2
TCB.PA LENGTH                    88
TCB.PA LOCATION                  92
STACK LENGTH                    500
CCB.PA LENGTH                     0
CCB.PA LOCATION                   0
TCB.PE LENGTH                     0
IFNB LENGTH                     510
IFNB LOCATION                    22
```

PL

PL SYS-HLD 16300

```
MON 11 JUN 84  AT  19:13:57          CMS LOG FILE PRINTOUT OF SYS-HLD16300 FROM B90 SYSTEM                PAGE   1
                                     STATUS OF FILE IS TRANSFERRED

  TIME     DATE   I/O MIX ENTRY   ODT                                  MESSAGE          TEXT
HH:MM:SS MM/DD/YY     NO. NO.   ID/TAG


00:00:53 11/11/11  0   27     1 FF/FFFF LOGGING IS INITIATED ON 11/11/11  AT 00:00:47 (MCP VERSION 03.05.08)
00:00:54 11/11/11  0   27     2 FF/FFFF SO PRBK ON
00:00:54 11/11/11  0   30     3 FF/FFFF SO PRBK ON COMPLETE
00:00:54 11/11/11  0   30     4 FF/FFFF PRINTER BACKUP OPTION IS ON. PB DISK IS SYS/  DEFAULT BACKUP MAXIMUM
                                  FILESIZE IS 10000
03:35:34 11/11/11  I   30     5 01/001D CTL ON
03:35:34 11/11/11  0   30     6 01/001D  ************************** * COMPUTER  MANAGEMENT  * * SYSTEM
                                  (CMS)  83328  * * MCP VERSION 03.05.08  * **************************
03:35:35 11/11/11  0   30     7 01/001D "CTL ON" COMPLETE - SYSTEM SPO STATUS ACQUIRED
03:35:35 11/11/11  0   30     8 01/001D IF ACCURATE DATE/TIME IS REQUIRED, UPDATE USING "DT"
03:35:56 11/11/11  I   30     9 01/001D DT 06/10/84 1722
17:22:00 06/10/84  0   30    10 01/001D 10 JUN 84 84162  SUN  1722
17:24:07 06/10/84  0    1    11 01/001D EX 0000000/CMSCANDE FILE01      CQ01        MX01
17:29:59 06/10/84  I   30    12 01/001D OL LP
17:29:59 06/10/84  0   30    13 01/001D LPA READY
17:30:00 06/10/84  0   30    14 01/001D LPB NOT READY
17:30:00 06/10/84  0   30    15 01/001D END OL
17:32:35 06/10/84  0    3    16 01/001D CANDE TERMINATING.
10:04:10 06/11/84  0   31    17 01/001D LPB READY
10:04:31 06/11/84  I   30    18 01/001D OL
10:04:31 06/11/84  0   30    19 01/001D DFA SYS/         SYS DISK 17  FILES OPEN
10:04:31 06/11/84  0   30    20 01/001D DMA NOT READY
10:04:31 06/11/84  0   30    21 01/001D MTA NOT READY
10:04:31 06/11/84  0   30    22 01/001D MTB NOT READY
10:04:31 06/11/84  0   30    23 01/001D MTC NOT READY
10:04:31 06/11/84  0   30    24 01/001D MTD NOT READY
10:04:31 06/11/84  0   30    25 01/001D RTA
10:04:31 06/11/84  0   30    26 01/001D DCA
10:04:31 06/11/84  0   30    27 01/001D LPA READY
10:04:31 06/11/84  0   30    28 01/001D LPB READY
10:04:31 06/11/84  0   30    29 01/001D END OL
10:04:38 06/11/84  I   30    30 01/001D SAVE LPB
10:04:44 06/11/84  0   30    31 01/001D LPB READY        SAVED
10:04:44 06/11/84  0   30    32 01/001D LPA READY
10:04:45 06/11/84  0    4    33 01/001D LPA READY FOR USE
10:04:45 06/11/84  0    4    34 01/001D END SAVE
10:14:30 06/11/84  0    1    35 01/001D EX 0000000/CMSCANDE FILE01      CQ01        MX02
10:15:19 06/11/84  I   30    36 01/001D DLR CMSC=
10:15:22 06/11/84  0    6    37 01/001D Searching for CMSC=        on SYS
10:15:22 06/11/84  0    6    38 01/001D   File Id      Act    Max   Rec  Block Type Area      Address      V
                                  ersion
10:15:24 06/11/84  0    6    39 01/001D CMSCANDE       1103   1103   180    1  @10@   1  237472 @039FA0@  03
                                  .05.04
10:15:32 06/11/84  0    6    40 01/001D END OF JOB
10:16:01 06/11/84  0    5    41 01/001D CANDE TERMINATING.
10:16:28 06/11/84  I   30    42 01/001D COPY  CMSCANDE TO CMSCANDE30S
10:17:02 06/11/84  0    1    43 01/001D EX 0000000/FILEUTL FILE01      CQ01        MX03
10:17:21 06/11/84  I   30    44 01/001D MX
10:17:22 06/11/84  0   30    45 01/001D NMBR  PROG-NAME  PR  STATUS
10:17:23 06/11/84  0   30    46 01/001D 01/MCSOBJ        C  EXECUTING
10:17:23 06/11/84  0   30    47 01/001D 07/COPY          B  SHORT WAITED ON I/O
10:17:23 06/11/84  0   30    48 01/001D 08/FILEUTL       B  EXECUTING
10:17:24 06/11/84  0   30    49 01/001D 29/NDLSYS        C  SHORT WAITED ON I/O
10:17:24 06/11/84  0   30    50 01/001D END MX
10:17:52 06/11/84  0    1    51 01/001D EX 0000000/FILEUTL FILE01      CQ01        MX04
```

2032801

