

Burroughs

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

**SYSTEM SOFTWARE
OPERATION GUIDE**

This Manual Replaces All Previous Editions of Form 2015228

PRICED ITEM

Burroughs

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

**SYSTEM SOFTWARE
OPERATION GUIDE**

This Manual Replaces All Previous Editions of Form 2015228

COPYRIGHT © 1982, BURROUGHS MACHINES LIMITED, Hounslow, England
COPYRIGHT © 1982, BURROUGHS CORPORATION, Detroit, Michigan, 48232

PRICED ITEM

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

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

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

Correspondence regarding this publication should be forwarded using the Remarks form at the back of the manual, or may be addressed directly to TIO Europe Documentation, Burroughs Machines Limited, Cumbernauld, Glasgow, Scotland G68 0LN.

LIST OF EFFECTIVE PAGES

Page	Issue
iii	Original
iv	Blank
v thru xix	Original
xx	Blank
1-1 thru 1-4	Original
2-1 thru 2-13	Original
2-14	Blank
3-1 thru 3-35	Original
3-36	Blank
4-1 thru 4-336	Original
5-1 thru 5-17	Original
5-18	Blank
6-1 thru 6-31	Original
6-32	Blank
7-1	Original
7-2	Blank
8-1 thru 8-117	Original
8-118	Blank
9-1 thru 9-102	Original
10-1 thru 10-54	Original
A-1 thru A-10	Original
B-1 thru B-27	Original
B-28	Blank
C-1 thru C-4	Original
D-1	Original
D-2	Blank

This revision incorporates the following PCNs :

001	November 1980
002	January 1981
003	October 1981

TABLE OF CONTENTS

Section	Title	Page	
1	INTRODUCTION	1-1	
	THE CMS CONCEPT	1-1	
	SOFTWARE RELEASE LEVELS	1-3	
	Software Patches	1-3	
	SOFTWARE SUPPORT	1-3	
	TO THE READER	1-3	
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	2-7	
	PRINTER FILES	2-7	
	OTHER PERIPHERALS	2-8	
	PROGRAMS	2-8	
	Executing Programs	2-8	
	INTRINSICS	2-9	
	MIX NUMBERS	2-9	
	OUTPUT MESSAGES	2-10	
	FORMAT DIAGRAMS	2-11	
	RAILROAD DIAGRAMS	2-12	
	3	CMS-COMMON INTRINSICS	3-1
		INTRODUCTION	3-1
		AD (ASSIGN PERIPHERAL DEVICE)	3-2
		AX (ACCEPT A MESSAGE FOR A PROGRAM)	3-3
		CL (CLEAR PERIPHERAL)	3-4
DB (DIRECT TO BACKUP)		3-5	
DC (DATA COMMUNICATIONS OPERATOR INPUT)		3-7	
DP (DISCONTINUE AND DUMP)		3-8	
DS (DISCONTINUE PROGRAM)		3-9	
DT (SYSTEMS DATA AND TIME)		3-10	
FD (FORM DEFINE)		3-12	
GO (RESTART A STOPPED PROGRAM)		3-13	
LG (LOG GO)		3-14	
LS (LOG STOP)		3-15	
MX (DISPLAY CURRENT MIX)		3-16	
OL (REQUEST FOR STATUS INFORMATION OF PERIPHERALS)		3-20	
PG (PURGE TAPE)		3-22	
PO (POWER OFF A DISK)		3-23	

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	PR (ASSIGN PROGRAM PRIORITY)	3-24
	RL (RELABEL DISK/PACK)	3-25
	Disk Name	3-25
	Pseudo-Packid And Packid	3-25
	RY (READY A PERIPHERAL)	3-27
	SF (SUBSTITUTE DISK FILE)	3-28
	SN (ASSIGN A SERIAL NUMBER TO A TAPE)	3-29
	SO (SET PRINTER BACKUP OPTION ON/OFF)	3-30
	ST (TEMPORARILY SUSPEND A RUNNING PROGRAM)	3-32
	SV (SAVE PERIPHERAL)	3-33
	TO (TEST PRINTER BACKUP OPTION)	3-34
	VF (VERTICAL FORMAT ON PRINTER)	3-35
4	CMS-COMMON UTILITIES	
	INTRODUCTION	4-1
	STAR FILES	4-1
	SYS-SUPERUTL	4-3
	SYSLANGUAGE	4-4
	PSEUDO-PACKS	4-5
	LOGGING IN THE 3.04 RELEASE	4-6
	Types of Log File	4-6
	Types of Logging	4-7
	Utilities Linked with Logging	4-8
	Suggested Logging Method	4-8
	Log File Sizes	4-9
	ARCHIVING USING UPDATE FACILITY	4-10
	COMMON UTILITY OUTPUT MESSAGES	4-12
	ADD (ADD FROM LIBRARY TAPE TO DISK)	4-14
	(LD-TAPE UTILITY)	4-14
	AMEND (Disk File Amending)	4-17
	Starting Byte for Modification	4-19
	CH (Change File Name)	4-22
	CHECKADUMP (Compare Library Tape with Disk)	4-24
	CHECK.DISK (Check all Sectors of a Disk)	4-27
	COMPARE (Compare Files)	4-30
	Additional Capabilities	4-33
	COPY (File Copy)	4-37
	Copying Keyfiles	4-37
	Additional Capabilities	4-38
	CP (Compute)	4-47
	CREATE (Create Disk File)	4-49
	DA (Disk Analysis)	4-54
	Analysis of Standard CMS Disks	4-54
	Analysis of Non-CMS Standard Disks	4-57
	General Notes	4-59
	Families	4-59
	Pseudo Disk Tags	4-59
	Abbreviations	4-60
	DSKUTL	4-63
	Bootstrap Table	4-64

TABLE OF CONTENTS (Cont.)

Section	Title	Page
4	DSKUTL – < COLDSTART> Option B 90 Only	4-65
	Format 1: IN	4-66
	Format 2: RF	4-71
	Format 3: LIST	4-77
	Format 4: COPY	4-78
	Format 5: BOOT	4-82
	DSKUTL Reformat and Initialize Parameters – via Accepts	4-84
	DUMP (Dump Files from Disk to a Library Tape) (LD – TAPE LIBRARY UTILITY).	4-87 4-87
	DUMPADISK (Disk Dump).	4-91
	Format 1: PRINT.DIR.	4-94
	Format 2: DUMP or UNLOAD	4-94
	Format 3: ADD or LOAD	4-95
	Error Recovery	4-95
	ECMA.LD (Load/Dump of ECMA Tape Files).	4-100
	Basic Initiation	4-100
	Compact Initiation	4-101
	FILEUTL (FILE MAINTENANCE UTILITY).	4-104
	Screen Formats and Functions.	4-104
	FL (Display File Attributes on Self-Scan)	4-123
	FS (File Squash)	4-126
	ICMD	4-128
	IR (Initiate Log Recall)	4-132
	KA (Analyze Disk Space Assignment).	4-133
	KEY.CHECK	4-134
	KX (Disk Allocation Information)	4-140
	LB (Look Back in Log)	4-142
	LD – TAPE LIBRARY UTILITY (ADD, LOAD, DUMP, UNLOAD)	4-143
	“UPDATED” Bit Options	4-143
	LF (Look Forward in Log)	4-145
	LIST (File List)	4-146
	Additional Capabilities.	4-147
	LOAD (LD – TAPE LIBRARY UTILITY)	4-152
	LOAD.VFU (Load Vertical Format Unit)	4-156
	Limitations	4-157
	LR (List Directory)	4-159
	MODIFY	4-163
	Interactive Mode	4-164
	File Attributes	4-164
	PB (List Printer Backup Files)	4-169
	PB Initiating Message Parameters	4-169
	DATAComm PB	4-171
	PB Messages	4-173
	Files Requiring Special Forms	4-173
	PD (Print Disk Directory)	4-178
	PL (Print Log Files)	4-179
	PPID (PSEUDO PACK IDENTIFIER DISPLAY).	4-185
	RB (Remove Printer Backup Files)	4-186
	RM (Remove Files from Disk)	4-188

TABLE OF CONTENTS (Cont.)

Section	Title	Page
4	SCR (System Confidence Routine)	4-190
	Execution Details	4-206
	SQ (Squash Disk)	4-215
	General Guidelines	4-220
	SYCOPY (Copy Library Tapes)	4-222
	CPY	4-222
	CMP	4-222
	CCM MRG	4-223
	TEACH	4-224
	TAPELR (List Library Tape Directory)	4-226
	TAPEPD (Print Name of a Library Tape)	4-228
	TL (Transfer Logs)	4-229
	TRANSLATOR	4-232
	Limitations	4-232
	Operating Instructions	4-232
	Initiation and Working Mode	4-233
	Functions	4-233
	System Parameters	4-240
	Escape Codes	4-240
	Utility Message Usage – SYSLANGUAGE Version 03.04.05	4-243
	Messages vs Utilities	4-243
	Utility Message Usage – SYSLANGUAGE Version 03.04.05	4-283
	Utilities vs Messages	4-283
	UNLOAD (LD – TAPE LIBRARY UTILITY)	4-326
	UPDATE (Disk File Update)	4-329
	WL (What Log File)	4-333
	XD (Delete Bad Disk Sectors)	4-333
	Warnings	4-333
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
	FUNCTIONAL DESCRIPTION	5-8
	Regular Record Sort	5-8
	Inplace Record Sort	5-9
	Keyfile Creation	5-9
	Tagfile Creation	5-9
	Merge	5-11
	Details Of Sort Keys	5-11
	ALPHA (Or UA)	5-11
	NUMERIC (Or UN)	5-11
	SA	5-12
	SN	5-14
	SSA	5-14

TABLE OF CONTENTS

Section	Title	Page
	Deleted Records	5-14
	Output Messages	5-14
	0-99	5-15
	170-200	5-15
6	COMPILATION FACILITIES	
	INTRODUCTION	6-1
	DETAILS OF "CO.MASTER" FOR RELEASE 3.4	6-1
	TO INITIATE A SINGLE COMPILATION	6-3
	General Parameters	6-3
	disk-id-1	6-3
	disk-id-2	6-3
	program-id	6-3
	disk-id-3	6-3
	compiler-name	6-4
	compiler-option	6-4
	print-option	6-5
	File Modification Parameters	6-6
	PATCHFILE	6-6
	SOURCEIN	6-7
	SOURCEOUT	6-7
	WORKFILE	6-8
	PRINTOUT	6-8
	CONTROLFILE	6-9
	ICFILE	6-9
	Descriptions of the Attributes	6-10
	TO INTERROGATE THE STATUS OF COMPILATIONS	6-14
	TO RESTART AN ABORTED COMPILATION	6-17
	TO CLEAR AN ABORTED COMPILATION	6-19
	USE OF MACRO CALLS	6-21
	COMPILER DOLLAR OPTIONS FOR CO	6-22
	RPG Compiler	6-22
	MPLII Compiler	6-22
	Stand-Alone OPTLIST	6-22
	Stand-Alone COBOLXREF	6-22
	COBOL Compiler.	6-22
	Dollar Option Precedence	6-23
	RPG Compiler	6-23
	MPLII Compiler	6-23
	COBOLXREF	6-23
	CO ERROR MESSAGES	6-24
	Error Number	6-24
	CO OTHER MESSAGES	6-27
	ZIP FAILURES	6-28
	RESTARTING EXECUTING CO VERSIONS	6-29
	LIST OF RETURN VALUES FOR CO	6-30
	RESERVED WORDS	6-31
7	NUMBERED SYSTEM SOFTWARE OUTPUT MESSAGES	
	TO BE SUPPLIED	7-1

TABLE OF CONTENTS (Cont.)

Section	Title	Page
8	B 90 DEPENDENT SYSTEM SOFTWARE	8-1
	INTRODUCTION	8-1
	POWER ON	8-1
	CMS BOOTSTRAP MODE	8-5
	A Note On 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	8-8
	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	8-14
	A Note On Dual Pack Files	8-15
	CH (CHANGE DISK FILE NAME)	8-16
	Output Messages	8-16
	CI (INITIALIZE A DISK WITH COLDSTART BOOTSTRAP)	8-18
	Output Messages	8-18
	CLEAN (CLEAN BSM DRIVE READ/WRITE HEADS)	8-20
	Output Messages	8-20
	COMPARE (COMPARE TWO DISK FILES)	8-21
	Output Messages	8-21
	COPY (COPY FILES DISK TO DISK)	8-23
	Output Messages	8-23
	Dual Pack Files	8-25
	DISCOPY (DUPLICATE A BSMII DISK)	8-26
	Output Messages	8-26
	FE (INITIALIZE MTR DISK)	8-27
	Output Messages	8-27
	IN (INITIALIZE A DISK WITH WARMSTART BOOTSTRAP)	8-29
	Output Messages	8-29
	LD (LOAD FILES FROM CASSETTE TO DISK)	8-31
	Output Messages	8-31
	LS (LIST FILE SIZES)	8-33
	Output Messages	8-33
	OL (PRINT STATUS OF DRIVES)	8-34
	Output Messages	8-34
	PDX (PRINT DISK DIRECTORIES)	8-35
	Output Message	8-35
	PO (POWER OFF)	8-36
	RL (RELABEL A DISK)	8-37

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Output Messages	8-37
	Warning	8-37
	RM (REMOVE DISK FILES)	8-38
	Output Messages	8-38
	Dual Pack Files	8-38
	WS (WARM START)	8-40
	Operating Procedure	8-40
	LOADING THE MASTER CONTROL PROGRAM (MCP)	8-41
	Console And SELF-SCAN SPO Device	8-41
	Terminal SPO Device	8-42
	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
	Hardware Error Conditions	8-50
	COLDSTART BOOTSTRAP ALGORITHM	8-52
	WARMSTART SCANNING ALGORITHM	8-53
	MEMORY DUMP TO MAGNETIC TAPE	8-54
	MEMORY DUMP TO DISK	8-55
	SYSTEM LOAD ERRORS	8-56
	Diagnosis Of Disk Errors At System Load Time	8-57
	Caelus Device Primary Status	8-57
	Caelus Disk Secondary Status	8-57
	SDI Host Controller Status	8-57
	SDI Device Status	8-58
	Magnetic Tape Device	8-58
	ERRORS UNDER MCP CONTROL	8-59
	B 90 DEPENDENT UTILITIES	8-63
	BUILDTRANS (BUILD TRANSLATION TABLE)	8-64
	Limitations	8-64
	Operating Instructions	8-64
	Mode 0	8-64
	Mode 1	8-65
	Mode 2 (Not Implemented On B 90)	8-65
	Shift Codes	8-65
	Workfile Creation	8-65
	Initiation And Working Mode	8-65
	Functions	8-66
	GET	8-66
	MAKE	8-66
	INCLUDE	8-66
	LIST	8-66
	BUILD	8-67

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	EDIT	8-67
	DELETE	8-68
	LET	8-68
	SAVE	8-68
	REMOVE	8-68
	RECOVER	8-68
	BYE	8-68
	BUILDTRANS Message Format	8-69
	SYSTRANS Format	8-70
	CONFIGURER (CONFIGURE B 90 SYSTEM)	8-72
	Operating Instructions	8-72
B 90	DUMPANALYSE	8-79
	Obtaining The Dumpfile	8-79
	Analysis Common To All Program Languages	8-80
	Program Parameter Block Information	8-81
	Data Segment And Unformatted Dump	8-83
	COBOL Dependent Dump Analysis	8-83
	Dump-Related Dollar Options	8-84
	COBOL Segment Allocation	8-85
	COBOL S-Interpreter Work Area	8-85
	Control Stack	8-85
	COP Table (Current Operand Table)	8-85
	RPG Dependent Dump Analysis	8-86
	Dump-Related Source Constructs	8-86
	Object Program Organization	8-86
	Further Information	8-87
	MPL Dependent Dump Analysis	8-87
	MPL S-Registers	8-87
	Formatted Code And Data Stack	8-88
	Volatile Registers	8-88
	Descriptor Analysis	8-88
	Relation To MPL Compilation Listings	8-89
	Location Of Program Failure Point	8-90
	Description Of Namefile For Use In Source Related Dump Analysis	8-90
	Output Messages	8-91
	GEN.DUMPFL (Create Empty B 90 Memory Dump File)	8-93
	Output Messages	8-93
	INSTALL.	8-94
	Install Environment Error Messages	8-95
	Install Copy Error Messages	8-96
	PATCHMAKER (Patching B 90 Machine-Code Object Program Files)	8-97
	Patching Procedure	8-98
	Operating Instructions	8-99
	Patchmaker Error Messages	8-101
	PMB90 (Analyze B 90 Memory Dump)	8-104
	Starting the Utility	8-104
	Using the Utility	8-105
B 90	DEPENDENT INTRINSICS	8-111
	GT (General Trace)	8-112

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Format of Diagnostic Printout	8-114
	LT (Load Translation Table)	8-115
	LT Error Messages	8-115
	ND (New Density)	8-116
	Output Messages	8-117
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	9-2
	SYSTEM STARTUP	9-4
	Physical Power-On	9-5
	WARMSTART	9-5
	System with ODT	9-5
	Warmstart Variations	9-6
	Coping with Problems	9-6
	COLDSTART	9-7
	Coldstart Initiation	9-7
	?DS	9-9
	FE	9-9
	HE (Help)	9-9
	IN (Initializing Disks)	9-10
	Initializing Fixed Disks	9-11
	LD (Load)	9-12
	OL (On Line)	9-13
	PT (Patch System Files)	9-14
	Patching Restrictions	9-15
	RF (Reformat)	9-16
	RL (Relabel)	9-18
	RP (Replace)	9-19
	WS (Warmstart)	9-20
	Coldstart Operator Unattended Mode	9-21
	Initialize, Load, Warmstart	9-22
	Patch, Warmstart	9-22
	Replace, Warmstart	9-23
	POWER OFF	9-23
	TAKING SYSTEM DUMPS	9-24
	Operating System Dumps Created Automatically	9-24
	Using GT MD	9-25
	Taking a ROM-Created Dump	9-25
	Analyzing Dumps.	9-27
	TAKING PROGRAM DUMPS	9-27
	FUNCTIONAL DESCRIPTION OF SYSTEM STARTUP	9-28
	B 900/CP 9500 Bootstraps	9-29
	WARMSTART FUNCTIONAL DESCRIPTION	9-34
	Default Configuration	9-34
	Buffer Memory Defaults	9-35

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Task Processor Defaults	9-35
	Data Comm Processor Defaults	9-35
COLDSTART	FUNCTIONAL DESCRIPTION	9-40
	FE (Initialize MTR Disk)	9-41
	Increasing an Existing Fixed Disk Assemblage	9-42
	Coldstart Disk Selection	9-42
	Coldstart Default Assignments	9-42
	Coldstart Messages	9-43
DUMPS	FUNCTIONAL DESCRIPTION	9-48
	Read-Only Memory (ROM) Dump Routine	9-48
	Disk Format	9-48
	ROM Dump Messages	9-49
	ROM Dump Display Reference Table	9-51
SPECIAL	CONSIDERATIONS	9-52
	Pseudo-Pack Concepts	9-52
	Data Communications	9-53
	Tape Concepts	9-54
	Tape Commands	9-54
	Disk Pack Concepts	9-54
	Disk Pack Used to Store System Software	9-54
	Testing Recording Surface of Disk Devices	9-55
	Backup Operating System and Disk Processors	9-57
UTILITIES	AND INTRINSICS	9-58
	ANALYZER (User Program Dump Analyzer)	9-58
	AP (Assign Pseudo-Packs)	9-60
	BINARY.COPY (Duplicate Disk Media)	9-62
	Special Considerations	9-63
	CONFIGURER (System Configuration Utility)	9-64
	Initiation	9-68
	LIST Mode	9-70
	MAKE Mode	9-70
	FIX Mode	9-70
	LOGINFO Area	9-72
	MISCINFO Area	9-72
	B 900INFO	9-73
	OSINFO	9-73
	PERIPHINFO	9-74
	DCINFO	9-75
	B 900CONFIG	9-76
	OS and Data Comm Buffer Memory Validation	9-78
	Processor Configuration	9-78
	DUMFILE.UTL	9-80
	File Partitioning	9-80
	File Reconstruction	9-81
	EX (Execute)	9-83
	FPP (Field Patch Program)	9-84
	Entering the Patch	9-85

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Format of Patch Line	9-86
	GT CF (Get System Configuration)	9-88
	GT MD (Get Full Memory Dump)	9-89
	GT MX (Get Mix) (Display Mix with Task Processor Assignment)	9-90
	PO (Power Off)	9-91
	RL (Relabel)	9-92
	RLD (Release Level Display)	9-93
	ROMANANALYZER (ROM Dump File Analyzer)	9-94
	ROMCONVERT (ROM Dump File Converter)	9-97
	SYSANALYZER (System Dump Analyzer).	9-99
	SYSANALYZER Error Messages	9-102
10	B 1800/B 1900 DEPENDENT SYSTEM SOFTWARE	
	CMS UTILITIES	10-1
	CONFIGURER	10-1
	Syntax	10-1
	Defaults	10-4
	Update Messages	10-5
	Error Messages	10-5
	Warning Messages.	10-6
	DC.INFO	10-7
	Syntax	10-7
	Error Messages	10-7
	Completion Message	10-7
	DP.ANALYZER	10-8
	How To Take A DCP Dump.	10-8
	How To Analyze A DCP Dump	10-8
	Syntax	10-8
	Commands	10-9
	DISKDUMP	10-14
	DSKDSK	10-14
	DSKMTP	10-15
	MTPDSK	10-15
	AND COMPARE	10-15
	Unlabelled Open	10-15
	Program Initiation	10-16
	BOJ Messages	10-16
	Error Handling And Error Messages	10-16
	Comparison Errors	10-16
	Recoverable Input Errors.	10-17
	Bad Sectors Errors	10-17
	Fatal Errors	10-17
	Program Warnings	10-17
	End Of Job Messages	10-18
	Warnings	10-18
	Approximate Duration of the Process	10-18
	DP.ANALYZER	10-19
	LT (Load Line Printer Train)	10-22
	MEM.ANALYZER	10-24
	NPC1.B 1000	10-27

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Introduction	10-27
	Use of the NPC1.B 1000	10-27
	Syntax	10-27
NPC34.B 1000		10-28
	Introduction	10-28
	Use of the NPC34.B 1000	10-28
	Syntax	10-28
PATCH.MAKER		10-29
	Error Messages Displayed	10-29
	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
	Operating Instructions For Patching the Interpreters or NPC34.B 1000	10-32
RETRIEVE		10-33
	Syntax	10-33
	Messages	10-33
STAND.ALONE UTILITIES		10-35
	Creation Of Cassettes	10-35
	Initiation of the Stand-Alone Utilities	10-36
	B 1000 System	10-36
	B 1825/30 System	10-36
COLD.START		10-37
	Operation	10-37
	Disk Pack and Cartridge Units	10-37
	Magnetic Tape Units	10-37
	Error Messages	10-38
CLEAR.START		10-40
	Function	10-40
	Selection of the MCP File	10-40
	Channel Selection	10-40
	Channel Identification	10-40
DISK INITIALIZERS (CART.INIT/PACK.INIT)		10-42
	General Information	10-42
	Error Limits	10-42
	Operating Instructions	10-42
	Card Format	10-43
	SPO Input	10-43
	Time Needed	10-45
	Error Messages	10-45
	End Of Job Messages	10-45
MEMORY.DUMP		10-46
	MEM.DUMP Cassette	10-46
	How To Trace A System Loop	10-46
	List of Possible Halts (in L Register)	10-46
	Known Problem	10-47
SYSTEM HALTS DOCUMENTATION		10-48
	System Halts	10-48
	CLEARSTART Halts	10-48
	MCP Halts	10-48
	System Dependent Fetch Values	10-50

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Physical I/O	10-51
	Logical I/O	10-52
	Data Communication Errors	10-53
Appendix		
A	COMPLETE INPUT MESSAGE DIAGRAMS	A-1
B	EXAMPLES OF PRINTED UTILITY OUTPUT	B-1
C	GLOSSARY OF TECHNICAL TERMS	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	2-11
2-6	Railroad Diagram Sample 1	2-12
2-7	Railroad Diagram Sample 2	2-13
2-8	Railroad Diagram Sample 3	2-13
4-1	Fixed Disk Directory Structure	4-5
4-2	Railroad Chart for COMPARE Utility	4-33
4-3	Railroad Chart for Copy Utility (Sheet 1 of 2)	4-39
4-3	Railroad Chart for Copy Utilities (Sheet 2 of 2)	4-40
4-4	Routing Diagram for FILEUTL	4-106
4-5	Railroad Chart for LIST Utility	4-148
4-6	Paper Vertical Format Tape	4-156
4-7	OTHER.IO Sample Printer Output	4-214
5-1	Regular Record Sort	5-8
5-2	Keyfile Creation	5-10
5-3	Tagfile Creation	5-10
5-4	File Merge	5-12
5-5	Multiple Key Sort	5-13
6-1	Operation of CO Utility	6-2
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	Operator Display Terminal (ODT)	9-3
9-3	Block Diagram of Operating Steps in Daily Running of B 900/CP 9500	9-4
9-4	Switch Settings for Unattended Operation	9-21
9-5	Summarized Structure of SYSCONFIG	9-67

LIST OF TABLES

Table	Title	Page
4-1	Peripherals Required by CMS-Common Utilities	4-2
4-2	DSKUTL – Supported Disk Types	4-63
4-3	DSKUTL IN Bootstrap Table	4-64
4-4 A	DSKUTL BOOT Bootstrap Table If MTR Specified	4-83
4-4B	DSKUTL BOOT Bootstrap Table If COLDSTART Specified	4-83
4-5	File Attributes Accessible by MODIFY	4-166
4-6	PPB Attributes Accessible by MODIFY	4-167
4-7	Mnemonics for Device Attribute for MODIFY.	4-168
5-1	Sign Convention for Signed 8-Bit Alphanumeric Fields	5-13
5-2	Sign Convention for Signed 4-Bit Numeric Fields	5-14
5-3	Sign Convention for Separate Sign Character with 8-bit Alphanumeric Fields	5-14
6-1	Valid CMS Devices	6-10
6-2	Default Settings for Parameters Modifiable Through CO	6-12
6-3	CO Error Messages	6-24
6-4	Zip Failure Messages	6-28
6-5	CO Reserved Words	6-31
9-1	Coldstart Functions	9-8
9-2	Bank A Hex Displays for Startup Programs	9-29
9-3	General Description of Warmstart Messages	9-36
9-4	Disk Attribute Default Values for IN, FE, and RF Functions	9-43
9-5	ROM Dump Display Reference Table	9-51
9-6	Bank E Display for ROM Dump	9-51
9-7	Pack Initialization Procedure	9-55
9-8	CONFIGURER Printer Translation	9-74

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.

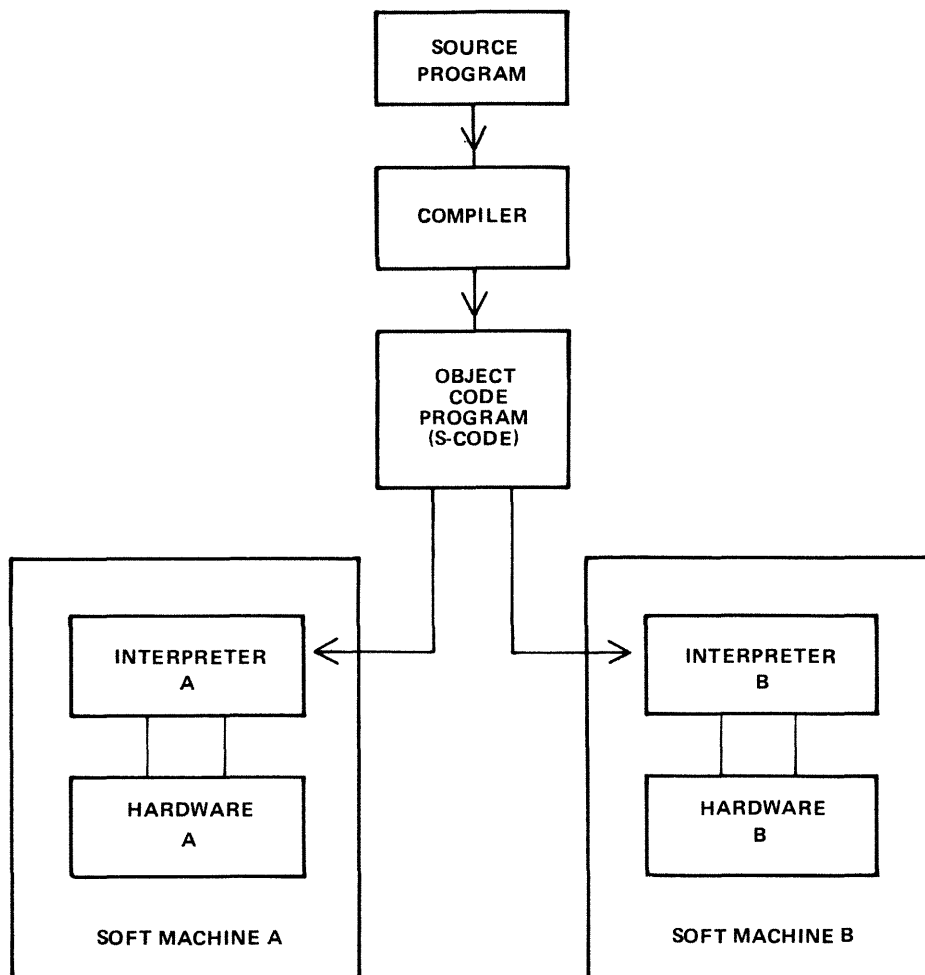


Figure 1-1. CMS Portability

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 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.

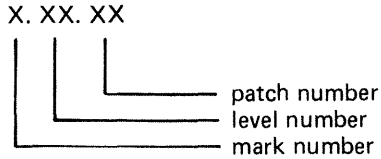
Certain common features needed in all programs (such as the handling of peripheral devices) have been collected together into a Master Control Program (MCP). The MCP is a micro-code program and is therefore specifically written for each hardware product. Thus there is a B 90 MCP, a B 1800 MCP and a B 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.

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 3.01.08 is the COBOL compiler included in the 3.01 release of system software, with patch number 08.

Software items from different releases should not be used together. For example, an interpreter from release 3.01 should not be used with an MCP from release 3.00.

This book describes system software relative to the 3.04 release.

Software Patches

Within a particular release, patches to individual items may be issued. For example, an MCP identified by 3.02.12 contains certain improvements over an MCP identified by 3.02.11. 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 (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)
- 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.

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).

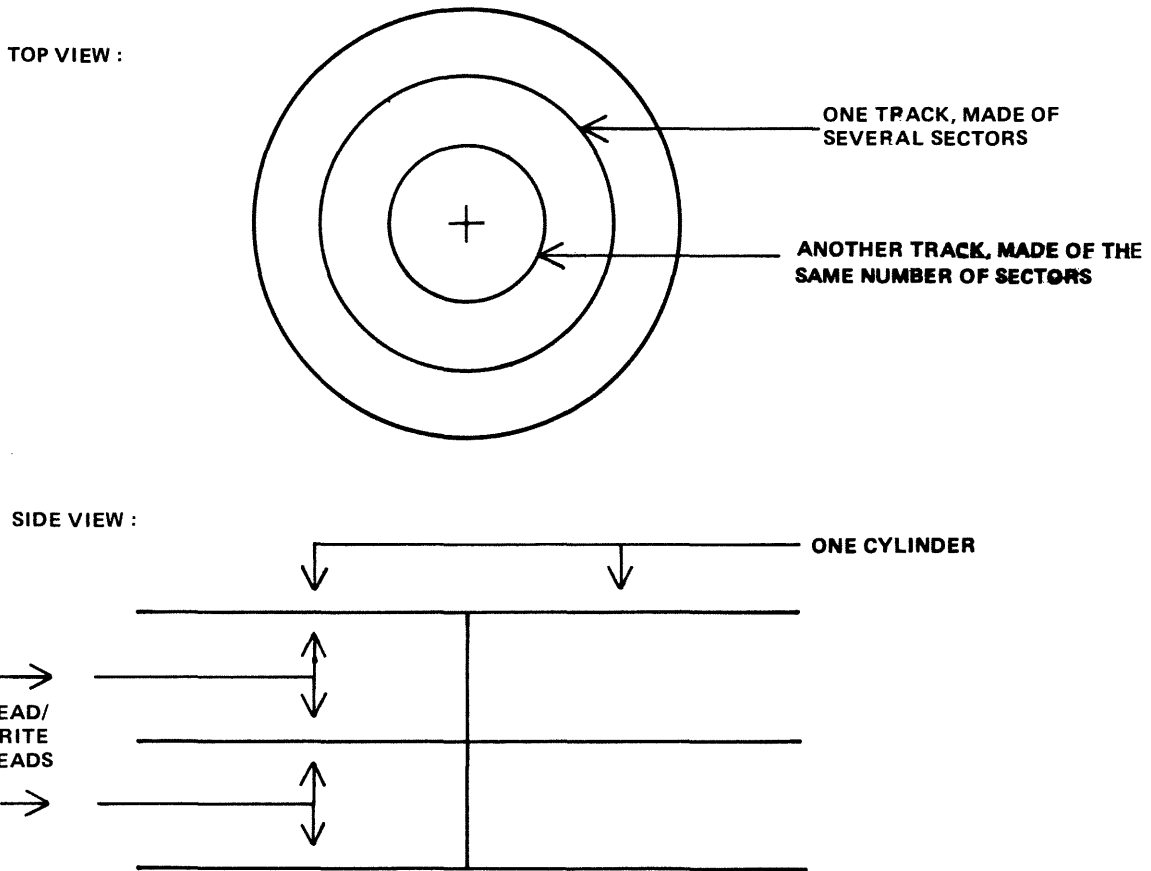


Figure 2-1. Physical Disk Structure

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 (“-”).

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.

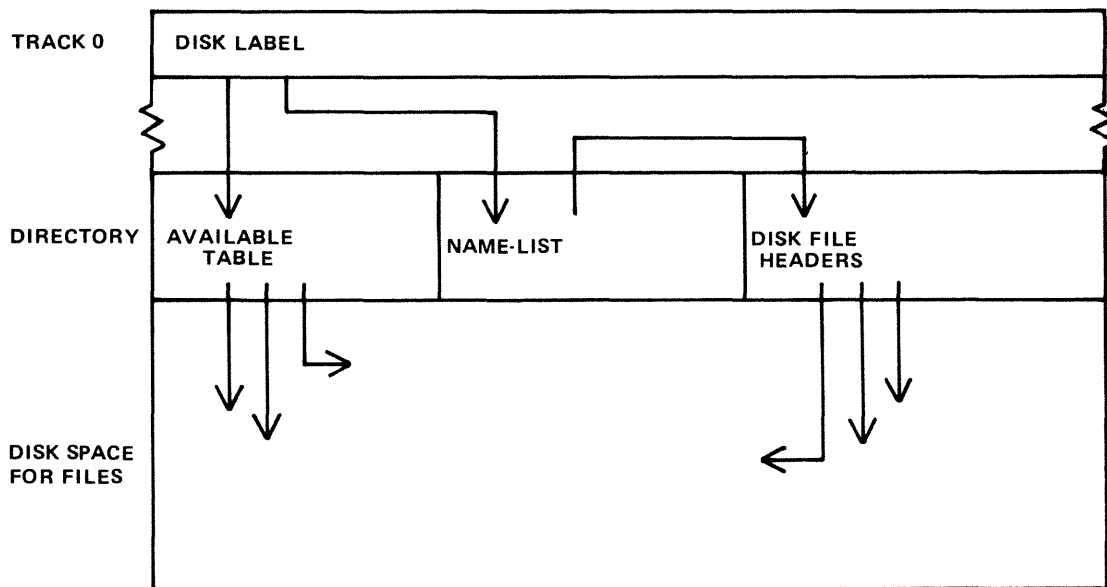


Figure 2-2. Disk Directory Structure

If there is insufficient space on a disk to allocate new disk file areas, a "NO USER DISK" message is given by the MCP. 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 Stand-Alone utility LS will also give the available space on a disk whose files have been listed by an LS/=.