```
MON 11 JUN 84  AT  19:13:57          CMS LOG FILE PRINTOUT OF SYS-MLD16300 FROM B90 SYSTEM                PAGE   2
                                     STATUS OF FILE IS TRANSFERRED

  TIME     DATE   I/O MIX ENTRY   ODT                                  MESSAGE          TEXT
HH:MM:SS MM/DD/YY     NO. NO.   ID/TAG


10:17:57 06/11/84  O    7    52 01/001D CMSCANDE COPIED TO CMSCANDE305
10:17:57 06/11/84  O    7    53 01/001D END COPY
10:18:18 06/11/84  I   30    54 01/001D FT CMSCANDE305 00
10:18:21 06/11/84  O   10    55 01/001D FILETYPE OF         CMSCANDE305  CHANGED FROM $10$ TO $00$
10:20:26 06/11/84  O    1    56 01/001E EX 0000000/CMSCANDE FILE02      CQ02        MX05
10:23:40 06/11/84  I   30    57 01/001D MX
10:23:40 06/11/84  O   30    58 01/001D NMBR  PROG-NAME  PR  STATUS
10:23:40 06/11/84  O   30    59 01/001D 01/MCSOBJ        C  SUSPENDED WAITING ON I/O
10:23:40 06/11/84  O   30    60 01/001D 09/FILEUTL       B  EXECUTING
10:23:40 06/11/84  O   30    61 01/001D 11/CMSCANDE      C  SWAPPED OUT, WAITING ON I/O
10:23:40 06/11/84  O   30    62 01/001D 29/NDLSYS        C  SHORT WAITED ON I/O
10:23:40 06/11/84  O   30    63 01/001D END MX
10:24:50 06/11/84  I   30    64 01/001D LIST CMSCANDE 317 1
10:24:58 06/11/84  O   12    65 01/001D END LIST
10:25:21 06/11/84  O   31    66 01/001D LPA READY
10:25:32 06/11/84  O   31    67 01/001D LPA READY
10:25:42 06/11/84  O   31    68 01/001D LPA READY
10:26:32 06/11/84  I   30    69 01/001D LIST CMSCANDE305 315 10
10:26:50 06/11/84  O   13    70 01/001D END LIST
10:26:57 06/11/84  O   31    71 01/001D LPA READY
10:33:25 06/11/84  I   30    72 01/001D DC GT TD830XB
10:33:30 06/11/84  O    1    73 01/001D FROM MCS:  GT OK (126)
10:34:04 06/11/84  O   11    74 01/001E CANDE TERMINATING.
10:34:20 06/11/84  O    1    75 01/001E EX 0000000/CMSCANDE FILE02      CQ02        MX06
10:37:24 06/11/84  I   30    76 01/001D DC NT
10:37:25 06/11/84  O    1    77 01/001D FROM MCS:  SYNTAX ERROR - PLEASE RE-ENTER (100)
10:37:32 06/11/84  I   30    78 01/001D DC NT TD830XB
10:37:35 06/11/84  O    1    79 01/001D FROM MCS:  NT OK (126)
10:39:34 06/11/84  O   14    80 01/001E CANDE TERMINATING.
10:40:04 06/11/84  I   30    81 01/001D DC GT TD830XB TEXT 0
10:40:09 06/11/84  O    1    82 01/001D FROM MCS:  GT OK (126)
10:40:20 06/11/84  O    1    83 01/001E EX 0000000/CMSCANDE FILE02      CQ02        MX07
10:41:05 06/11/84  I   30    84 01/001D DC NT TD830XB
10:41:14 06/11/84  O    1    85 01/001D FROM MCS:  NT OK (126)
10:41:26 06/11/84  O   16    86 01/001D SECONDARY LOG FILE SYS-MLG16300 CREATED
10:41:27 06/11/84  O   16    87 01/001D SECONDARY LOG FILE SYS-MLD16300 CREATED
10:41:27 06/11/84  O   16    88 01/001D LOG INFORMATION TRANSFERRED
10:41:27 06/11/84  O   16    89 01/001D END TL
10:49:44 06/11/84  I   30    90 01/001D FT CMSCANDE305 10
10:49:48 06/11/84  O   17    91 01/001D FILETYPE OF         CMSCANDE305  CHANGED FROM $00$ TO $10$
10:50:00 06/11/84  I   30    92 01/001D DC SO TMCS
10:50:02 06/11/84  O    1    93 01/001D FROM MCS:  TMCS OPTIONS SET (1074)
10:50:15 06/11/84  O    1    94 01/001D EX 0000000/CMSCANDE305 FILE01      CQ01        MX08
10:50:21 06/11/84  O   18    95 01/001D 18/CMSCANDE305  PROGRAM ERROR <105>  SEGMENT BOUNDARY VIOLATION AT
                                CODE SEGMENT 66   OFFSET 957  PROC/DEBUG LINE 1
10:50:23 06/11/84  O   18    96 01/001D 18/CMSCANDE305  MUST BE TERMINATED    ENTER :  "DS 18/CMSCANDE305 "
                                - TO TERMINATE OR  "DP 18/CMSCANDE305 " - TO TERMINATE WITH DUMP
10:50:50 06/11/84  I   30    97 01/001D DS 18/CMSCANDE
10:50:51 06/11/84  O   30    98 01/001D DS 18/CMSCANDE INVALID - PROGRAM NAME  DOES NOT MATCH MIX NUMBER   E
                                NTER "MX" FOR MORE DETAILS
10:51:09 06/11/84  I   30    99 01/001D DS 18/CMSCANDE305
10:51:10 06/11/84  O   18   100 01/001D 18/CMSCANDE305  TERMINATED (DS'ED)
10:51:30 06/11/84  O    1   101 01/001D EX 0000000/FILEUTL FILE01      CQ01        MX09
10:52:09 06/11/84  I   30   102 01/001D FT CMSCANDE305 00
10:52:13 06/11/84  O   20   103 01/001D FILETYPE OF         CMSCANDE305  CHANGED FROM $10$ TO $00$
```