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.

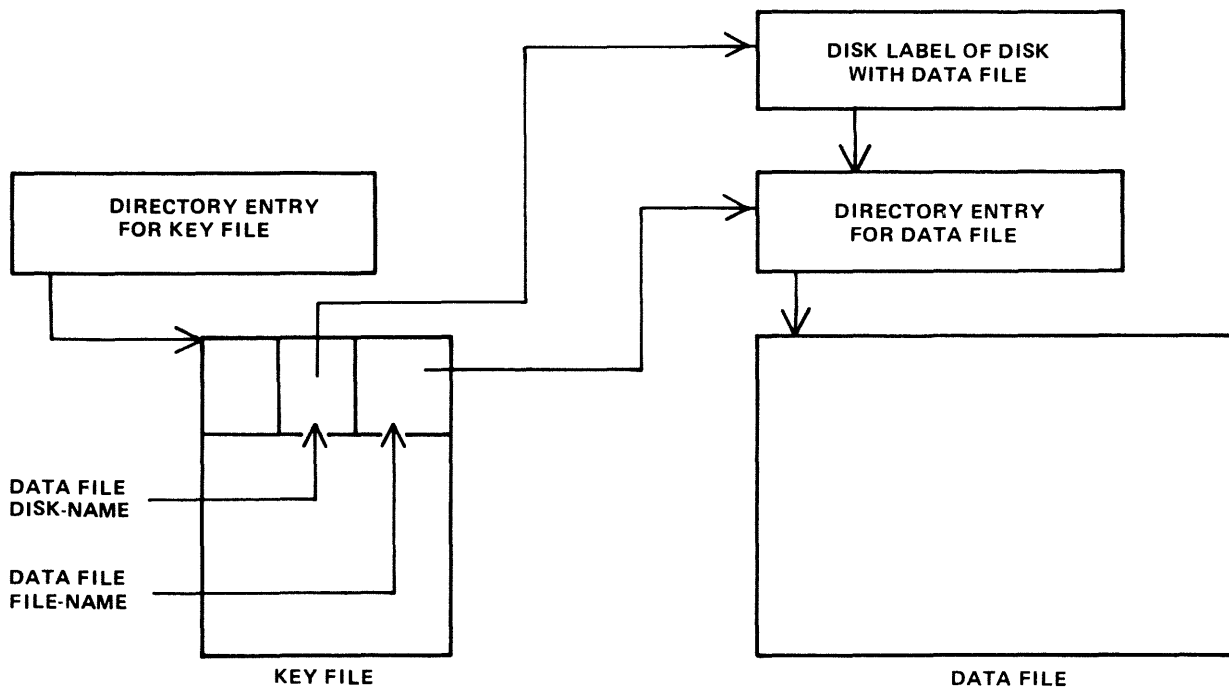


Figure 2-3. Indexed Files

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).

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”. Such files may be created by the AD intrinsic in response to a “NO USER DISK” message (see section 3).

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.

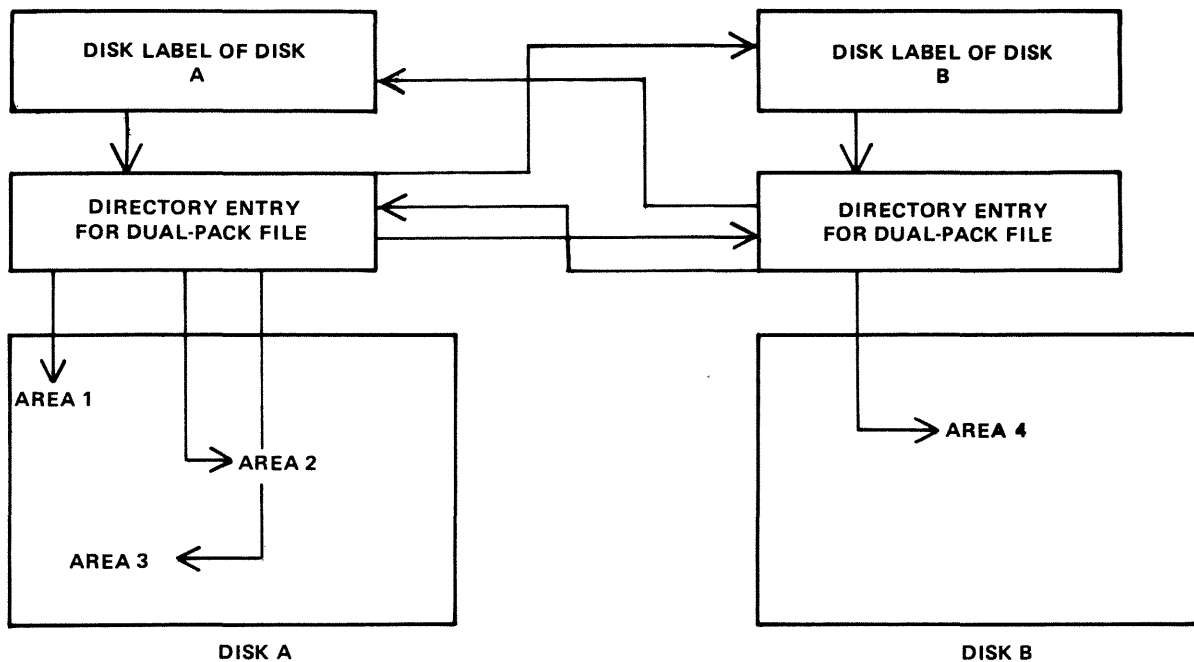


Figure 2-4. Dual Pack Files

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.

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 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 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. This file-name is also used in MCP messages. Refer to the CMS MCP 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.

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 "intrinsic" 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".

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.

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/LIST <17> WAITING UNLAB LISTPRT AP NO FILE

indicates an MCP message with event number 17, and reference should be made 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 needs space on a particular disk, a "NO USER DISK" 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).

```

input command to run LIST --> LIST COLLETTE
MCP output message event 10-> 01/LIST <10> WAITING COLLETTE DK NO FILE
input command to run PROGA -> PROGA
MCP message for PROGA BOJ --> 02/PROGA BOJ PR IS A
input command to run LR ----> LR =
next line is PROGA display -> 02/PROGA DISP:
actual display information -> PROGRAM A VERSION 3.01.05
input request OL intrinsic -> OL LPA
MCP response to OL message -> LPA LRPRINT IN USE BY 03/LR
input request MX intrinsic -> MX
MCP response to MX message -> 01/LIST SUSPENDED WAITING ON NO FILE
-> ...CONDITION
-> 02/PROGA A EXECUTING
-> 03/LR B EXECUTING
MCP message for PROGA EOJ --> 02/PROGA ECJ
input request ST intrinsic -> ST 3
MCP response to ST message -> 03/LR STOPPED

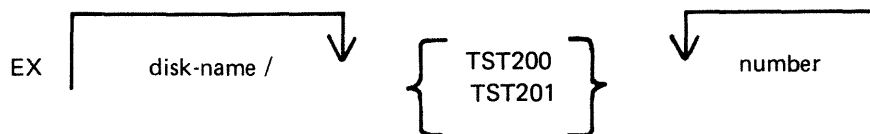
```

Figure 2-5. Sample SPO List

FORMAT DIAGRAMS

Most of the descriptions of input messages in this book are given as simple format diagrams with corresponding descriptive text and examples. An example will illustrate how to read such format diagrams.

Example:



In this format, items in lower-case (“disk-name” and “number” in this example) are to be replaced by actual values (such as “PR2” and “27”). Other items are included in the input message as they are found. Spaces are required whenever necessary to avoid ambiguity. In the example, it is not strictly necessary to separate the disk-name and the slash (“/”) with a space because the slash cannot be part of the disk-name according to the rules for disk-names. Extra spaces may, however, be added for legibility. If an arrow in the left-to-right direction is encountered, the items under the arrow may be omitted. Curly brackets are used to denote alternatives. The alternatives are placed in a list underneath each other. (Each alternative item may be more complex than the example quoted: it may contain optional parts and further alternatives.) If an arrow in the right-to-left direction is encountered, one may return to the point underneath the arrow and continue building up a valid input message. In the example quoted, after adding a valid number (say “27”) one may return to add a second number (say “52”). In fact, the format diagram does not specify how many times one may continue to do this, but details are given in the text.

Here are several valid input messages which can be generated from the example (note that a disk-name can consist of up to seven characters, see earlier):

EX TST200 57

```

EX TST201 259
EX PR2/TST200 36
EX PR2/TST200 2 52 574 361
EX M101A/TST201 1 2

```

Here are several invalid input messages according to the example:

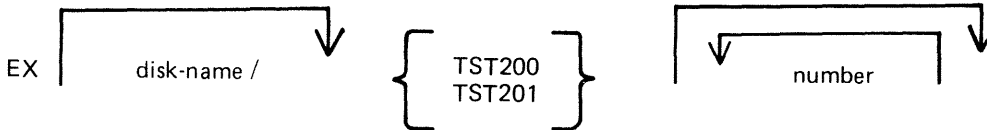
```

EX PR2/TST200
EX PR2 TST200 36
EX TST202 36
TST201 259
EX PR2/M101A/TST201 1 2

```

Example:

Here is a slightly more complicated example, which makes the number or list of numbers optional:



The input messages:

```

EX PR2/TST200
EX PR2/TST200 56
EX PR2/TST200 27 56

```

are now all valid.

These simple format diagrams are easy to understand in conjunction with descriptive text and examples, but cannot be used if the format becomes too complex. In the latter case, a rigorous notation known as “railroad diagrams” is used (see below). In some cases in the text of this book, the format has been deliberately simplified for the sake of clarity, with further details given in the text. More complex features have been described by railroad diagrams (see, for example, the COPY and LIST utilities in section 4). Appendix A gives complete railroad diagrams as a handy reference for those who need the exact definition on any input message.

RAILROAD DIAGRAMS

The equivalent railroad diagram to the first format diagram is given in figure 2-6.



Figure 2-6. Railroad Diagram Sample 1

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.

The equivalent railroad diagram to the second format diagram is given in figure 2-7.

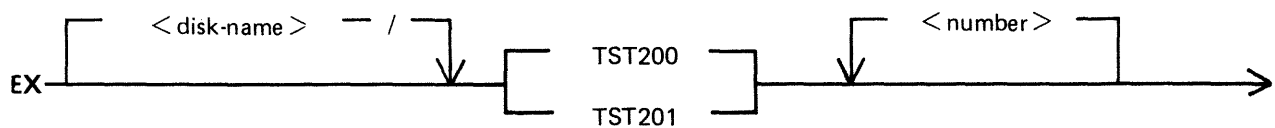


Figure 2-7. Railroad Diagram Sample 2

There are two other features available in railroad diagrams to make possible the exact specification of any input message. These are illustrated in figure 2-8.

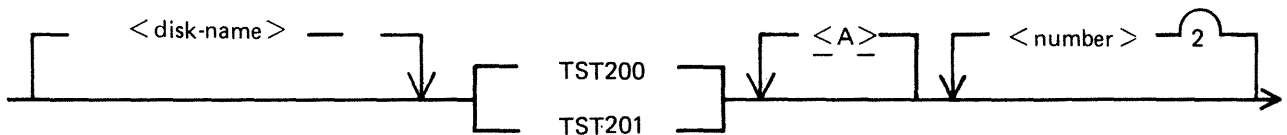


Figure 2-8. Railroad Diagram Sample 3

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.

As discussed in section 2, 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 by nature they are part of the MCP which is on the system disk.

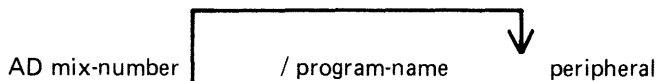
The response to these intrinsics may vary slightly between different CMS products, due to different hardware being used. These variations have been noted in the text where applicable.

AD (Assign Peripheral Device)

This intrinsic allows the operator to assign a particular peripheral that has called for an “unlabelled input file”, or that requests a particular output device.

It may also be used to allow file overflow onto a second disk if no disk space is available.

Format diagram:



Example 1:

Copy utility requires another disk:

```
COPY INIST TO INDISK3/INIST
10/COPY <12> WAITING FILE 10 NO
... USER DISK
AD 10 DMB
```

(The first message is output by the MCP and the operator responds with the AD message by assigning DMB as the disk to which the remainder of file INIST will be copied. This creates file INIST as a “dual-pack file”).

Example 2:

Program “COBOL7”, mix number 03, requires a line printer type device:

```
03 COBOL7 <17> WAITING LP NO FILE
AD 03 LPA
```

(The first message is output by the MCP, and the operator responds with the AD message by assigning LPA to mix number 03).

Example 3:

The LIST utility requires an unlabelled tape:

```
LIST TAPE1 MTP NO.LABEL
01/LIST <14> WAITING UNLAB SPURIUS/TAPE1 AT
...DEVICE REQUIRED
AD 01 CTB
```

Output Messages:

Message	Possible Causes	Suggested Action
mix number/program AD INVALID	Specified program was not suspended waiting for a device assignment.	Check with MX for name of suspended program.

AX (Accept a message for a program)

This intrinsic allows the operator to communicate with a program in the mix. The program must already be suspended waiting for an “accept” (ACPT).

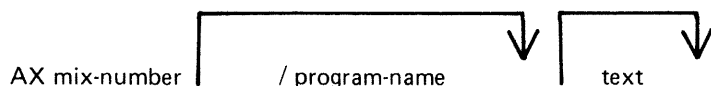
The MCP will prompt the operator for input by printing “mix number/program-name ACPT” on the SPO.

The total number of characters input including AX and mix/program name is 255. The number of characters in the text field is determined by the program.

NOTE

“Text” begins at the first non-blank character following mix number (/program-name).

Format diagram:



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.

```
BM001
01/BM001 BOJ
ENTER BM202 FILE NAME
01/BM001 ACPT
AX 01 ARSCHG
```

Output Messages:

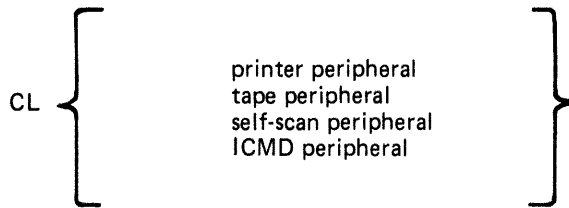
Message	Possible Causes	Suggested Action
mix-number program name AX INVALID	Specified program was not waiting for an “accept”, or mix number and specified program name do not match.	Check with MX for proper mix-number and program-name combination.

CL (Clear Peripheral)

This intrinsic allows the operator to clear the peripheral from the program and bring the program to End of Job (EOJ). It breaks the “links” between the program and the peripheral.

For example, if the line printer “hangs” during the printing of a report and an attempt is made to DS the program, it will not be possible to discontinue the program unless the line printer is made ready or CL is used to break the “link” between the program and the line printer.

Format diagram:



Examples:

```
CL LPA
CL SSA
```

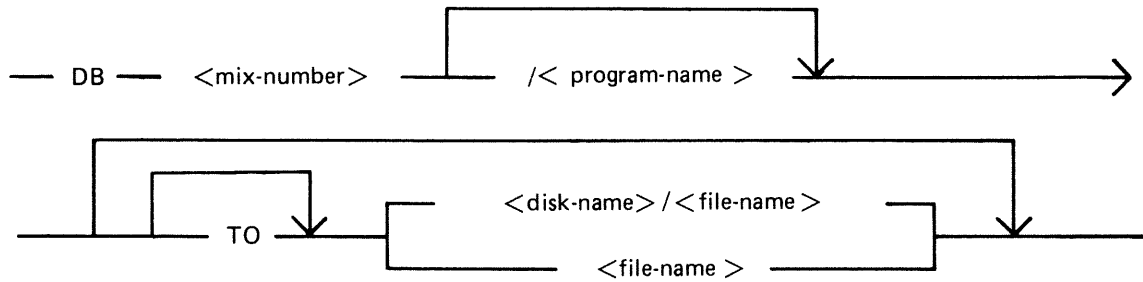
Output Messages:

Message	Possible Causes	Suggested Action
CL peripheral INVALID	Program is not waiting on “hung” peripheral, and has not been DS’ed or DP’ed (see later).	Check input.

DB (Direct To Backup)

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 defaults contained in SYSCONFIG file (see CONFIGURER).

Railroad diagram:



<mix-number>

This is the mix number of the program requiring a printer.

<program-name>

This is the name of the program requesting use of a printer.

<disk-name> / <file-name>

If a disk and file name are specified, the printer backup file is placed on the named disk with the given file name.

<file-name>

The printer backup file is written to the system disk with the given file name.

NOTE

If neither a disk and file name or file name are specified, then the printer backup designated disk (see CONFIGURER for specifying printer backup designate disk) and a filename of the form PBnnnnn (where nnnnn is a five digit progression number in the range 00001 to 65535) will be used.

Examples:

Example 1

The program WRITER has the following message output by the MCP:

```
22/WRITER <28> WAITING WRPRINT AP DEVICE  
ASSIGNMENT REQUIRED
```

To direct this program's output to backup using the system default parameters, the following input is used.

```
DB 22
```

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

Example 2

To direct the output of the program PRINTER, mix number 22, to printer backup on the disk named PRINT with the file name FNAME, the following syntax is used:

DB 22/PRINTER TO PRINT/FNAME

DB Error Messages:

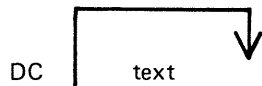
Error Message	Description
<faulty input> INVALID MIX NUM	No mix number or the wrong mix number was given in the initiating message.
<faulty input> INVALID	The mix number specified did not require a printer or the initiating message was incomplete.
<disk-name> INVALID – SPECIFIED PRINTER BACKUP DISK NOT ONLINE	The disk named in the initiating message is not present. Load and ready the required disk and re-issue DB.
<faulty input> INVALID – DUPLICATE BACKUP DISK	There already exists a disk file with specified name on the system. Remove one of these disks and re-execute DB. This will only occur on systems which run with Pseudo packs. Remove one of the duplicate disks.

DC (Data Communications operator input)

The intrinsic enables the operator to enter messages from the SPO to the Message Control System (MCS) if data communications activity is in process. The message text, after being stripped of the “DC” characters and the following blank character, 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.

Format diagram:



Example:

To enter the text “MAKE STATION 2 READY”:
DC MAKE STATION 2 READY

Output Messages:

Message	Possible Causes	Suggested Action
none	MCS input successful	none
DC INVALID	no MCS in the mix	check input; execute the MCS
DC NOSPACE	There is no available message space in memory for this message	Wait a short time then re-input message; if unsuccessful several times, request technical assistance.

DP (Discontinue and Dump)

This intrinsic is similar to the “DS” intrinsic. The difference is that the disk work space (Virtual Memory on Disk, Virtual Disk) is not freed up and returned to an available status.

The disk work space is, instead, updated from memory with all the most current information about the program. The disk backup is then made into a file (locked) and given a name, “DMFILnn” (‘nn’ is the mix number for user programs, utilities, and MCP intrinsics).

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.

Format:

DP mix-number/program-name

Example:

DP 01/GL060

Output Messages:

Messages	Possible Causes	Suggested Action
mix number/program-name DP'ed	DP successful	none
input INVALID	mix number does not correspond to program name	Check input (re-input if necessary), or check with MX for proper mix number and program name combination.
input INVALID – NEEDS PROGRAM-ID	program-name is missing	Check with MX and re-input

DS (Discontinue Program)

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

Format:

DS mix-number/program-name

Example:

To terminate the program AR040 which has the mix number 2:

DS 02/AR040

Output Messages:

	Message Possible Causes	Suggested Action
mix/prog DS'ed	DS successful	none
input INVALID	mix number does not correspond to program name; or program is an MCS	Check with MX, re-input
input INVALID – NEEDS PROGRAM ID	program name not specified	Check with MX, re-input

Note: if the program is waiting on a “hung” peripheral device, try the CL intrinsic.

DT (Systems Data and Time)

This intrinsic allows the operator to inquire about or change the system date and time maintained by the MCP.

Format diagram:



Examples:

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

DT

To change the system date:

DT 01/01/82

To change the system date and time:

DT 03/23/82 1234

(March 23, 1982 is the new date, 12:34 is the time).

Output Messages:

Message	Possible Causes	Suggested Action
“DD MON YY YYDDD HHMM DOW” where DD = day of month, MON = 3 letter abbreviation of month, YY = year, YYDDD = Julian date, HHMM = time (hours and minutes), DOW = day of week	Normal response to “DT”	none
<INVALID DATE>	An error was made in one of the following fields: MM DD YY. For example, a MM entry of “0” or greater than 12 is invalid. The entire date is rejected, but a valid time entry in the same message will be accepted.	Re-input date portion of message
<INVALID TIME>	A time greater than 2359 was entered. The time is rejected. A valid date entry in the	Re-input time portion of message.

Message	(continued) Possible Causes	Suggested Action
<NO CLOCK>	same message will be accepted. Time entry was made, but system has no real-time clock. Valid date entry will be accepted in the same message.	none
MM/DD/YY HH:MM	Normal response to DT inquiry .	none

FD (Form Define)

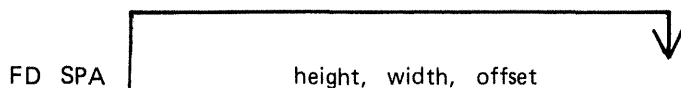
This intrinsic allows the operator to define a logical page for a serial printer (SPA) or set top of page for the SPA.

Unless the operator indicates otherwise, the current position is taken as the top of the page.

If the three parameters (HEIGHT, WIDTH, and OFFSET) are specified, then they are used to define a logical page on the SPA. HEIGHT specifies the number of lines on a logical page; WIDTH the maximum number of characters in one line; and OFFSET the number of characters that the printing area is to be offset from the left. An OFFSET of zero specifies the left-most physical position.

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 has 255 columns the maximum printing area is given by a WIDTH of 255 and OFFSET of zero. The logical page will remain the same as defined by FD or next warmstart.

Format diagram:



Examples:

1. FD SPA 66, 120, 5

defines a logical page on SPA where height is 66 lines and the printing area is 120 characters wide offset 5 columns from the left (that is, from columns 6 through 125, numbering the left-most column as column 1).

2. In order to change the top-of-page position, move the paper to the desired position then execute

FD SPA <empty>

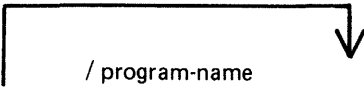
Output Messages:

Message	Possible Causes	Suggested Action
FD SPA numbers INVALID	FD specifications for height, width, and/or offset are not acceptable. Attempt to print beyond SPA capabilities.	Check input and re-enter.

GO (Restart a Stopped Program)

This intrinsic allows the operator to restart a program which has been stopped with the “ST” command.

Format diagram:

GO mix-number  / program-name

Examples:

To restart program whose mix-number is 3:

GO 3

To restart program PR020:

GO 3/PR020

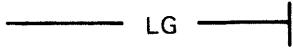
Output Messages:

Message	Possible Causes	Suggested Action
mix-number/prog-name NOT STOPPED	Specified program was not waiting for a “GO” command.	Check with MX for suspended program waiting for “GO”. Re-input.
mix-number/prog-name INVALID	Optional program name was used and it did not match the mix number specified.	Check with MX for correct mix number and matching program name. Re-input.

LG (Log Go)

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 upon, or countermanded by the LS intrinsic.

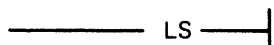
Railroad diagram:



LS (Log Stop)

This intrinsic stops the logging of system messages. If logging was initiated by LG, then error logging messages will also be stopped. Logging will remain stopped until either the system is powered off, at the next warmstart the default option set using CONFIGURER will be acted upon, or countermanded by the LG intrinsic.

Railroad diagram:



MX (Display Current Mix)

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

Format diagram:



Examples:

To inquire about all programs currently processing:

MX

To inquire about a particular program:

MX 03/PR020
or MX 03

Output Messages:

Message	Possible Causes	Suggested Action
INVALID MIX	Specified program is not currently running.	Check input (re-input if necessary).
NULL MIX	No programs are currently processing.	None
INVALID PROGRAM ID	Optional program name was used and it did not match the mix number specified.	Re-input

For each program specified, the following information is provided:

MIX NUMBER

a number assigned by MCP to this program as it was loaded into memory.

PROGRAM NAME

PROGRAM PRIORITY – “A”, “B”, “C”

A = lowest priority (that is, application program)

B = medium priority (that is, system priority)

C = highest priority (that is, data communication)

STATUS OF PROGRAM

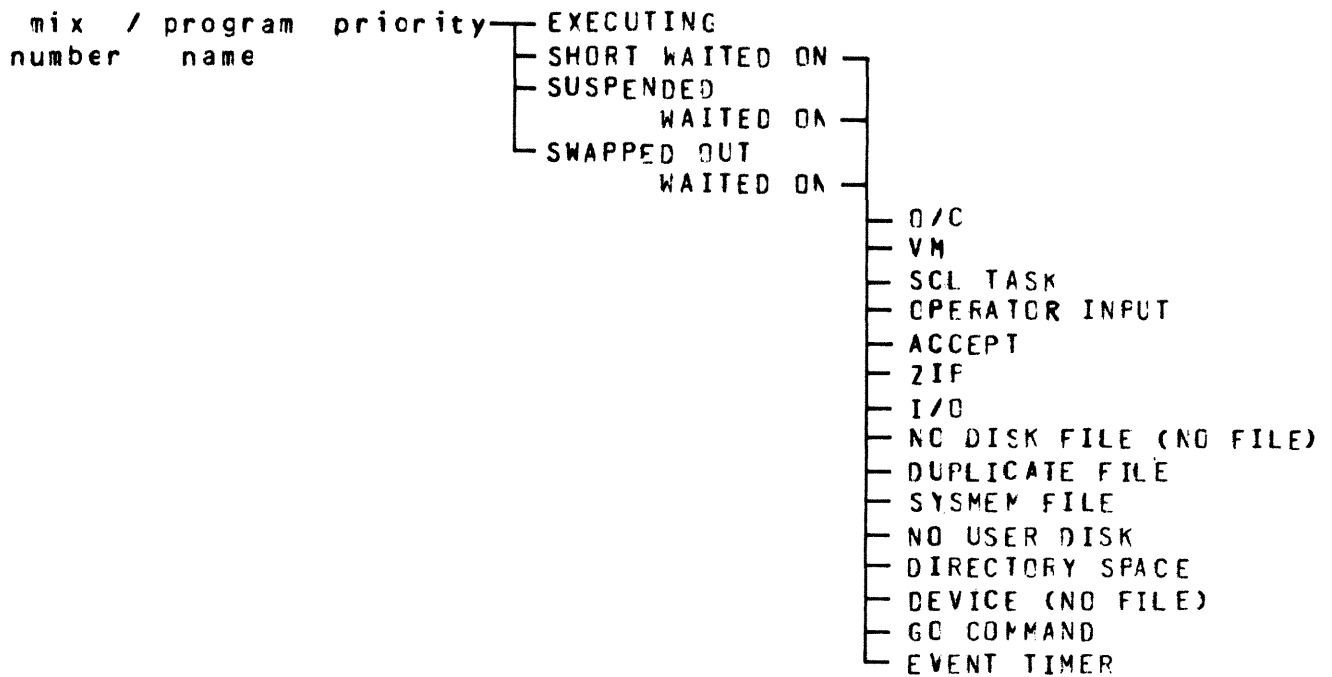
EXECUTING – program processing normally

SUSPENDED WAITED ON – program processing was temporarily halted. For reasons, see chart below.

SHORT WAITED ON – program is waiting on a resource (that is, Virtual Memory or I/O buffer) which the system can guarantee will be made available on a resource in a relatively short time.

SWAPPED OUT WAITED ON – portions of this previously suspended program were temporarily removed from real memory and returned to disk. Memory space was required for other programs in the mix. (Reasons for “swap outs” are same as for program suspension).

Possible messages are summarized by the chart below:



Output message examples:

Message	Possible Causes	Suggested Action
04/PR060 A EXECUTING	Program processing normally	None
04/PR060 A SUSPENDED WAITED ON O/C	Program is waiting on a file open or close.	None: program will be resumed when file has been opened or closed
04/PR060 A SUSPENDED WAITED ON VM	Program is waiting on Virtual Memory	None: do not try to execute too many programs at this time.
04/PR060 A SUSPENDED WAITED ON SCL TASK	Program is waiting for a “command” from the MCP to be completed (such as response to an “OL” input).	None: program will be resumed when SCL task has completed.

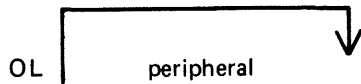
Message	(continued) Possible Causes	Suggested Action
10/LR B SUSPENDED WAITED ON OPERATOR INPUT	Program is waiting for some input from operator. (EX: A program previously suspended by ST requires a GO command to continue)	Provide program with appropriate input. Program will continue processing.
08/GL060 A SWAPPED OUT WAITED ON ACCEPT	Program has displayed an "ACPT" message on SPO and is waiting for appropriate response.	Refer to this program's operating instructions for suggested responses to ACPT. Then enter AX, mix number and selected response.
05/AP020 A SUSPENDED WAITED ON ZIP	Program requested assistance of another program in order to complete this job. MCP will automatically load into memory the necessary program(s).	None
04/PR060 A SHORT WAITED ON I/O	Usually indicative of normal processing, involving I/O activity disk or peripheral.	None
05/PR020 A SUSPENDED WAITED ON NO DISK FILE	Program needs (and has not found) a particular file order to continue processing.	Check SPO for message indicating name of file this program is seeking, then supply missing file (COPY from backup medium or create it). If in doubt refer to program instructions.
02/PR020 A SUSPENDED WAITED ON DUPLICATE FILE	Program is attempting to place a file of a certain name on disk. However, another file by the same name currently resides on disk. A disk may not contain two files with the same name.	Normally, remove the existing file from disk with RM. If in doubt, refer to program instructions.
10/COPY B SUSPENDED WAITED ON DIRECTORY SPACE	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.
10/COPY B SUSPENDED WAITED ON NO USER DISK	There is no more available space on disk; or space available is insufficient to hold the file the system is attempting to write; or if disk is "checkerboarded"	With KA, analyze amount of available space remaining. If disk is filled, remove with RM any unnecessary files; or if disk is filled and dual-pack file is desired, assign a different disk to this program (see AD

Message	(continued) Possible Causes	Suggested Action
10/LIST B SUSPENDED WAITED ON SYSMEM FILE	SYSMEM file cannot be located.	intrinsic); or if disk is checkerboarded, use SQ utility to consolidate disk space, then re-execute program that encountered suspension. Request technical assistance
04/PR060 A SUSPENDED WAITED ON NO FILE	Device that a program needs in order to continue processing is either unavailable or not ready; or Program needs (and has not found) a particular file in order to continue processing	RY required device; or assign program to alternative device (see AD intrinsic) Check SPO for message indicating name of file program is seeking. Supply missing file (COPY from backup medium or create).
04/PR060 A SUSPENDED WAITED ON DEVICE	Device that a program needs in order to continue processing is either unavailable or not ready.	RY required device; or assign program to alternate device (see AC intrinsic)
02/LR B SUSPENDED WAITED ON GO COMMAND	Program was suspended by ST command.	Type "GO" plus mix number of suspended program.

OL (Request for Status Information of Peripherals)

This intrinsic allows the operator to request the status of peripherals on the system.

Format diagram:



Examples:

To display status of all peripherals:

OL

To display status of a particular peripheral:

OL DKB

OL LPA

Output Messages:

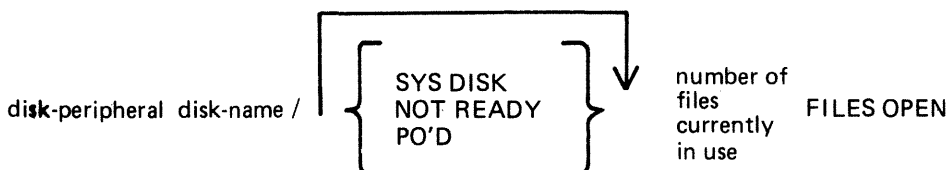
Message	Possible Causes	Suggested Action
peripheral NOT READY	Peripheral is not on the system; peripheral may have been "saved"; peripheral may not be correctly loaded.	Check input (re-input if necessary). Ready peripheral if necessary.
OL peripheral NOT ON SYSTEM	peripheral is not configured on machine	none
OL peripheral INVALID	A non-existent device has been specified (that is, OL CCC)	Check input (re-input if necessary)

Other output messages produced by OL depend on type of peripheral. Refer to the following examples for details.

Examples of disk device output:

The general format of the output message is:

Format diagram:



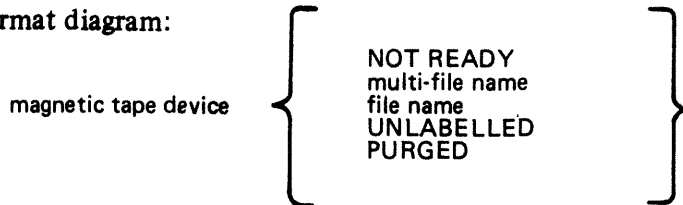
Examples:

DKA AR1/0 FILES OPEN
 DKB AR2/SYS DISK 2 FILES OPEN
 DMA PRA/NOT READY 0 FILE OPEN
 DKA AR1/PO'D 0 FILES OPEN

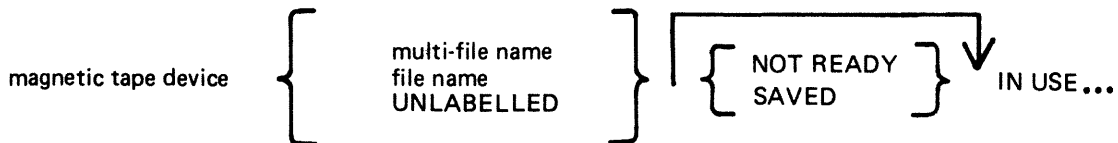
Examples of magnetic tape device output:

The general format of the output message is:

Format diagram:



or



... BY mix-number / program-name

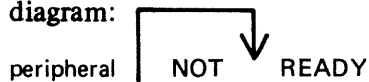
Examples:

CTA NOT READY
 CTA ARTAPE
 CTA ARTAPE/IN USE BY 10/TAPELR
 CTA ARTAPE/NOT READY IN USE BY 10/TAPELR

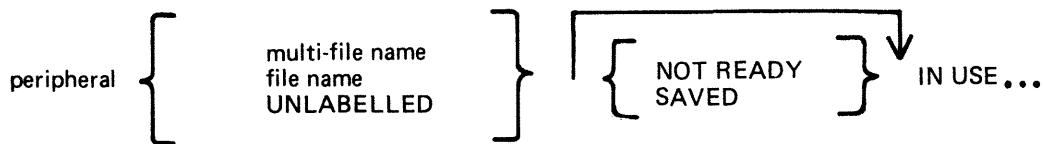
Examples of output from any other device:

The general format of the output message is:

Format diagram:



or



... BY mix-number / program-name

Examples:

LPA NOT READY
 LPA NOT READY IN USE BY 04/PR020
 SSA SAVED
 SPA SAVED IN USE BY 06/PR060

PG (Purge Tape)

This intrinsic allows the operator to purge (erase) magnetic tape and cassette tape files, thus labelling them as available for output.

Format:

PG tape or cassette peripheral

Examples:

To purge a cassette tape on drive CTA:

PG CTA

To purge a magnetic tape on drive MTC:

PG MTC

Output Messages:

Message	Possible Causes	Suggested Action
peripheral * PURGED *	PG successful	None
PG INVALID	peripheral not specified in message	Re-input message
PG peripheral INVALID	Tape could not be purged, as it is "write inhibited", or peripheral is not on the system.	Make certain red tabs on top of cassette are turned outward; make certain "write permit ring" is inserted in tape. Retry PG.

Note: if an attempt is made to purge a tape which is in use, then the response to the OL message for that peripheral is displayed.

PO (Power Off a disk)

This intrinsic allows the operator to power off a disk “logically” (instruct the MCP that the disk is no longer required). At any time when the MCP is idle it is valid to power off the system disk logically with the PO command. This will cause the MCP to terminate. All systems disk files will be closed and SYS-SUPERUTL will go to End of Job (EOJ).