```
  TIME     DATE   I/O MIX ENTRY   ODT                                    MESSAGE          TEXT
HH:MM:SS MM/DD/YY     NO. NO.   ID/TAG


10:55:26 06/11/84  I   30   104 01/001D FT CMSCANDE305 10
10:55:30 06/11/84  O   21   105 01/001D FILETYPE OF         CMSCANDE305  CHANGED FROM $00$ TO $10$
10:55:54 06/11/84  O   15   106 01/001E CANDE TERMINATING.
10:56:09 06/11/84  O    1   107 01/001E EX 0000000/CMSCANDE305 FILE02      CQ02        MX10
10:56:49 06/11/84  O   22   108 01/001E CANDE TERMINATING.
10:57:18 06/11/84  I   30   109 01/001E FT CMSCANDE305 00
10:57:22 06/11/84  O   23   110 01/001E FILETYPE OF         CMSCANDE305  CHANGED FROM $10$ TO $00$
11:00:40 06/11/84  I   30   111 01/001D FT CMSCANDE305 00
11:00:43 06/11/84  O   24   112 01/001D FILETYPE OF         CMSCANDE305  CHANGED FROM $00$ TO $00$
11:00:58 06/11/84  O    1   113 01/001D EX 0000000/DC.AMEND FILE01      CQ01        MX11
11:01:01 06/11/84  O   25   114 01/001D 25/DC.AMEND  COMMENCING 1101
11:03:49 06/11/84  O    1   115 01/001E EX 0000000/CMSCANDE305 FILE02      CQ02        MX12
11:03:50 06/11/84  O   30   116 01/001E 30/LOADER.SCL  PROGRAM ERROR <166>   INVALID SIGN CODE AT CODE SEGME
                                NT 0   OFFSET 63776  DEBUG LINE 8192
11:04:19 06/11/84  O    1   117 01/001E EX 0000000/CMSCANDE305 FILE02      CQ02        MX13
11:04:21 06/11/84  O   30   118 01/001E 30/LOADER.SCL  PROGRAM ERROR <166>   INVALID SIGN CODE AT CODE SEGME
                                NT 0   OFFSET 63776  DEBUG LINE 8192
11:04:47 06/11/84  I   30   119 01/001E RM CMSCANDE305
11:04:49 06/11/84  O   27   120 01/001E CMSCANDE305 REMOVED
11:04:49 06/11/84  O   27   121 01/001E END RM
11:05:06 06/11/84  O   25   122 01/001D 25/DC.AMEND  COMPLETED  1105
11:05:29 06/11/84  O    1   123 01/001D EX 0000000/CMSCANDE FILE01      CQ01        MX14
11:05:35 06/11/84  I   30   124 01/001D DC RO TMCS
11:05:37 06/11/84  O    1   125 01/001D FROM MCS:  TMCS OPTIONS RESET (1075)
18:04:20 06/11/84  O   31   126 01/001D MTA CBL357/  001 <30508>
18:17:20 06/11/84  I   30   127 01/001D PD CMSLIV=
18:17:28 06/11/84  O   27   128 01/001D CMSLIV ON SYS CONTAINS -
18:17:28 06/11/84  O   27   129 01/001D CMSLIVMISRC
18:17:28 06/11/84  O   27   130 01/001D END PD
18:17:59 06/11/84  I   30   131 01/001D CH CMSLIVMISRC TO LIVM.I.SRC
18:18:01 06/11/84  O   27   132 01/001D CMSLIVMISRC CHANGED TO LIVM.I.SRC
18:18:01 06/11/84  O   27   133 01/001D END CH
18:18:18 06/11/84  I   30   134 01/001D PDL =SPEC=
18:18:22 06/11/84  O    2   135 01/001D =SPEC= CONTAINS :
18:18:25 06/11/84  O    2   136 01/001D INSPECTS     INFSPECSQQ   INFSPECS
18:18:28 06/11/84  O    2   137 01/001D INSPECT      CBLS.SPECP8K BILS.SPECP8K
18:18:30 06/11/84  O    2   138 01/001D SCLSPEC.C38K
18:18:30 06/11/84  O    2   139 01/001D END PD
18:19:10 06/11/84  I   30   140 01/001D CH BILS.SPECP8K TO SPECP8K.BILS
18:19:12 06/11/84  O   27   141 01/001D BILS.SPECP8K CHANGED TO SPECP8K.BILS
18:19:12 06/11/84  O   27   142 01/001D END CH
18:19:32 06/11/84  I   30   143 01/001D CH CBLS.SPECP8K TO SPECP8K.CBLS

18:19:34 06/11/84  O   27   144 01/001D CBLS.SPECP8K CHANGED TO SPECP8K.CBLS
18:19:34 06/11/84  O   27   145 01/001D END CH
18:20:17 06/11/84  I   30   146 01/001D CH SCLSPEC.C38K TO SPECP8K.SCL
18:20:19 06/11/84  O   27   147 01/001D SCLSPEC.C38K CHANGED TO SPECP8K.SCL
18:20:19 06/11/84  O   27   148 01/001D END CH
18:20:48 06/11/84  I   30   149 01/001D PD B90.D=
18:20:55 06/11/84  O   27   150 01/001D B90.D ON SYS CONTAINS -
18:20:55 06/11/84  O   27   151 01/001D B90.DIAGS
18:20:55 06/11/84  O   27   152 01/001D END PD
18:22:40 06/11/84  I   30   153 01/001D DUMP & CHECK TO SPECDOC LIVM=,SPECP8K=,B90.DIAGS=
18:23:22 06/11/84  O    3   154 01/001D 03/LD  PROGRAM SUSPENSION <17>   ASSISTANCE REQUIRED FOR TAPE FILE S
                                PECDOC/FILE000 001   TAPE NOT ONLINE
```

```
  TIME     DATE   I/O MIX ENTRY   ODT                                  MESSAGE          TEXT
HH:MM:SS MM/DD/YY     NO. NO.   ID/TAG


18:23:30 06/11/84  I   30   155 01/001D OL MT
18:23:30 06/11/84  O   30   156 01/001D MTA C8L357/  001 <30508>
18:23:30 06/11/84  O   30   157 01/001D MTB NOT READY
18:23:30 06/11/84  O   30   158 01/001D MTC NOT READY
18:23:30 06/11/84  O   30   159 01/001D MTD NOT READY
18:23:30 06/11/84  O   30   160 01/001D END OL
18:23:45 06/11/84  I   30   161 01/001D SN MTA 11JUN
18:23:46 06/11/84  O    4   162 01/001D LOG INFORMATION TRANSFERRED
18:23:46 06/11/84  O    4   163 01/001D END TL
18:23:52 06/11/84  O   31   164 01/001D MTA PURGED       <11JUN>
18:24:04 06/11/84  O    3   165 01/001D LIVM.I.SRC DUMPED
18:24:10 06/11/84  O    3   166 01/001D SPECPBK.C8LS DUMPED
18:24:18 06/11/84  O    3   167 01/001D SPECPBK.BILS DUMPED
18:24:27 06/11/84  O    3   168 01/001D SPECPBK.SCL DUMPED
18:24:33 06/11/84  O    3   169 01/001D B90.DIAGS DUMPED
18:24:39 06/11/84  O    5   170 01/001D 05/CHECKADUMP  PROGRAM SUSPENSION <17>  ASSISTANCE REQUIRED FOR TAP
                                        E FILE SPECDOC/FILE000 001   TAPE NOT ONLINE
18:24:39 06/11/84  O    3   171 01/001D END DUMP
18:24:58 06/11/84  O   31   172 01/001D MTA SPECDOC/ 001 <11JUN>
18:25:44 06/11/84  I   30   173 01/001D MX
18:25:45 06/11/84  O   30   174 01/001D NMBR  PROG-NAME  PR  STATUS
18:25:45 06/11/84  O   30   175 01/001D 01/MCS08J        C  SWAPPED OUT, WAITING ON I/O
18:25:45 06/11/84  O   30   176 01/001D 05/CHECKADUMP     B  EXECUTING
18:25:45 06/11/84  O   30   177 01/001D 26/CMSCANDE       C  SWAPPED OUT, WAITING ON I/O
18:25:45 06/11/84  O   30   178 01/001D 29/MDLSYS         C  EXECUTING
18:25:46 06/11/84  O   30   179 01/001D END MX
18:26:30 06/11/84  O    5   180 01/001D NO DISCREPANCIES BETWEEN DUMP TAPE SPECDOC AND DISK SYS
18:26:30 06/11/84  O    5   181 01/001D END CHECKADUMP
18:26:50 06/11/84  O   31   182 01/001D MTA SPECDOC/ 001 <11JUN>
18:31:36 06/11/84  I   30   183 01/001D RM SPECPBK,LIVM=
18:31:37 06/11/84  O   27   184 01/001D SPECPBK NOT REMOVED
18:31:37 06/11/84  O   27   185 01/001D - FILE NOT FOUND
18:31:43 06/11/84  O   27   186 01/001D LIVM.I.SRC REMOVED
18:31:46 06/11/84  O   27   187 01/001D END RM
18:33:12 06/11/84  I   30   188 01/001D DLR MPL=
18:33:17 06/11/84  O    6   189 01/001D Searching for MPL=        on SYS
18:33:17 06/11/84  O    6   190 01/001D   File Id      Act    Max   Rec  Block Type Area      Address      V
                                        ersion
18:33:19 06/11/84  O    6   191 01/001D MPL.1           518    518   180    1  @10@   1  198712 @0308380 03
                                        .05.05
18:33:21 06/11/84  O    6   192 01/001D MPL.2           226    226   180    1  @10@   1  310432 @04BCA0@ 03
                                        .05.05
18:33:23 06/11/84  O    6   193 01/001D MPL.BINDER      154    154   180    1  @10@   1  116944 @01C8D0@ 03
                                        .05.05
18:33:25 06/11/84  O    6   194 01/001D MPL.NAMES        61     61   128    1  @40@   1  428656 @068A70@
                                        . .
18:33:27 06/11/84  O    6   195 01/001D MPL.LISTS        34     34   180    1  @10@   1   59040 @00E6A0@ 03
                                        .05.00
18:33:29 06/11/84  O    6   196 01/001D MPL.SIZE         24     24   180    1  @10@   1  310664 @04BD88@ 03
                                        .05.05
18:33:31 06/11/84  O    6   197 01/001D MPL.ERRORS      305    305    60    3  @40@   1  117104 @01C970@
                                        . .
18:33:39 06/11/84  O    6   198 01/001D END OF JOB
18:54:37 06/11/84  I   30   199 01/001D SAVE LPA
18:54:44 06/11/84  O   30   200 01/001D LPA READY        SAVED
```