No disk should be removed from the disk drive, no disk units should be powered down, nor should the main cabinet be switched off, until disks have been logically powered off with PO. Failure to observe this practice might cause disk problems at a later date.

Format:

PO disk peripheral

Examples:

PO DKA (disk cartridge)
PO DMB (mini disk)
PO DFA (fixed disk)

Output Messages:

Message	Possible Causes	Suggested Action
disk peripheral OK or disk peripheral POWERED OFF	Disk was logically powered off.	It is now permissible to power off and remove the disk from the disk drive physically.
disk peripheral REMOVED WITHOUT PO	Disk was physically removed before being logically powered off.	Check disk for possible corrupted data before re-use.
PO disk peripheral INVALID	specified disk peripheral is non-existent (ex: PO DKW)	Re-input
PO disk peripheral NOT ON SYSTEM	Specified disk is not currently on line.	Re-input
CANNOT POWER OFF SYSTEM. MIX NOT EMPTY or PO disk peripheral INVALID	Attempt has been made to PO the system disk while a program is processing.	Allow program to go to End of Job (EOJ), then re-input

If an attempt to Power Off a disk is made while files on that disk are in use, the OL message for the disk is printed. No further program will be allowed to open files on the disk and when all files in use have been closed, the disk will be logically powered off. If the disk is in use, it will not be powered off immediately after giving the PO command though it will print “disk peripheral PO’ED X FILES OPEN”, for all disks.

If a disk is removed without being logically powered off, any program using files on that disk will eventually terminate with an error condition indicating hardware failure.

A PO’ed user disk may be made ready again by the RY command or by physically powering the unit off and on.

PR (Assign Program Priority)

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

Priority "A" is low or normal priority, used for regular work. Within this class, programs which perform more physical I/O operations are given precedence.

Priority "B" 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.

Priority "C" 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 I/O are given precedence. This intrinsic is not implemented on B 90 systems.

Format diagram:

PR mix-number / program-name { A
B
C }

Example:

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

```
PR 03/REP506 B
```

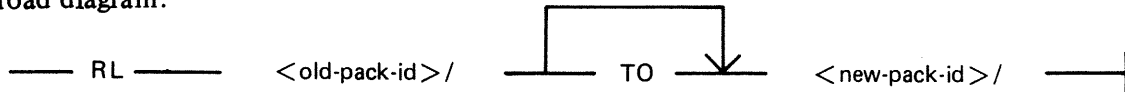
Output Messages:

Message	Possible Causes	Suggested Action
mix-number/program-name PR IS priority	Input accepted	None
mix-number/program-name PR INVALID	Mix-number and program-name do not match or priority value incorrect	Check with MX for proper input, and re-enter.

RL (Relabel Disk/Pack)

The RL intrinsic enables the operator to change the disk-name/packid and also provides a mechanism for changing the pseudo-packids that are defined for the system.

Railroad diagram:



where <old-pack-id> and <new-pack-id> can be a disk name, packid or a pseudo-packid depending on what is supported on the system that RL is being executed on.

Two conditions are required for this intrinsic to be accepted.

1. If a pseudo-packid is being relabelled then all physical units that have packids listed in the PPIT must be physically ready.
2. A null mix must exist. On a remote SPO system the only programs running should be SPIM, the MCS that is managing the remote system operation (usually MCSOBJ) and NDLSYS, if this condition only applies for the system disk.

Disk Name

When a disk name is specified, the disk name contained in the label of the named disk is changed to the new disk name given.

Pseudo-packid and Packid

When the packid requested to be changed is a pseudo-pack, this intrinsic 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 pack, this intrinsic also causes the packid in the disk label field of the specified physical unit to be updated with the new packid and assigns the unit to the system with this new packid.

WARNING

RL will not correct disk file headers of dual pack file components or KFPB's.

When RL has completed the relabelling successfully, the message

<valid input> COMPLETE

will be displayed.

RL Error Messages:

Error Message	Description
<faulty input> INVALID	This is caused by one of the following errors: 1. A physical unit from the PPIT is not ready. 2. The packid/disk name is illegal
<faulty input> INVALID – PROGRAMS IN MIX	The operator has tried to relabel the system disk without having a null mix. That is, there were programs other than SPIM, NDLSYS and MCSOBJ (or the equivalent) run-ning. Ensure that null mix exists before using RL.

(continued)

Error Message	Description
<faulty input> INVALID - DISK IN USE	There is another program using this disk. Ensure that no programs are using the disk before using RL.
RL <packid or disk-name>/ DUPLICATE PACK	There is already a disk name or packid by that name on the system.
RL <disk-name or packid> NOT ON SYSTEM	This packid or disk-name is not on the system.
RL FAILED - I/O ERROR ON <device>	A hardware error prevented RL from being executed.

RY (Ready a Peripheral)

This intrinsic is used to “ready” a peripheral so the MCP can use it as a resource. When warmstarting, the system will automatically ready all peripherals on the system that are powered on. RY may also be used to Ready a previously PO’d user disk.

Format:

RY peripheral

Examples:

To ready a self-scan:

RY SSA

To ready a line printer:

RY LPA

Output Messages:

Message	Possible Causes	Suggested Action
RY peripheral INVALID	Attempt was made to ready a non-existent peripheral (that is, RY LLP); attempt was made to ready a device already “ready”.	Check input (re-input if necessary).
RY peripheral NOT ON SYSTEM	Attempt was made to ready a peripheral not on-line to the computer.	Check input (re-input if necessary).
Peripheral READY	RY successful	None

SF (Substitute Disk File)

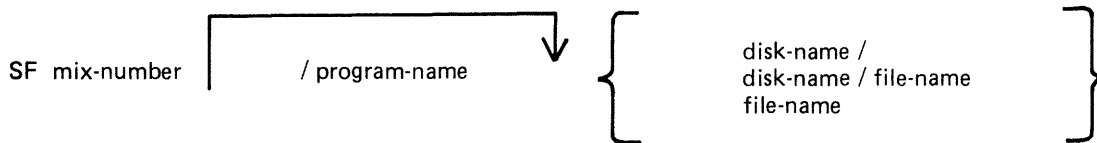
This intrinsic allows the operator to direct a program to a particular disk file if it is waiting on a “NO FILE”, “NO PACK”, “DUPLICATE FILE”, or “BAD FILE NAME” condition.

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

The command can only be used when the program is suspended waiting on one of the above conditions. It is not possible to anticipate the program’s requirements and modify the file parameter block in advance.

This intrinsic is not implemented on B 90 and B 900 systems.

Format diagram:



Examples:

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

```
01/AP10 <10> WAITING APD/APD2T DK NO FILE
SF 01/AP10 APD2S
```

(The first line is the MCP output message; the second is the input SF message in response to the “NO FILE” condition).

To direct the same program to use file APD2T on APD1:

```
SF 01/APD10 APD1/
or
SF 01 APD1/
```

To direct the same program to use file APTEMP on disk ARTD:

```
SF 01 ARTD/APTEMP
```

Output Message:

Message	Possible Cause	Suggested Action
mix-number/program-name SF INVALID	Program is not waiting on a “no file” or other condition, or mix-number and program-name do not correspond.	Check with MX and re-enter

SN (assign a Serial Number to a tape)

This intrinsic allows the operator to assign a serial number to a magnetic tape. The intrinsic, in assigning the serial number, purges the tape in a manner similar to the PG intrinsic, giving the tape a label and a serial number. The tape may now be written to.

Railroad diagram:

SN < peripheral > < serial number > —|

where:

peripheral is either magnetic tape mnemonic (MTA, MTB, etc.) or cassette mnemonic (CTA, CTB, etc.)

serial number is one (1) to five (5) ASCII characters.

Ensure that the magnetic tape is write enabled, that is, has a write ring, or that the cassette is write enabled, that is, its tabs are set for write enable. The drive must be ready also.

Examples:

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

SN CTB 12345

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

SN MTA 345

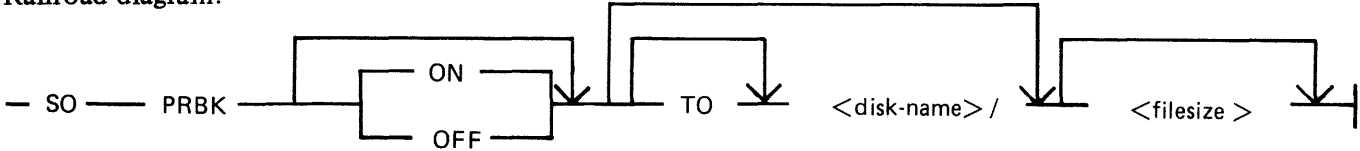
Output Messages:

Message	Possible Causes	Suggested Action
peripheral PURGED serial number	SN was successful.	None.
faulty input INVALID	Either peripheral or serial number not specified in message or Specified serial number has more than 5 characters or Specified peripheral invalid.	Re-input message.
faulty input NOT ON SYSTEM device status	The specified device is not connected to the system. The device status is the information which would be output if an OL of the device had been done. Either the device is in use or not ready.	Check input and device. Ensure that the media is write enabled. If the device is in use, use another drive.
faulty input FAILED – I/O ERROR ON peripheral	An I/O error occurred while writing the serial number or purge label.	Re-place media and try again.

SO (Set Printer Backup Option ON/OFF)

This intrinsic allows the operator to change any or all of the printer backup options.

Railroad diagram:



The following printer backup related options may be changed:

1. The ON/OFF option sets the system printer backup option to the requested state, allowing printer output to be directed to either 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 to which a printer backup file will be directed, unless otherwise specified by its creating program. This disk must be on the system. This disk name will remain in force until changed by the "SO PRBK" intrinsic or until 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 Backup file opens waiting on a no disk condition to be restarted. If no disk name is given, the printer backup designate disk 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 replaces the system default printer backup file size and 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

```
PRINTER BACKUP OPTION IS ON or OFF
PB DISK IS <disk-name>/, DEFFAULT MAX SIZE IS <file size>
```

is displayed.

Examples:

Example 1 To enable the system printer backup option, the following syntax is used:

```
SO PRBK ON
```

Example 2 To enable the system printer backup option and to specify the printer backup designate disk to be PBDISK,(which is on line and ready) also the default printer backup filesize of 300 records (to be used with programs that have a size of zero declared). The following syntax is used:

```
SO PRBK ON TO PBDISK/300
```

Example 3 To disable the system printer backup option the following is used:

```
SO PRBK OFF
```

-
- Example 4 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 is used:
SO PRBK OFF TO SYSTEM/
- Example 5 To set the default file size to 10000 records the following syntax is used:
SO PRBK 10000
- Example 6 To set the default printer backup designate disk to BACKUP (which is on line and ready), the following syntax is used:
SO PRBK BACKUP/

SO PRBK Error Messages:

Error Message	Description
<faulty input> INVALID – OPTION(S) NOT SPECIFIED CORRECTLY	The initiating message is not correct (see syntax for initiating message).
<faulty input> INVALID – PRINT MAX SIZE CANNOT BE ZERO	The file size cannot be zero.
<faulty input> INVALID – SPECIFIED PRINTER BACKUP DISK NOT ONLINE	The disk given in the initiating message is not present.

ST (Temporarily Suspend a Running Program)

This intrinsic places a temporary halt on a program that is running. The program still appears in the mix. The data needed to restart the program exactly where it 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 command must be used to restart the program.

Format diagram:



Examples:

To stop the program whose mix-number is 3:

```
ST 3
```

To stop the program PR020:

```
ST 3/PR020
```

Output Messages:

Message	Possible Causes	Suggested Action
mix-number-program name STOPPED	ST successful	None
mix-number/program name INVALID	Program has already been stopped or program is not in the mix.	Check with MX for status of program; (re-input if necessary).

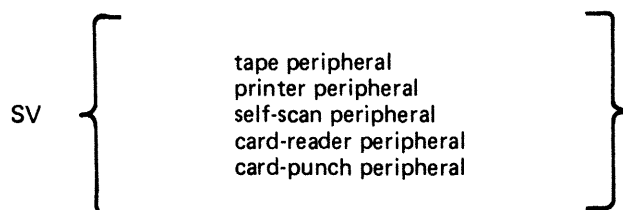
SV (Save Peripheral)

This intrinsic allows the operator to power off any input/output device “logically” (except disks, see PO intrinsic) in order to prevent their use by any program.

“Tape peripherals” include magnetic tape (MT) and cassette tape (CT).

“Printer peripherals” include line printer (LP) and serial printer (SP).

Format diagram:



Examples:

```

SV LPA
SV SSA

```

It is possible to “save” a device that is being used by a program. This will allow the program presently assigned to this device to continue using it, but will prevent any subsequent programs from using the device. For example:

```

SV LPA
LPA SAVED IN USED BY 06/PR060

```

A “saved” device may be made “ready” again with the RY command or by physically powering the unit off and on.

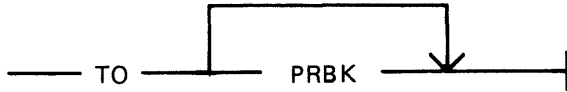
Output Messages:

Message	Possible Causes	Suggested Action
peripheral SAVED	SV successful	None
SV peripheral INVALID	Attempt was made to save a disk peripheral or device has already been saved.	Re-input (if necessary using correct peripheral).
SV peripheral NOT ON SYSTEM	Specified peripheral is not on-line to the computer.	Check input; re-input if necessary.
peripheral POWERED OFF	SV successful	None

TO (Test Printer Backup Option)

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

Railroad diagram:



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, then in addition to the above information, the following will also be output by the system:

PB DISK IS <printer backup designate disk name>/,
DEFAULT MAX SIZE IS <default filesize in records>

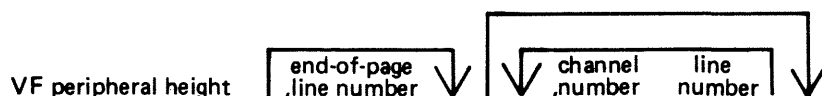
TO Error Messages:

Error Message	Description
<faulty input> INVALID – OPTION(S) NOT SPECIFIED CORRECTLY	The only option available is PRBK. Check input and re-enter.

VF (Vertical Format on Printer)

This command allows the operator to define the actions to be taken by the printer when certain vertical format commands are sent. This command applies only to printers which have soft vertical format control.

Format diagram:



The height field specifies the page height in lines. The channel number and the 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.

Example:

```
VF LPA 66, 60, 2 10
where page height = 66
      end of page = 60
      channel number = 2
      line number = 10
```

NOTE

For details, see LOAD.VFU utility.

Output Messages:

Message	Possible Causes	Suggested Action
VFU LOAD FAIL – ILLEGAL PARAMETER LIST	Typing error	Correct the input and re-enter
VFU LOAD FAIL – peripheral NOT ON LINE	The specified peripheral is not ready	RY the peripheral and re-enter
VFU LOAD FAIL – peripheral IN USE	The specified peripheral is in use by a program	Wait until program has closed the printer file, then re-input
VFU LOAD FAIL – peripheral HAS NO SOFT VFU	The peripheral is not a B 9249-30/50 line printer	None

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 and UPDATE cannot be run on a B 1800.

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:

- TRANSLATOR
- 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
- ARCS – Automatic Run Control System
- RB – Remove Backup files

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:

Railroad diagram:

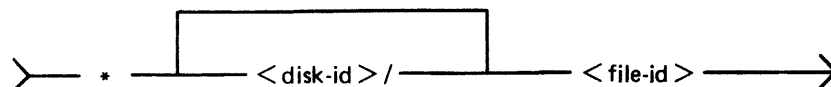


Table 4-1. Peripherals Required by CMS-Common Utilities

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							
PB *	O	R	O		O							O
PD		R										
PL *		R	O		O							
RB	R	R										
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										

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.)

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, a “<file-name> NOT FOUND” 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

Filetype	(continued)	Description
14-16		Interpreter/SORTINTRINS/MCPX/SAU
21		SYSLANGUAGE local language
22		SYSCONFIG – a required file
30		Virtual Memory/Dump
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
AX <mix-no>/RM ACPT
```

Then, to remove a system file:

```
AX <mix-no> <file-name> 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 12/PD or 12/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 “<64> LOAD FAILURE UTILITY BUSY” is returned.

SYSLANGUAGE

All CMS utilities which output SPO messages, with the exception of PATCHMAKER, GEN-DUMPFL, RB 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:

```
“INVALID DICTIONARY ENTRY <entry-no>”
```

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 SYSFILEUTL 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.

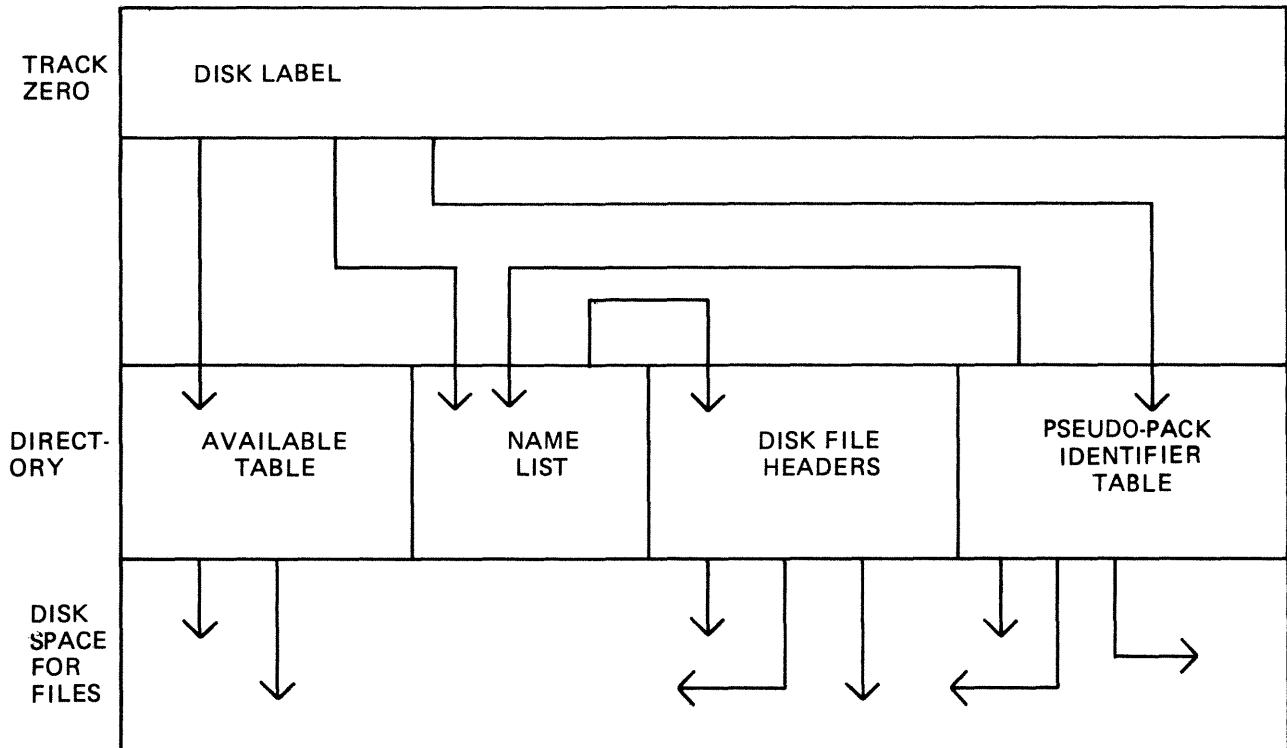


Figure 4-1. Fixed Disk Directory Structure

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.