```
MON 11 JUN 84  AT  19:13:57          CMS LOG FILE PRINTOUT OF SYS-MLD16300 FROM B90 SYSTEM                PAGE   5
                                     STATUS OF FILE IS TRANSFERRED

  TIME     DATE   I/O MIX ENTRY   ODT                                    MESSAGE          TEXT
HH:MM:SS MM/DD/YY     NO. NO.   ID/TAG


18:54:44 06/11/84  O   30   201 01/001D LPB READY
18:54:48 06/11/84  O    7   202 01/001D LPB READY FOR USE
18:54:48 06/11/84  O    7   203 01/001D END SAVE
18:55:07 06/11/84  O   31   204 01/001D LPB READY
18:56:08 06/11/84  O   31   205 01/001D MTA GOODIES/ 001 < TOMS>
18:56:26 06/11/84  I   30   206 01/001D TAPEPD GOODIES
18:57:01 06/11/84  I   30   207 01/001D MX
18:57:02 06/11/84  O   30   208 01/001D NMBR  PROG-NAME  PR  STATUS
18:57:02 06/11/84  O   30   209 01/001D 01/MCSOBJ        C  SWAPPED OUT, WAITING ON I/O
18:57:02 06/11/84  O   30   210 01/001D 08/TAPEPD        B  EXECUTING
18:57:02 06/11/84  O   30   211 01/001D 26/CMSCANDE      C  EXECUTING
18:57:02 06/11/84  O   30   212 01/001D 29/NDLSYS        C  EXECUTING
18:57:02 06/11/84  O   30   213 01/001D END MX
18:57:52 06/11/84  I   30   214 01/001D QL LP
18:57:52 06/11/84  O   30   215 01/001D LPA READY        SAVED
18:57:52 06/11/84  O   30   216 01/001D LPB CBL.MAN      IN USE BY 26/CMSCANDE
18:57:53 06/11/84  O   30   217 01/001D END QL
18:57:58 06/11/84  I   30   218 01/001D SAVE LPA
19:00:30 06/11/84  O   30   219 01/001D LPA READY        SAVED
19:00:30 06/11/84  O   30   220 01/001D LPB CBL.MAN      IN USE BY 26/CMSCANDE
19:00:39 06/11/84  O    9   221 01/001D LPB READY FOR USE
19:00:47 06/11/84  O    9   222 01/001D END SAVE
19:00:49 06/11/84  O    8   223 01/001D PE TAPE GOODIES < TOMS> DUMPED ON SUN 20 MAY 84  AT 14:19:59.0 CONTA
                                       INS -
19:00:51 06/11/84  O    8   224 01/001D DOC.EDIT DOC.LIST NDL.ANALYZER
19:00:52 06/11/84  O    8   225 01/001D SYENDLSRC CANDETEACH CMSCANDE
19:00:53 06/11/84  O    8   226 01/001D SYCOM SYCOMLOAD SYETCLSRC
19:00:54 06/11/84  O    8   227 01/001D PDL DCLR DCCP
19:00:55 06/11/84  O    8   228 01/001D EDITOR DC.AMEND NDLCO
19:00:56 06/11/84  O    8   229 01/001D LEARN VDUTEACH DLR
19:00:58 06/11/84  O    8   230 01/001D CMSTEACH DLRS SLIST
19:00:59 06/11/84  O    8   231 01/001D TUTOR TUT-BASIC NDL.LINE
19:01:02 06/11/84  O    8   232 01/001D NDL.LINES FT XLTNDLSRC
19:01:03 06/11/84  O    8   233 01/001D PATCHIT FF.S STRIP.DISK
19:01:03 06/11/84  O    8   234 01/001D BF FF STRIP.DISK-S
19:01:04 06/11/84  O    8   235 01/001D ENDOC.EDIT.S ENDOC.EDIT ENDOCMAN
19:01:06 06/11/84  O    8   236 01/001D ENDOC.MANUAL ENDOC.LIST ENDOC.LIST.S
19:01:06 06/11/84  O    8   237 01/001D END TAPEPD
19:01:10 06/11/84  O   31   238 01/001D MTA GOODIES/ 001 < TOMS>
19:01:25 06/11/84  O   10   239 01/001D LOG INFORMATION TRANSFERRED
19:01:25 06/11/84  O   10   240 01/001D END TL
19:01:29 06/11/84  I   30   241 01/001D BF
19:01:32 06/11/84  O   11   242 01/001D BF DISPLAYS BACKUP FILE NAMES.   SYNTAX:
19:01:32 06/11/84  O   11   243 01/001D    !<----------    ;  ----------------!
19:01:32 06/11/84  O   11   244 01/001D    !                 !<-- , ---------!!
19:01:32 06/11/84  O   11   245 01/001D BF ---------------------- <FID> ------>
19:01:32 06/11/84  O   11   246 01/001D     !- FROM -<MFID>-! !- *<FID>* ---!
19:01:32 06/11/84  O   11   247 01/001D     !- ON  -!         !- = ---------!
19:01:32 06/11/84  O   11   248 01/001D                       !-  ALL ------!
19:01:32 06/11/84  O   11   249 01/001D                       !-  <INT> ----!
19:01:32 06/11/84  O   11   250 01/001D                       !- <I1>-<I2> -!
19:01:32 06/11/84  O   11   251 01/001D
19:01:32 06/11/84  O   11   252 01/001D  >---------------------------------->
19:01:32 06/11/84  O   11   253 01/001D     !-- DETAIL ---------------!
19:01:32 06/11/84  O   11   254 01/001D                  !-- PRINT ----!
```

MON 11 JUN 84 AT 19:13:57 CMS LOG FILE PRINTOUT OF SYS-MLD16300 FROM B90 SYSTEM PAGE 6
STATUS OF FILE IS TRANSFERRED

TIME HH:MM:SS	DATE MM/DD/YY	I/O	MIX NO.	ENTRY NO.	ODT ID/TAG	MESSAGE TEXT				
19:01:32	06/11/84	O	11	255	01/001D					
19:01:32	06/11/84	O	11	256	01/001D	>----------->!				
19:01:32	06/11/84	O	11	257	01/001D	!- BEFORE <MM>/<DD>/<YY> -!				
19:01:44	06/11/84	I	30	258	01/001D	BF ALL DETAIL				
19:01:50	06/11/84	O	12	259	01/001D	SUPERNOTE.PB	PB00010	SUPERSP	3132	05/26/83 @ 11:45 DOC.LIST
19:01:53	06/11/84	O	12	260	01/001D	PB00002	PB00002	GENTESL	71	06/08/84 @ 13:38 COBOL4
19:01:53	06/11/84	O	12	261	01/001D	PB00003	PB00003	GENTESL	71	06/08/84 @ 13:41 COBOL4
19:01:54	06/11/84	O	12	262	01/001D	PB00005	PB00005	GENTESL	70	06/08/84 @ 14:03 COBOL4
19:01:54	06/11/84	O	12	263	01/001D	PB00004	PB00004	GENTESL	78	06/08/84 @ 13:46 COBOL4
19:01:55	06/11/84	O	12	264	01/001D	PB00006	PB00006	GENTESL	70	06/08/84 @ 14:12 COBOL4
19:01:56	06/11/84	O	12	265	01/001D	B1000MCP.D60	PB00086	MLFILE1	374	05/09/84 @ 13:27 DOC.LIST
19:01:56	06/11/84	O	12	266	01/001D	DCCH.DSS	PB00364	MLFIL		

PL SYS-MLG 16000

MON 11 JUN 84 AT 19:14:21 CMS LOG FILE PRINTOUT OF SYS-MLG16000 FROM B90 SYSTEM PAGE 1
STATUS OF FILE IS TRANSFERRED

SYSTEM CONFIGURATION :

PROCESSOR NO.	CHANNEL NO.	SUB-CHANNEL NO.	DEVICE NO.	DEVICE	HARDWARE DEVICE KIND	TOTAL I/O COUNT LOGGED (EXCLUDING RETRIES)
1	3		1	DFA	FIXED DISK (211)	61116
1	2		1	DMA	BURROUGHS STANDARD MINI-DISK	0
1	7		1	MTA	PE REEL	1162
1	7		2	MTB	PE REEL	0
1	7		3	MTC	PE REEL	0
1	7		4	MTD	PE REEL	0
1	4		1	RTA	REAL TIME CLOCK	0
1	5		1	DCA	DATA COMM	0
1	1		1	LPA	LINE PRINTER	0
1	6		1	LPB	LINE PRINTER	0

MAINTENANCE ENTRY ON 06/08/84 AT 11:06:03

MIX	PROCESSOR/ CHANNEL/ SUB-CHANNEL/ DEVICE NO./ DEVICE	INPUT/ OUTPUT	OPCODE	TYPE OF OPERATION	MFID	SERIAL NUMBER	NO. OF RETRIES	SUCCESS?	BUFFER POINTER	BUFFER LENGTH (BYTES)	SECTOR START ADDRESS
202	01/07/ /01/MTA	INPUT	40	READ	GMC600	84072	6	NO	@001C20@	7200	

ERROR TYPE:- PARITY
STATUS :- 0307 1C01
DESCRIPTOR:- C1E3 BA9F FD00 0014 0020 1C00 0940 0802 0000 0601 0500 0000 0000 00

MAINTENANCE ENTRY ON 06/08/84 AT 11:07:09