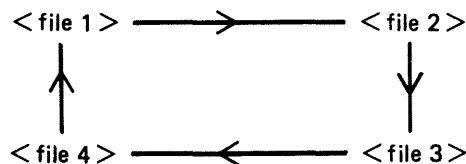
When searching for a file on an unrestricted pseudo-pack, the PPIT and file directories on all fixed disk units are searched, otherwise, only the PPIT and file directory on the designated unit are searched.

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.

LOGGING IN THE 3.04 RELEASE

Logging in the 3.04 release functions in the following manner.

Four log files termed primary log files are used by the MCP to store maintenance entries and, if requested, System message entries. These files are used in the method shown in the following diagram:



This form of file usage is called CYCLIC.

To preserve the maintenance entries, they are copied from a primary log file when the MCP has filled the primary log file and moved on to the next primary log file in the chain. The maintenance entries are transferred into log file(s) termed secondary log file(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 System message logging has been requested in NON-CYCLIC System message logging, the System message entries will be handled in a similar manner to the maintenance entries.

There are two forms of System message logging:

1. CYCLIC System message logging. This is when the MCP places System messages in the primary log files but the System messages are not transferred to NON-CYCLIC secondary log files.
2. NON-CYCLIC System message logging. This is the same as CYCLIC System message logging except the System messages are transferred to NON-CYCLIC secondary log files.

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

Types Of Log File

Primary:

SYS-LOG-01	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
SYS-LOG-02	
SYS-LOG-03	

SYS-LOG-04 messages (if requested to in CONFIGURER <SYSCONFIG>) are written to these files in a cyclic nature 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-HLDjjjhh 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 the remaining System message entries from the primary log files at warmstart which have not had their maintenance entries transferred by TL in the previous session. Used in cyclic System message logging.

Where: jjj is the Julian date and hh and mm are progression numbers in the range 00-99.

These secondary files are created and maintained by TL. Both SYS-HLDjjjhh 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. TL continues 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 when SPO logging has been requested when running CONFIGURER that System message entries are placed in the primary log files by the MCP. If the console screen is chosen as the SPO, System messages will be logged. Maintenance only logging functions similarly to CYCLIC System message logging.

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 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-HLDjjjhh 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-MLGjjjmm. TL also transfers all the System messages from this primary log file(s) into the current SYS-HLDjjjhh.

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-HLDjjjhh. Therefore, System messages are preserved along with maintenance entries in their respective secondary log files, SYS-HLDjjjhh and SYS-MLGjjjmm.

Utilities Linked With Logging

CONFIGURER 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(s) (that is, SYS-HLDjjjhh).
 5. The size of the secondary maintenance entry file(s) (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.

PL	Permits the printing of a complete log file or a portion of a log file within 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 writes maintenance entries and, if requested, System messages into the primary log files.
TL	Creates and maintains secondary log files when required (TL is ZIPPed by the MCP automatically).
WL	Displays the current state of the primary log files and the active secondary log files.
IR	Initiates recall of the entries in the current primary log file to the SPO device, the use of LF (look forward) and LB (look back) intrinsics. IR is terminated when any other SYS-SUPERUTL function is called.
LS	Stops logging, that is, the MCP stops writing System message entries into the primary log files.
LG	Starts logging, that is, the MCP begins writing System message entries to the primary log files.

Suggested Logging Method

The following method 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 is required to be switched ON.
 2. NON-CYCLIC logging is required.
 3. The sizes of the primary and secondary log files will have to be determined with 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:

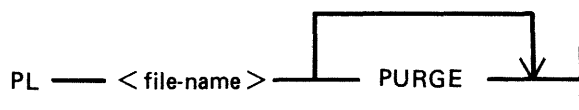
PD SYS-HLD = , SYS-MLG =

NOTE

The file(s) with the family prefix SYS-HLD contains 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:

Railroad diagram:



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 carried out during every warmstart with the PURGE option in the initiating message of PL (for all the SYS-HLDjjhh and SYS-MLGjjmm), the user should have only the current session's log files on the system.

Log File Sizes

The size of log files required depends on several factors.

In order to minimize the operator overhead 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 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 times TL will be zipped to transfer entries to the secondary files. TL is fairly heavy on system resources and its effect,

though temporary, will be noticeable, particularly with terminal 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 size of the primary log files can be made large enough to ensure that the primary log files are never filled. In this way, TL is only zipped at warmstart and no interruption to the system occurs during normal use.

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

$$\text{TOT.size} = \text{TOT.msgs.1} + \frac{(\text{TOT.msgs.2} \times \frac{\text{AVG.msgs.2}}{82})}{4}$$

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% to allow for fluctuations 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 system 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 DFH 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

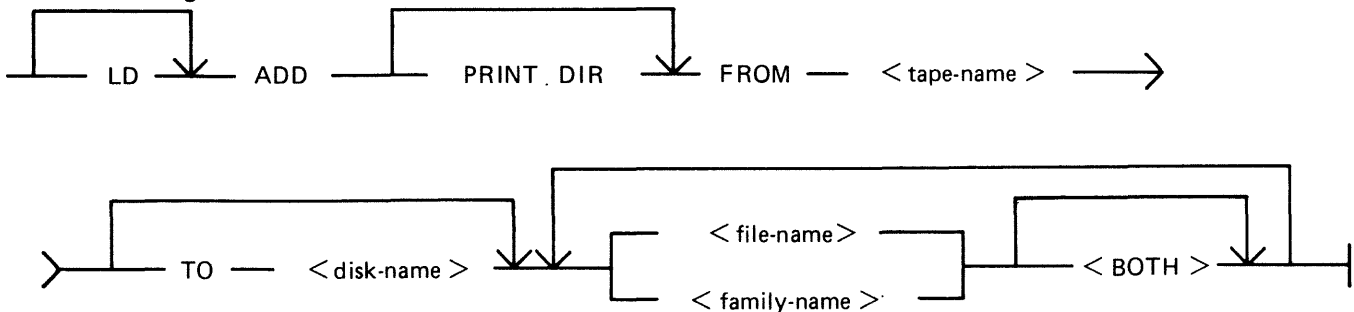
Message	Possible Causes	Suggested Action
INVALID CHARACTER IN IDENTIFIER	Disk name or file name contains character(s) not permitted by the system. Valid characters are: A-Z, 0-9, . (dot), - (dash).	Check input and re-input if necessary.
ILLEGAL PARAMETER LIST	Typing error.	Check input. After words "ILLEGAL PARAMETER LIST" system will display portion of input message that contains error.
file-name NOT FOUND or file-name NOT ON LINE	Specified file name is not on disk.	Check input and re-enter if necessary; check for correct disk; supply specified file (COPY file from backup medium or create file).
NO SPECIFICATIONS GIVEN	Input message is incomplete.	Check input and re-enter.
DISK disk-name NOT OPENED NOT ON LINE or DISK disk-name NOT AVAILABLE or DISK disk-name FOR XD NOT AVAILABLE	Specified disk is not on-line to computer.	Check input and re-enter if necessary; check for correct disk; ready disk.
FILE LIST MAY BE INCOMPLETE	A group of files on an unrestricted pseudo-disk was requested, but the disk is off-line.	
PSEUDO-DISK pseudo-disk-name ON DISK disk-name NOT AVAILABLE	The pseudo-disk specified has not been found.	
DISK disk-name is a PSEUDO-DISK	The disk specified is not a physical disk but a pseudo-disk: probably the input disk name is incorrect. This message is printed by those utilities which cannot handle pseudo-disks.	Correct the disk-name and re-enter.
SPECIFICATION TOO LONG	The initiating message is too long.	Check input and re-enter.
DISK disk-name LOCKED or SYSTEM DISK IS LOCKED	Specified disk is being used by another utility/program.	Re-enter when disk becomes available.

Message	(continued) Possible Causes	Suggested Action
file-name NOT AVAILABLE – DISK LOCKED or file- name NOT AVAILABLE – IN USE	Specified file is being used by another utility/program.	Re-enter when disk becomes available.
file-name IS NOT TYPE SOURCE OR DATA	The file requested must be file type source or data @00@ – @0F@	Check file and re-enter.

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.

Railroad diagram:



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 “,” or a space “ ”.

<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

Message	Description
<file-name> NOT A RECOGNIZED DUMP TAPE	A tape which has been provided for loading does not have a recognizable format for the LD utility. The utility will go to End of Job. Re-execute the utility with a valid LD format tape.
NO FILE <file-name> ON TAPE <tape-name> FOR ADD	This file is not present on the tape. The utility will continue looking for other file(s) to load, if specified.
NO FILES IN THE FAMILY <family-name> ON TAPE <tape-name> FOR ADD	No files have been found by ADD in the family <family-name>. The utility will continue looking for other files to load, if specified.
NO FILES TO LOAD	The utility has found that none of the files specified will be loaded. The utility goes to End of Job. Check that the tape has the required file(s).
<file-name> LOAD DISCREPANCY	The end of file has been reached sooner than expected, and implies that there is erroneous information in the Disk File Header for that file. The utility will continue trying to load the rest of the file(s) specified. Try loading this file using another tape unit.
<file-name> NOT LOADED - ALREADY ON DISK	This file on the tape is not loaded because a file with the same name already exists on the disk. The utility will continue trying to load the rest of the file(s) specified.
ALTHOUGH WITH DIFFERENT ATTRIBUTES	The file on the tape differs in record size, file size or file type from that on the disk. This is for information only.
<file-name> NOT FOUND	The file containing the <parameter-list> (star file) is not present. The utility goes to End of Job. Make the star file available or specify the parameters in the initiating message.
DISK <disk-name> NOT AVAILABLE or with a pseudo-disk arrangement PSEUDO-DISK 'pseudo-disk-id' ON DISK	The specified disk is either not on the system or not ready. The utility goes to End of Job. Ensure that the required disk is on the system

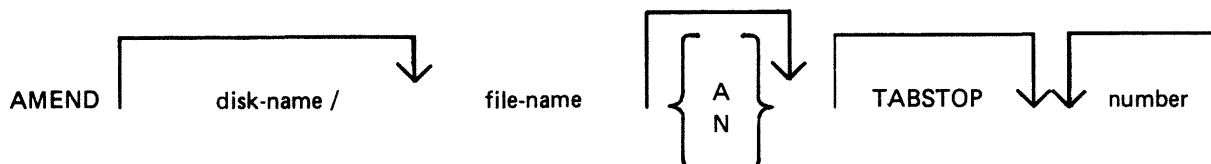
(continued)

Message	Description
<disk-name> NOT AVAILABLE and if the pseudo-disk is unrestricted, then FILE LIST FOR <family-name> MAY BE INCOMPLETE or FILE LIST MAY BE INCOMPLETE	and ready before re-running the utility.
<file-name> - DATA FILE NOT FOUND ON TAPE FOR LOAD	The data file for the given key file does not follow on the tape. The data file is not loaded. Note that the key file has been loaded. Check that the data file is on the tape. Reload the data file and SORT it to produce the correct keyfile.
LOAD ABANDONED - ERROR IN DUMP TAPE SHOULD HAVE BEEN PURGED	The tape should have been purged at time. All the files loaded so far will remain, except the file that the utility was attempting to load when the tape error is encountered. The utility goes to End of Job.
INVALID CHARACTER IN IDENTIFIER <file-name> - WARNING	The utility has generated a file name which contains invalid characters. The utility will continue.
<file-name> LOADED	This file has been loaded onto the specified disk.
FILE NOT LOADED - NOT ON TAPE <reason> AT DUMP TIME where <reason> can be: WAS IN USE or HAD BEEN REMOVED or HAD BEEN ALTERED or WAS INCOMPLETE or REASON UNKNOWN or DISK WAS LOCKED In this case, the additional message below will be displayed	The requested file could not be loaded as it was not dumped. The reason for this is explained in the message. If the reason was that the disk was LOCKED, that is, unavailable to LD, then the utility will go to End of Job as there are no further files on the tape. For any other reason, the utility will continue trying to load any further requested file(s).
LOAD ABANDONED - ERROR IN DUMP DUPLICATE - <file-name> ALREADY BEING LOADED	More than one request has been made to load this file in the initiating message. The utility will continue. Note this is usually caused by specifying individual files and the family group containing the files.

AMEND (Disk File Amending)

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 diagram:



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).

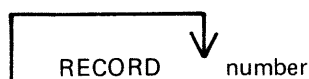
PK1	PK2	PK3	PK4	PK5	PK6
NEXT	MODIFY	SELECT	—	—	END

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:

Format diagram:



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 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 can be specified in the console input.

Examples:

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 : and terminate with OCK1
```

resulting in “20 ABCDZZZZIJKLMNOPQRST”

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

```
Z : XXXXXX : and terminate with 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	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE
2	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE
3	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE
4	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE
5	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE
6	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE
7	AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE

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 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDDEEEEE

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

BBBBB:XXXXX: and terminate with OCK1

AMEND now displays the AMENDED record:

1 AAAAABBBBBXXXXDDDDD AAAAABBBBBCCCCDDDDDEEEEEE

Press PK1.

2 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE
:XXXXX:11 (OCK1)

2 AAAAABBBBBXXXXDDDDD AAAAABBBBBCCCCDDDDDEEEEEE

Press PK 1

3 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE
BBBBB:XXXXX:2 (OCK1)

3 AAAAABBBBBXXXXDDDDD AAAAABBBBBCCCCDDDDDEEEEEE

Press PK1

4 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE
BBBBB:XXXXX:26 (OCK1)

4 AAAAABBBBBCCCCDDDDD AAAAABBBBBXXXXDDDDDEEEEEE

Press PK1

5 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE
:XXXXX:31 (OCK1)

5 AAAAABBBBBCCCCDDDDD AAAAABBBBBXXXXDDDDDEEEEEE

Press PK1

6 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE
BBBBB:XXXXX:7 (OCK1)

6 AAAAABBBBBCCCCDDDDD AAAAABBBBBXXXXDDDDDEEEEEE

?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 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE

Enter the amendment followed by OCK2:

BBBBB:XXXXX: (OCK2)

1 AAAAABBBBBXXXXXCCCC DDDDDAAAAABBBBBCCCCDDDDD

Press PK1.

2 AAAAABBBBBCCCCDDDDD AAAAABBBBBCCCCDDDDDEEEEEE
:XXXXX:11 (OCK2)

2 AAAAABBBBBXXXXXCCCC DDDDDAAAAABBBBBCCCCDDDDD

Press PK1

3 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEEE
BBBBB:XXXXX:2 (OCK2)

3 AAAAABBBBBXXXXXCCCCC DDDDDAAAAABBBBBCCCCCDDDDD

Press PK1

4 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEEE
BBBBB:XXXXX:26 (OCK2)

4 AAAAABBBBBCCCCCDDDDD AAAAABBBBBXXXXXCCCCCDDDDD

Press PK1

5 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEEE
:XXXXX:31 (OCK2)

5 AAAAABBBBBCCCCCDDDDD AAAAABBBBBXXXXXCCCCCDDDDD

Press PK1

6 AAAAABBBBBCCCCCDDDDD AAAAABBBBBCCCCCDDDDDEEEEEE
BBBBB:XXXXX:7 (OCK2)

6 AAAAABBBBBCCCCCDDDDD AAAAABBBBBXXXXXCCCCCDDDDD

?END AMEND

Output Messages

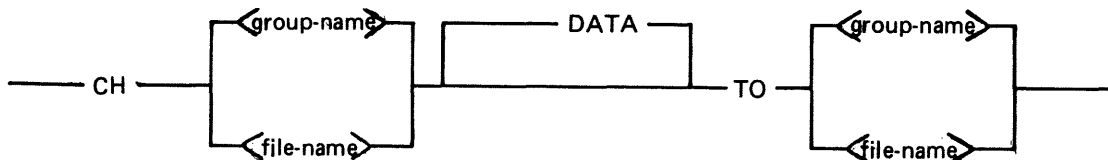
Refer to the section on the CREATE utility for output messages.

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).

Railroad diagram:



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 or groups of 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
"file name" CHANGED TO "file name"	File name successfully changed.	None.
"file name" NOT CHANGED - NOT FOUND	Specified file name is not on disk.	Check input and re-input if necessary; check for correct disk.

Message	(continued) Possible Causes	Suggested Action
NO FILES FOUND FOR CHANGING IN THE FAMILY "group-name"	Specified group name is not on disk.	Check input (re-input if necessary). Check for correct disk.
"file name" NOT CHANGED IN USE	File's name cannot be changed because it is currently being used by system.	Wait until file is no longer in use, then re-input.
"name" FILE IDENTIFIER TOO LONG	Attempt has been made to change a file name to more than 12 characters in length.	Re-input.
"file name" NOT CHANGED - ILLEGAL REQUEST	Attempt has been made to change the name of a file to "SYSMEM" (a name reserved for system use) or all spaces.	Re-input.
"file name" NOT CHANGED - "file name" ALREADY ON DISK	Attempt has been made to duplicate the name of a file already on disk.	Re-input.
KEYFILE "file name" NOW POINTS TO DATA FILE "file name".	Successful completion of data file name change.	None.
<64> LOAD FAILURE UTILITY BUSY	Another function of SYS-SUPERUTL is being executed.	Wait until SYS-SUPERUTL is free, then re-input.