MIX	PROCESSOR/ CHANNEL/ SUB-CHANNEL/ DEVICE NO./ DEVICE	INPUT/ OUTPUT	OPCODE	TYPE OF OPERATION	MFID	SERIAL NUMBER	NO. OF RETRIES	SUCCESS?	BUFFER POINTER	BUFFER LENGTH (BYTES)	SECTOR START ADDRESS
0	01/07/ /01/MTA	INPUT	40	READ	GMC600	84072	6	NO	@001C20@	7200	

ERROR TYPE:- PARITY
STATUS :- 0307 1C01
DESCRIPTOR:- C1E3 BA9F FD00 0014 0020 1C00 0B40 0800 0000 0601 0500 0000 0000 00

MAINTENANCE ENTRY ON 06/08/84 AT 11:10:21

MIX	PROCESSOR/ CHANNEL/ SUB-CHANNEL/ DEVICE NO./ DEVICE	INPUT/ OUTPUT	OPCODE	TYPE OF OPERATION	MFID	SERIAL NUMBER	NO. OF RETRIES	SUCCESS?	BUFFER POINTER	BUFFER LENGTH (BYTES)	SECTOR START ADDRESS
143	01/07/ /01/MTA	INPUT	40	READ	GMC600	84072	1	YES	@001C20@	7200	

STATUS :- 0306 1C05
DESCRIPTOR:- C1E3 BEAB 3406 0014 0020 1C00 0C40 0802 0000 0101 0500 0000 0000 00

MAINTENANCE ENTRY ON 06/08/84 AT 11:10:28

MIX	PROCESSOR/ CHANNEL/ SUB-CHANNEL/ DEVICE NO./ DEVICE	INPUT/ OUTPUT	OPCODE	TYPE OF OPERATION	MFID	SERIAL NUMBER	NO. OF RETRIES	SUCCESS?	BUFFER POINTER	BUFFER LENGTH (BYTES)	SECTOR START ADDRESS
143	01/07/ /01/MTA	INPUT	40	READ	GMC600	84072	6	NO	@001C20@	7200	

ERROR TYPE:- PARITY
STATUS :- 0306 1C05
DESCRIPTOR:- C1E3 BAAB 3406 0014 0020 1C00 0C40 0802 0000 0601 0500 0000 0000 00

SUMMARY BETWEEN ENTRY DATED 11:06:03 ON 06/08/84
AND ENTRY DATED 11:10:28 ON 06/08/84

PROCESSOR/ CHANNEL/ SUB-CHANNEL/ DEVICE NO./ DEVICE	TOTAL I/O COUNT LOGGED IN FILE	NO. INPUT MAINTENANCE ENTRIES	NO. OF INPUT RETRIES	NUMBER OF SUCCESSES	NUMBER OF FAILURES	NO. OUTPUT MAINTENANCE ENTRIES	NO. OF OUTPUT RETRIES	NUMBER OF SUCCESSES	NUMBER OF FAILURES
01/03/ /01/DFA	61116	0	0	0	0	0	0	0	0
01/02/ /01/DMA	0	0	0	0	0	0	0	0	0
01/07/ /01/MTA	1162	4	19	1	3	0	0	0	0
01/07/ /02/MTB	0	0	0	0	0	0	0	0	0
01/07/ /03/MTC	0	0	0	0	0	0	0	0	0
01/07/ /04/MTD	0	0	0	0	0	0	0	0	0
01/04/ /01/RTA	0	0	0	0	0	0	0	0	0
01/05/ /01/DCA	0	0	0	0	0	0	0	0	0
01/01/ /01/LPA	0	0	0	0	0	0	0	0	0
01/06/ /01/LPB	0	0	0	0	0	0	0	0	0

SQ MYDISK VERIFY FAST 400 LIST

*** SQUASH VERSION 3.4.02 (03/05/82) ***
*** VALIDATION PHASE BEGINS ***
LARGEST AVAILABLE SPACE IS 2488 SECTORS
TOTAL AVAILABLE SPACE IS 3537 SECTORS IN 2 AREA(S)
*** SQ COMPLETED ***

USAGE MAP OF DISK MYDISK BEFORE SQUASH DATE : FRI 6 AUG 82 PAGE 1.

FILE NAME	AREA NB	FILE OPEN	DISK ALLOCATION LENGTH	FROM	TO	AVAILABLE TABLE INFORMATION	REMARKS
			32	@000000@	@00001F@	*RESERVED	
			10	@000020@	@000029@	*AVAIL.TABLE	
			10	@00002A@	@000033@	*FILE DIRECTORY	
			100	@000034@	@000097@	*FILE HEADERS	
TEST1	1		4	@000098@	@00009B@		
COBOL1	1		351	@00009C@	@0001FA@		
COBOL2	1		98	@0001FB@	@00025C@		
COBOLXREF	1		223	@00025D@	@00033B@		
COBOLINTX	1		64	@00033C@	@00037B@		
COBOL.NOTES	1		256	@00037C@	@00047B@		
COBOL.NOTES	2		256	@00047C@	@00057B@		
COBOL.NOTES	3		176	@00057C@	@00062B@		
COBOL3	1		110	@00062C@	@000699@		
XA	1		24	@00069A@	@0006B1@		
AMEND	1		50	@0006B2@	@0006E3@		
			2488	@0006E4@	@00109B@	AVAILABLE SPACE	
COBOL4	1		331	@00109C@	@0011E6@		
			1049	@0011E7@	@0015FF@	AVAILABLE SPACE	

NOTE : NO ERRORS DISCOVERED IN VALIDATION PHASE - USAGE MAP OF THE DISK ENDS

TAPELR DEMO

MON 9 AUG 82 AT 11:28:55.7 TAPE SERIAL NUMBER 12345 PAGE 1.