CHECKADUMP (Compare Library Tape with Disk)

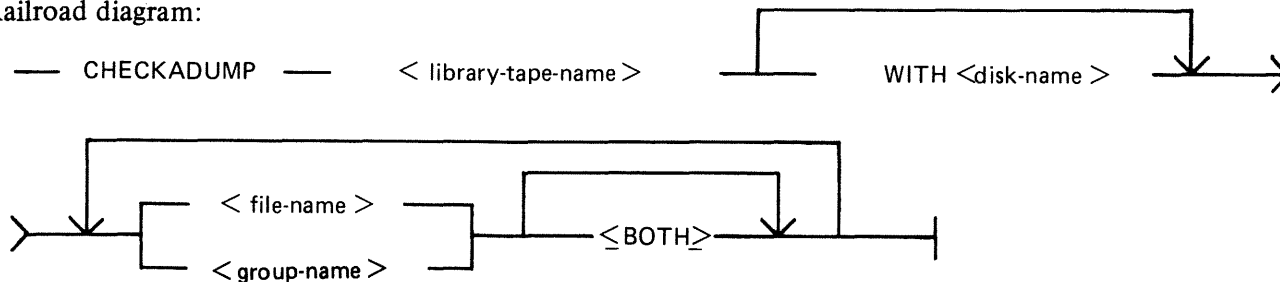
This utility allows the operator to compare information in files on a library tape with corresponding files on disk. It is used to verify that a library tape is correct after files have been DUMPed, or that disk files are correct after files have been ADDED or LOADED. Specified tape is processed sequentially, 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:

Railroad diagram:



<library-tape-name> This is the name of the library tape.

WITH <disk-name> This is the name of the disk on the files to be compared reside.

<file-name> These are the files to be compared. If the option "<BOTH>" follows a <group-name> 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
FILE <file-name> NOT COMPARED - NOT ON TAPE - <reason> where <reason> can be: WAS IN USE AT DUMP TIME HAD BEEN REMOVED AT DUMP TIME HAD BEEN ALTERED AT DUMP TIME DISK WAS LOCKED AT DUMP TIME WAS INCOMPLETE AT DUMP TIME REASON UNKNOWN	The file was not dumped for the reason given.	None. Utility will continue.
library-tape-name NOT A RECOGNIZED DUMP TAPE	First record of tape not recognized by CHECKADUMP. Tape may not have been created by LD utility.	None. Utility ends.
library-tape-name INVALID DIRECTORY ON TAPE	More or fewer entries in directory on tape than specified in the first record of tape.	None. See CMS MCP Reference Manual for additional information on tape formats.
COMPARISON ERROR ON library-tape-name ON DISK FILE HEADERS	Header in body of tape is not identical to respective header in disk directory. The error count for the file is increased by 1.	Recreate dump tape.
COMPARISON ERROR ON file-name FILE NOT FOUND FOR CHECK	Corresponding disk file cannot be found for file on tape.	Recreate dump tape.
DISK EXPIRING BACKUP RECOMMENDED	Self explanatory.	Backup disk.
COMPARISON ERROR ON file-name FILE NOT AVAILABLE FOR CHECK	Corresponding disk file cannot be read for this file on tape.	Recreate dump tape.
COMPARISON ERROR ON file-name (AROUND RECORD number)	Discrepancy between disk file and tape file. Record number in vicinity of error in file is printed, if possible. One is added to error count for this file.	Recreate dump tape.
COMPARISON ERROR ON file-name AROUND END OF FILE	Difference in length of tape and disk files. One is added to error count for that file.	Recreate dump tape.

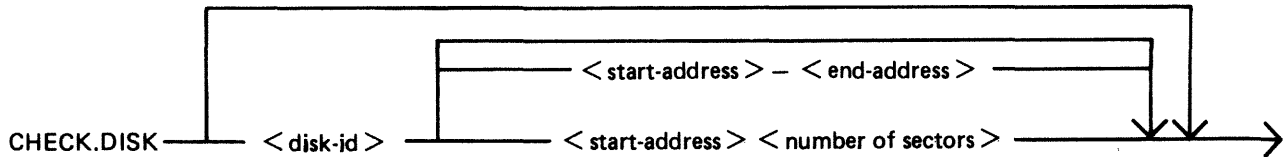
Message	(continued) Possible Causes	Suggested Action
COMPARISON ERROR ON file-name DIFFERING FILE SIZES	Difference in sizes of disk and tape files.	Recreate dump tape.
COMPARISON ERROR ON file-name DIFFERING FILE TYPES	Difference in file types of files being compared.	Recreate dump tape.
COMPARISON ERROR ON file-name DIFFERING RECORD SIZES	Difference in record sizes of the files being compared.	Recreate dump tape.
COMPARISON ERROR ON file-name DIFFERING BLOCK SIZES	Difference in block sizes of files being compared.	Recreate dump tape.
NO DISCREPANCIES BETWEEN DUMP TAPE library-tape-name AND DISK disk-name	CHECKADUMP successful.	None.
DISCREPANCIES FOUND BETWEEN DUMP TAPE library-tape-name AND DISK disk-name	Discrepancy discovered between disk file and tape file.	Recreate dump tape.
COMPARISON ERROR ON filename DIFFERING GENERATION NUMBERS	Mismatch in generation numbers of files being compared. That is, the file has been modified.	Recreate dump tape.

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.

Railroad diagram:



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 ERRORS” 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 utility determines whether a sector with a read error is denoted as BAD in the directory available table and displays these messages on encountering such an error:

```
ERROR NOTIFIED ON READING DISK disk-name  
followed by SECTOR < address > DENOTED AS BAD IN DISK DIRECTORY  
or SECTOR < address > NOT DENOTED AS BAD IN DISK DIRECTORY
```

If one or more inconsistent errors occur, the message:

```
“INCONSISTENT ERROR(S) NOTIFIED – POSSIBLE MEDIUM/DRIVE FAULT”
```

is displayed on completion of a disk or disk area check.

If no consistent errors are found, but one or more inconsistent errors occur, then “NO CONSISTENT ERRORS” is displayed with the above message.

If one or more consistent errors are encountered, one of the following two messages is displayed on completion of a disk or disk area check:

ONE CONSISTENT ERROR NOTIFIED
or
<n> CONSISTENT ERRORS NOTIFIED

If no errors – consistent or inconsistent – occur, the message “NO ERRORS” is displayed.

On completion of checking an area or the whole disk, if any read errors were encountered on sectors which are not denoted as BAD in the available table, the message:

nnn BAD SECTOR(S) NOT DENOTED IN DIRECTORY

is displayed following the summary and error count messages.

If the utility detects that it is being run on 1 megabyte floppy disk, it will give an additional message at End of Job if circumstances dictate, as follows:

If 15-30 bad sectors are found, the following message is displayed:

DISK disk-id SHOULD BE REINITIALISED SOON

If more than 31 bad sectors are found, the following message is displayed:

DISK disk-id EXCEEDS BAD SECTOR LIMIT
PLEASE POWER OFF DISK disk-id

If this message is given, the disk should not be used again.

The following FETCH VALUES are returned by CHECK.DISK for an abnormal termination:

FETCH VALUE	Description of Cause
2	Illegal parameters found.
32	A non-fatal parameter error found: for example, Sector Range exceeding maximum disk address.

Output Messages

Message	Possible Causes	Suggested Action
CHECK.DISK ON disk-name COMPLETED – ONE CONSISTENT ERROR NOTIFIED	This message is displayed after all sectors have been read and one error has been detected.	None.
CHECK.DISK ON disk-name COMPLETED – number ERRORS NOTIFIED	The utility has found more than one bad sector.	None.
READ ERROR ON DCL OF disk-name	An error was encountered while reading sector zero of the specified disk.	None.
ERROR NOTIFIED ON READING SECTOR n	Normal output message. Utility continues.	None.

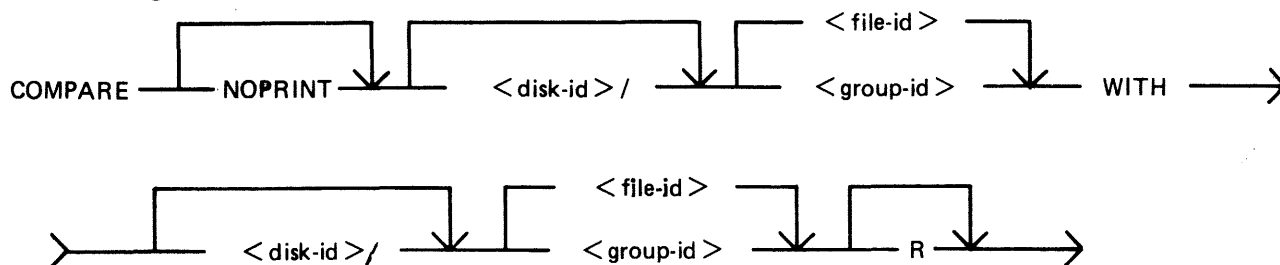
Message	(continued) Possible Causes	Suggested Action
CHECK.DISK ON disk-name COMPLETED – NO ERRORS	Normal EOJ messages	None.
CHECK.DISK ON disk-name SECTOR address – NO ERROR	Only one sector specified for checking and found to be readable.	None.
CHECK.DISK ON disk-name AREA address – address COMPLETED	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 messages.

COMPARE (Compare Files)

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

Railroad diagram:



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 -

file-name-1 RECORD record-number AT BYTE offset
file-name-2 RECORD record-number AT BYTE offset

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

1. WARNING - <file-name1> FILETYPE IS <hex. filetype>
<file-name2> FILETYPE IS <hex. filetype>

if the filetypes of the two files are different.

2. WARNING - <file-name1> RECORD SIZE IS <rec. size>,
<file-name2> RECORD SIZE IS <rec. size>
- ONLY <x> BYTES WILL BE COMPARED.

if the default or specified record sizes of the two files are different. Only the first x bytes of the records, where x is the smaller of the two record sizes, will be compared.

3. WARNING - COMPARISON KEY LENGTHS FOR <file-name1> AND <file-name2>
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

If corresponding records are different, the following is printed on a line printer file (or console printer if the line printer is not available).

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-name1> 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 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 <file-name2> 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 differs 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 corresponds 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)

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,

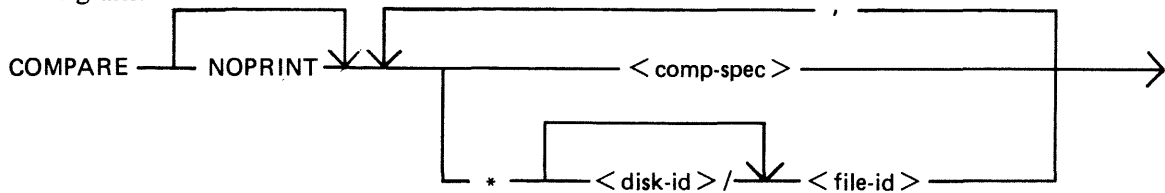
record E is compared with records N, O, P and Q. If none of these compare, the comparison is terminated.

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

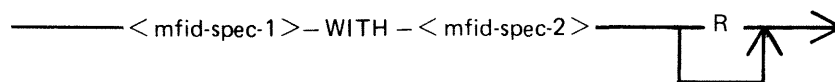
Further features in this utility are summarized in the railroad chart given in figure 4-2, which gives the complete input specifications.

Railroad diagram:



< comp-spec > is defined as :

Railroad diagram:



< mfid-spec > is defined as :

Figure 4-2. Railroad Chart for COMPARE Utility

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.

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 files.

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 will be the comparison. (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

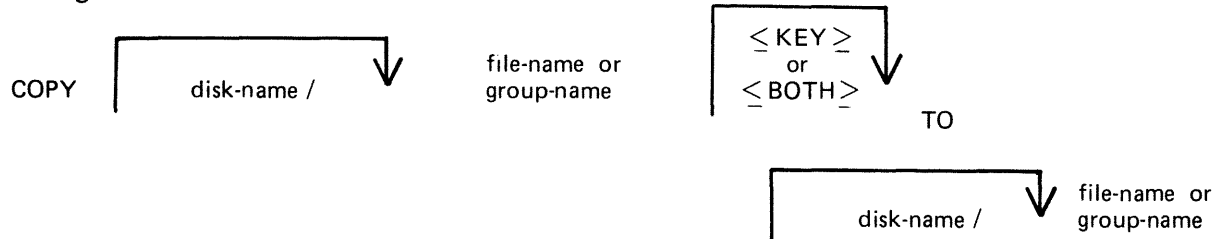
Message	Possible Causes	Suggested Action
END OF FILE filename BEFORE filename - n ERRORS	End of one file is detected before the end of the other file.	None.
filename WITH filename COMPARED - n ERRORS	Normal ending message if both files are the same size.	None.
ILLEGAL SYNTAX FOR ITEM input	Initial input mistyped.	Check input and re-enter.
CANNOT REALIGN filename WITH filename - n ERRORS	Ending message if realignment has been specified but has failed.	None.
INCOMPATIBLE FILESPECS input	A file is specified to be compared with a group of files, or vice versa.	Re-input correct message.
ITEM TOO LONG input	Input message greater than 400 characters.	Divide input into separate parts, and re-enter.
DIFFERENCE DETECTED AT BYTE @nnnn@	See example earlier.	See example earlier.
CANNOT READ RECORD n OF filename	Parity error on disk file.	Use backup copy of file concerned, if possible.

Message	(continued) Possible Causes	Suggested Action
ILLEGAL KEY FOR filename: number-number	A key has been specified with a length zero or which does not lie completely within the record. The utility will proceed with next item.	Check the key length and re-input if necessary.
ILLEGAL RECORDS FOR filename: number	The record number specified for the starting point of comparisons is not present in the file.	None.
CANNOT COMPARE PAST POSITION number IN filename	The utility limitation came into effect due to a request to compare beyond an offset of number bytes.	None.
filename EXHAUSTED AT number	The file had a record range specified which ran beyond the end of file. The range has been truncated.	None.

COPY (File Copy)

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

Format diagram:



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 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 PR200 <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.

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. Abbreviation 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:

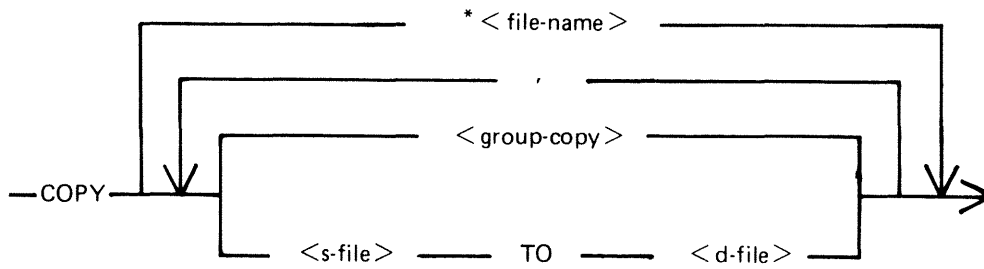
```
COPY PRFILE CRD TO PTFILE PTP
```

NOTE

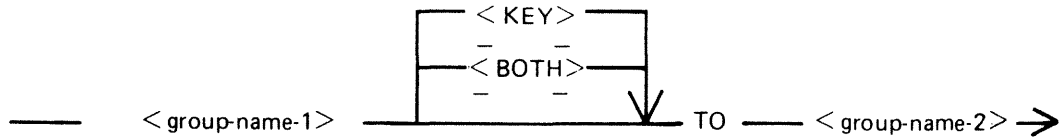
Paper tapes are always “unlabelled”, and when accessing it, MCP will issue an appropriate message requiring an “AD” intrinsic response from operator. See “AD” intrinsic.

Unlabelled Tapes

Input tapes having no CMS labels (“unlabelled” tapes) may be accessed by the COPY utility.