DIRECTORY OF PE TAPE DEMO DUMPED ON MON 9 AUG 82 AT 10:49:35.8

FILE NUMBER	FILE NAME	ACTUAL SIZE	MAXIMUM SIZE	RECORD SIZE	RECS/ BLOCK	CREATED	ACCESSED	GEN. NO.	FILETYPE
1	COLDSTART	144	144	180	8	82119	82221	0	SYSTEM @15@
2	CMSBOOT	121	121	180	1	82048	82221	0	SYSTEM @14@
3	CHECK.DISK	37	37	180	1	81058	82221	0	CODE @10@
4	CHECKADUMP	44	44	180	1	82014	82221	0	CODE @10@
5	CO	164	164	180	1	82048	82221	0	CODE @10@
6	COMPARE	49	49	180	1	81166	82221	0	CODE @10@
7	CONFIGURER	55	55	180	1	81189	82221	0	CODE @10@
8	CP	25	25	180	1	81058	82221	0	CODE @10@
9	CREATE	58	58	180	1	81058	82221	0	CODE @10@
10	COBOL4	331	331	180	1	82096	82221	0	CODE @10@
11	COBOL1	351	351	180	1	82096	82221	0	CODE @10@
12	COBOL2	98	98	180	1	82096	82221	0	CODE @10@
13	COBERRS	121	121	180	1	81318	82221	0	SYSTEM @43@
14	COBOLXREF	223	223	180	1	81197	82221	0	CODE @10@
15	COBSVERTER	80	80	180	1	81327	82221	0	CODE @10@
16	CONVERSION	66	66	90	20	80323	82221	0	DATA @00@
17	COBOLINTX	64	64	180	1	81261	82221	0	SYSTEM @14@
18	CMSNDL2	350	350	180	1	81317	82221	0	CODE @10@
19	CMSNDL3	211	211	180	1	81317	82221	0	CODE @10@
20	CMSNDL1	46	46	180	1	81317	82221	0	CODE @10@
21	COBOL.NOTES	1542	1542	80	9	82174	82221	0	DATA @00@
22	CMSCANDE	225	225	180	1	79269	82221	0	CODE @10@
23	COBOL3	110	110	180	1	82096	82221	0	CODE @10@
24	CO.MASTER	27	27	180	1	82215	82221	0	SYSTEM @45@
25	CMSBASIC	492	492	180	1	82134	82221	0	CODE @10@
26	COPY	79	79	180	1	81334	82221	0	CODE @10@
27	MPL.BINDER	141	141	180	1	82076	82221	0	CODE @10@
28	MPL.2	190	190	180	1	82077	82221	0	CODE @10@
29	MPL.1	430	430	180	1	82078	82221	0	CODE @10@
30	MPL.SIZE	21	21	180	1	82081	82221	0	CODE @10@
31	MPL.ERRORS	305	305	60	3	82042	82221	0	SYSTEM @40@
32	MPL.NAMES	45	45	128	1	82076	82221	0	SYSTEM @40@
33	MPL.LISTS	34	34	180	1	81057	82221	0	CODE @10@
34	MPL.NOTES	432	432	80	9	82174	82221	0	DATA @00@
35	UPDATE	64	64	180	1	81327	82215	0	CODE @10@
36	UTIL.NOTES1	11134	11134	80	9	82186	82215	0	DATA @00@
37	UTIL.NOTES2	7391	7391	80	9	82174	82215	0	DATA @00@

TAPEPD ARTAPE

```
   NRZI TAPE ARTAPE <00000> DUMPED ON THU 21
... JUN 79 CONTAINS -
   MYFILE MYFILEQQ A999
   END TAPEPD
```

APPENDIX D
RELATED DOCUMENTATION

The following manuals provide information concerning CMS System Software:

Manual	Form Number
CMS ARCS Reference Manual	2012713
CMS COBOL Reference Manual Relative to Release 3.05	2033007
CMS MCP Reference Manual	2007555
CMS RPG Reference Manual Relative to Release 3.05	2033130
CMS MPLII Reference Manual Relative to Release 3.05	2033023
CMS NDL Reference Manual	1090925
CMS Data Communications Subsystem Reference Manual	1090909
CMS BASIC Reference Manual	1155819

Documentation Evaluation Form

Title: B 90/B 900/CP 9500/B 1800/B 1900 CMS
Software Operation Guide Relative to Release 3.05

Form No: 2032801
Date: March 1985

Burroughs Machines Ltd is interested in receiving your comments and suggestions regarding this manual. Comments will be utilized 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:

Corporate Product Information
Burroughs Machines Limited
Cumbernauld, Glasgow G68 0BN
Scotland