<group-copy> is defined as :



<s-file> is defined as :

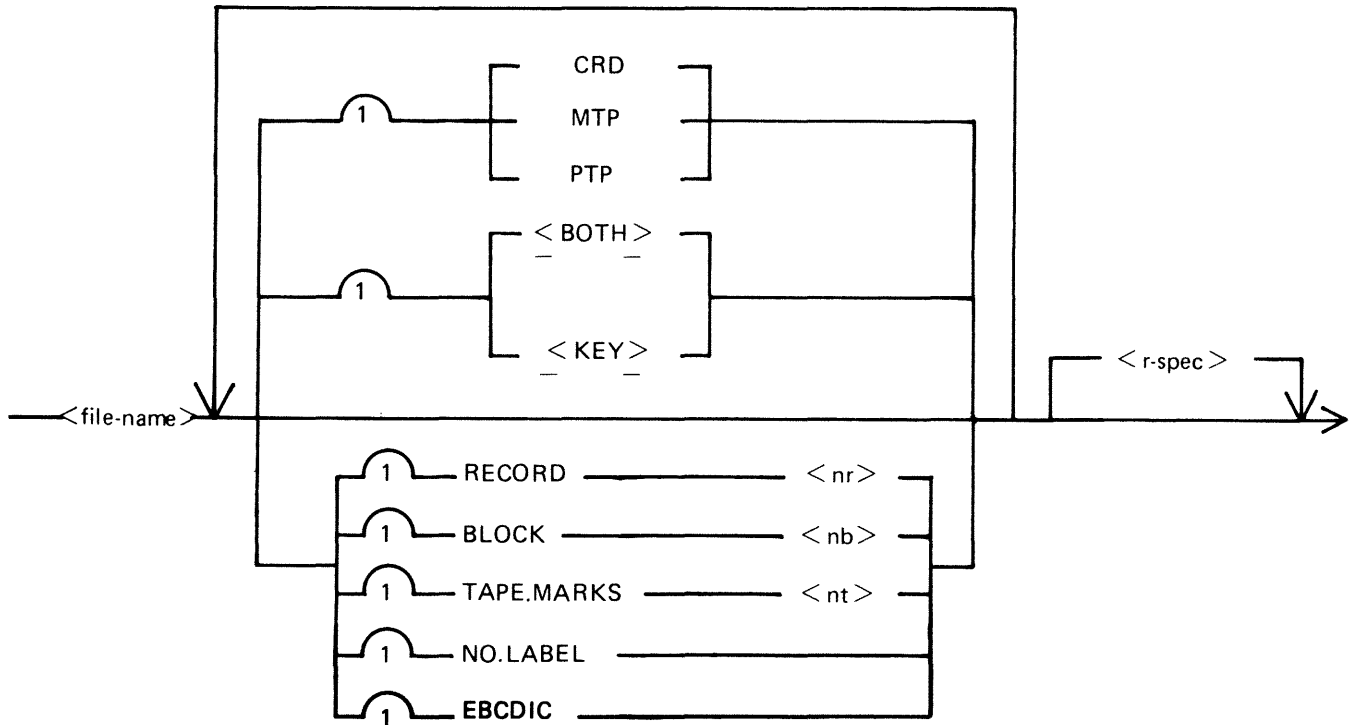
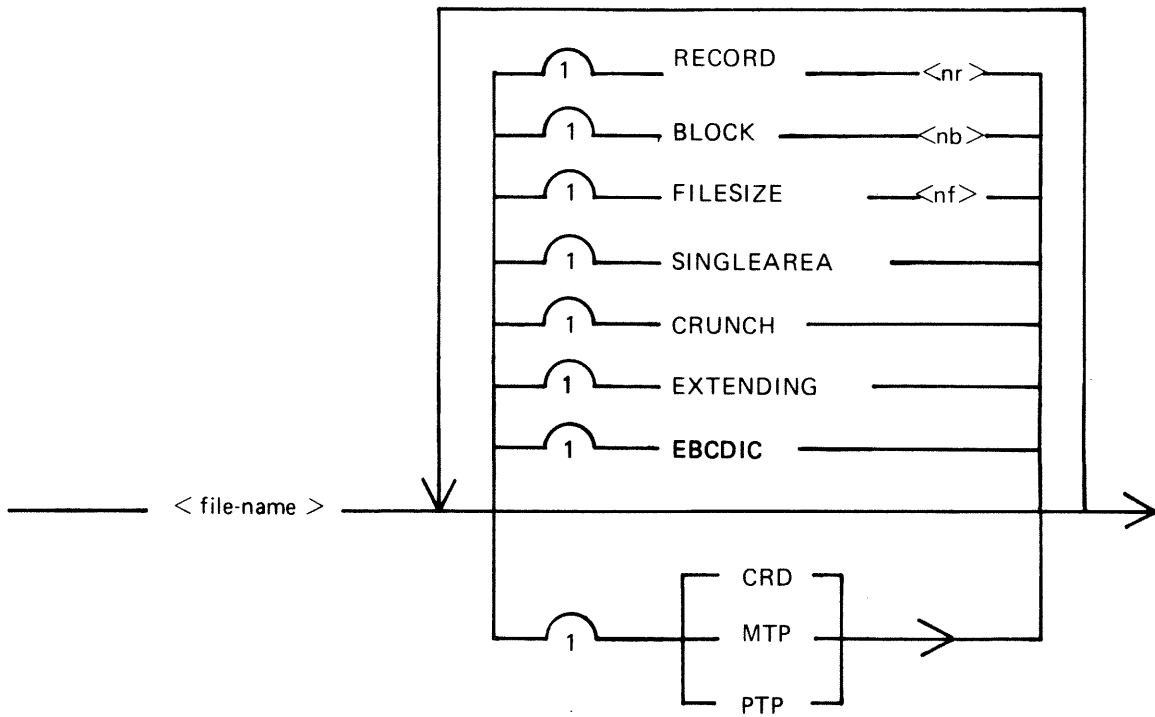


Figure 4-3. Railroad Chart for Copy Utility (Sheet 1 of 2)

< d-file > is defined as :



< r-spec > is defined as :

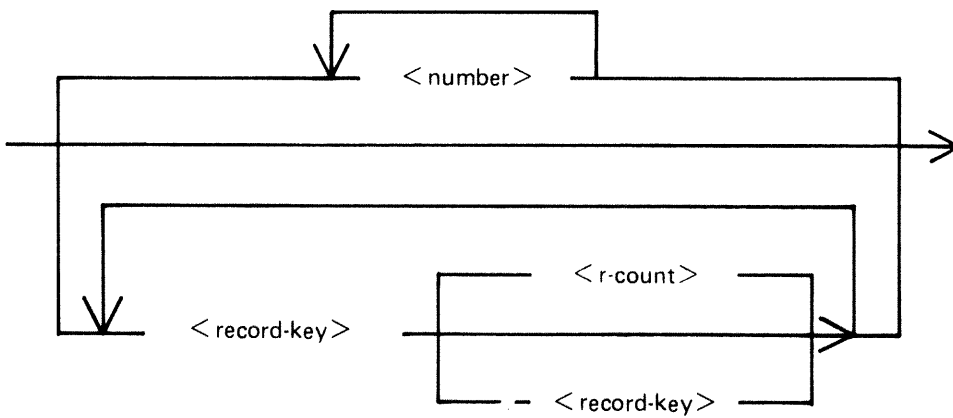


Figure 4-3. Railroad Chart for Copy Utility (Sheet 2 of 2)

The NO.LABEL option allows the copying of unlabelled files. Upon 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 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.

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.

Example:

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).

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 1000 TO FILE2
```

Selected Index File Copy

For indexed files, copying of records can be selected based on content of the key. There are 2 options: the number of records can be specified, or an ending key value.

Examples:

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

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

Using the same keyfile, to copy all data records from personnel #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 # in this case).

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

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

Save Factor

New magnetic tapes are given a save factor of 999.

2015228

Output Messages

Message	Possible Causes	Suggested Action
number, number IN file name NOT COPIED	Error found in pair of numbers for selected record copy option. One record number in the pair indicates a section of the file address than a previously specified section. This pair is ignored by COPY.	Check input and re-enter.
file-name EXHAUSTED	End-of-file was encountered while the section of the file indicated was being copied.	None
filename TO filename BAD ATTRIBUTES	Particular attribute specifications are meaningless or inconsistent. The inconsistencies will be in relationship between output device, record size, and block size; or attempt was made to add records to a file whose attributes differ from those of input file.	Check input and re-enter.
FILE-IDENTIFIER file-name TOO LONG	Copy of a group of files was requested. At least one file-name that was to be generated by the copy would have been more than 12 characters long, maximum length of file-names is 12 characters.	Check input and re-enter.
file-name TO file-name COPY DISCREPANCY	End-of-file reached before expected. Disk File Header may be corrupt.	
file-name TO file-name COPIED	COPY successful.	None
FAMILY family-name-a TO family-name-b COPIED	COPY successful.	None
file-name TO file-name ILLEGAL REQUEST	Copy of a group of files was requested. An individual file-name of "SYSMEM" or "(spaces)" was to have been produced. "SYSMEM" (a file-name reserved for system use only) and "(spaces)" are not acceptable file-names.	Check input and re-enter.
file-name NOT ACCEPTABLE RECORD SIZE OF m EXCEEDS MAXIMUM FOR THIS RUN	Copy has encountered a file with a record size greater than expected. This can happen if a magnetic tape file	Check input and re-enter.

Message	(continued) Possible Causes	Suggested Action
– RESUBMIT	with a record size greater than 1024 characters is submitted to the utility without the record size being properly specified in the initial input.	
file-name EXHAUSTED DURING RANGE record-key number number	End-of-file encountered while section of file indicated by “record-key” “number” is being copied.	None
file-name EXHAUSTED DURING RANGE “record- key” “record-key”	No records were found in the range “record-key” “record-key”.	Check input and re-enter.
NO RECORDS FOR COPYING FROM file-name	Specified file contains no records for copying.	Check for correct file-name.
INPUT RECORD number OUTPUT RECORD number PERMANENT ERROR ON OUTPUT FILE	Error encountered utility cannot continue.	Do selective copies of the parts of the file before and after the bad record, as part recovery.
INPUT RECORD number OUTPUT RECORD number PERMANENT ERROR ON OUTPUT FILE	Error encountered utility cannot continue.	Remove the disk area in error with XD utility and re-run the COPY.
INPUT RECORD number OUTPUT RECORD number OUTPUT FILE TOO SMALL.	Error encountered – utility cannot continue.	Re-input the COPY with suitable FILESIZE option.
file-name TO file-name COPY FAILURE	COPY unsuccessful irrecoverable error was encountered. The reason should be displayed before this message.	None.
NO RECORDS IN THE KEYFILE – file-name	<BOTH> option was used. Utility was not able to access to a data file through some failure in the keyfile.	None.
file-name NOT FOUND FOR EXTENDING	Specified file was not found.	Check input and re-enter if necessary; check for correct medium.
file-name REMOVED	COPY successful. If any duplicate files are encountered by COPY they are automatically removed.	None.
file-name TO file-name SELECTION CRITERIA IGNORED	Confirms that selection criteria were specified when copying a pair of files, and have been discarded.	

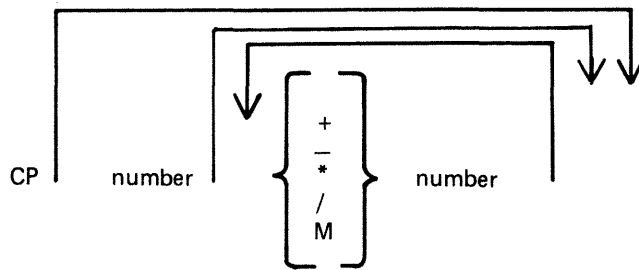
Message	(continued) Possible Causes	Suggested Action
BAD ATTRIBUTES SPECIFIED file-name TO file-name EXTENDING FLAG IGNORED.	Inconsistent attributes were specified for input file. EXTENDING option was ignored by utility when pair of files was copied.	Check input and re-enter.

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 diagram:



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

0 99999999999999999999999999999999[@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

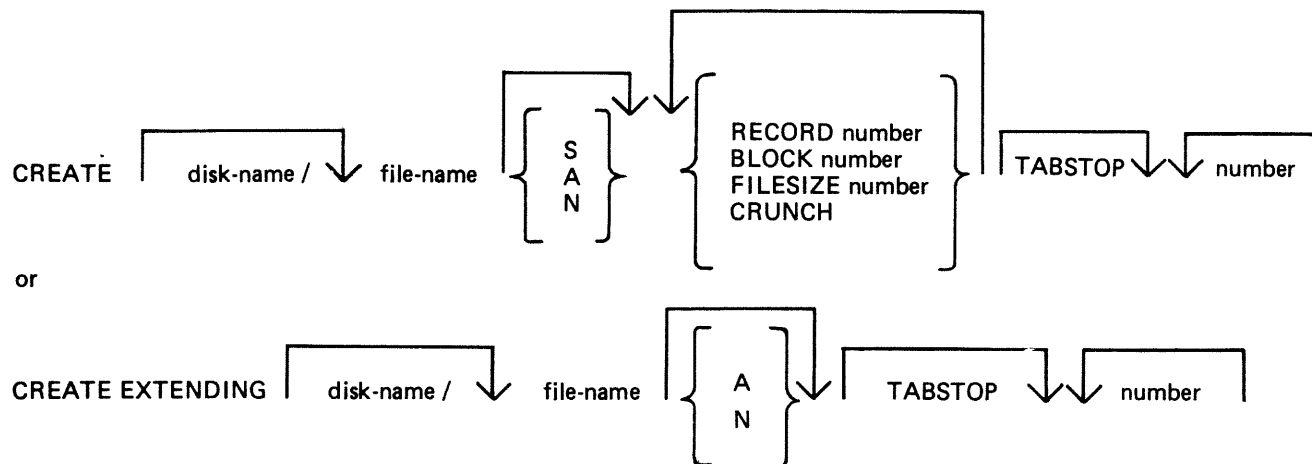
Message	Possible Causes	Suggested Action
CP : NUMBER TOO LARGE	At least one number is out of range.	Re-input.
CP : MISSING OPERAND	Two consecutive operators (+, etc) have been entered with no number between.	Re-input.

Message	(continued) Possible Causes	Suggested Action
CP : HEX. NO. WITH MISSING "@"	Illegal symbol; or missing @ symbol at front or back of a hex value.	Re-input.
CP : INVALID OPERATOR	Illegal symbol; or an operator (+, etc) has been omitted.	Re-input.
CP : OVERFLOW	An intermediate value in the computation is greater than the maximum value.	Re-input.
CP : DIVISION BY ZERO	Self-explanatory.	Re-input.
CP : ILLEGAL NUMBER	Self-explanatory.	Re-input.
CP : NUMBER TOO LARGE	Self-explanatory.	Re-input.
CP : MISSING NUMBER	Self-explanatory.	Re-input.
CP : ILLEGAL HEX NUMBER	Self-explanatory.	Re-input.

CREATE (Create Disk File)

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

Format diagram:



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	51 61 111 221 331 441
221	
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.

If the FILESIZE is specified and records are entered beyond the given filesize, then the error message is displayed after (filesize + 2) records have been entered. The last two records will not be written, due to the blocking of the output file by CREATE. For example, if the request

```
CREATE ICFILE FILESIZE 15
```

is followed by more than 15 records entered, then the message 'OUTPUT FILE TOO SMALL' will be given after the seventeenth record is entered; and the utility will go to EOJ without writing records 16 and 17 to the output file.

Output Messages

Message	Possible Causes	Suggested Action
FILETYPE IS NOT SOURCE OR DATA	A file other than a "source" file or a "data" file was specified.	Check for correct file; check input and re-enter if necessary.
ILLEGAL PARAMETER LIST - BAD ATTRIBUTE	Specified record and block sizes are incompatible or blocking factor >255.	Check input and re-enter.
ILLEGAL PARAMETER LIST - TABS ERROR	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.
ODD NUMBER OF HEXADEXIMAL CHARACTERS INPUT	Warning message that an odd number of hexadecimal characters was input. When input mode is "hexadecimal", utility expects even number 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 re-enter record modification.
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.

Message	(continued) Possible Causes	Suggested Action
OUTPUT FILE TOO SMALL	Attempt was made to add records to specified file beyond its maximum filesize. 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.
PERMANENT ERROR ON OUTPUT FILE	Utility is not able to write any more records to the new disk file because of errors on the disk. File is closed with lock.	List the file to check which records have been entered; then COPY to a different file and continue using CREATE EXTENDING.
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.
END OF OUTPUT FILE REACHED	The last logical record of the output file has been stored (applies to the utility "UPDATE").	None.

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

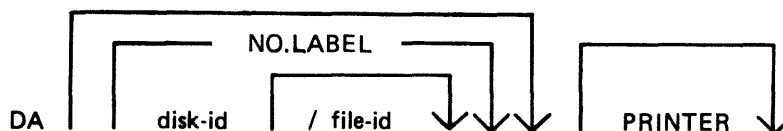
DA (Disk Analysis)

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 diagram:



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 via the keyboard a number of commands. 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.

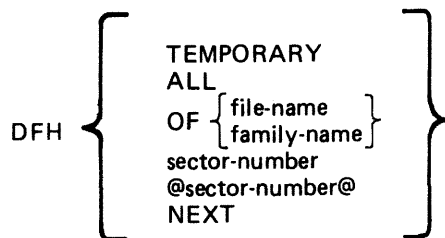
LABEL

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:

Format diagram:



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.

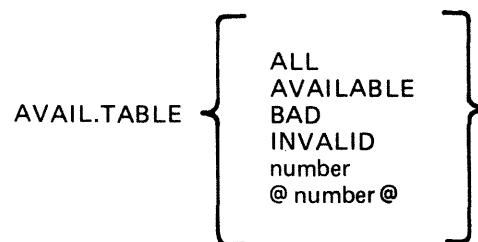
The “sector-number” option displays any sector in disk file header format, where the number is a decimal sector address. If preceded by an @ symbol, the sector-number is in hexadecimal. This feature can be used after displaying other parts of the directory, which include sector addresses for disk file 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:

Format diagram:



The “ALL” option displays the entire available table.

The “AVAILABLE” option displays entries for available area only.

The “BAD” option displays entries for bad sectors only.

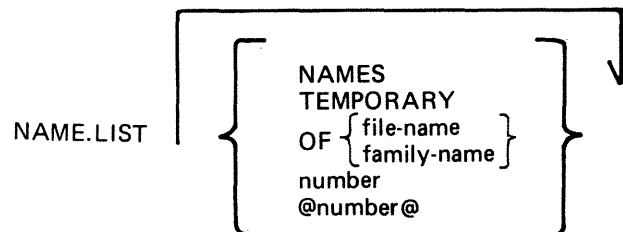
The “INVALID” option displays all entries in the available table which are invalid, in that the “length” entry does not equal the difference between “start address” and “end address”.

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:

Format diagram:



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 header of the specified file, or group of files : all headers will be checked and duplicates will be displayed if found.

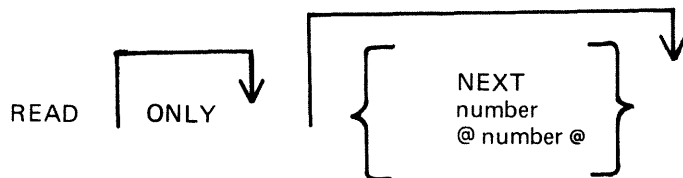
The “sector-number” option displays any sector in name-list format, where the number is a decimal sector address. If preceded by an @ symbol, the sector-number is in hexadecimal.

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:

Format diagram:



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.

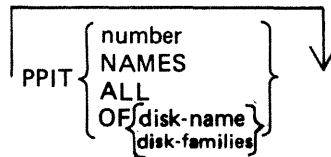
```
KFPB OF {file-name }
        {family-name }
```

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:

Format diagram:



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 or a group of disks.

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.

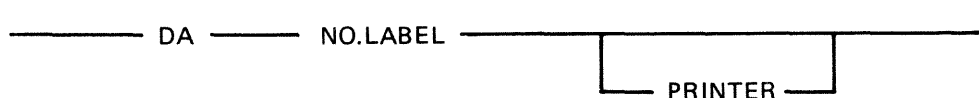
Analysis of Non-CMS Standard Disks

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.

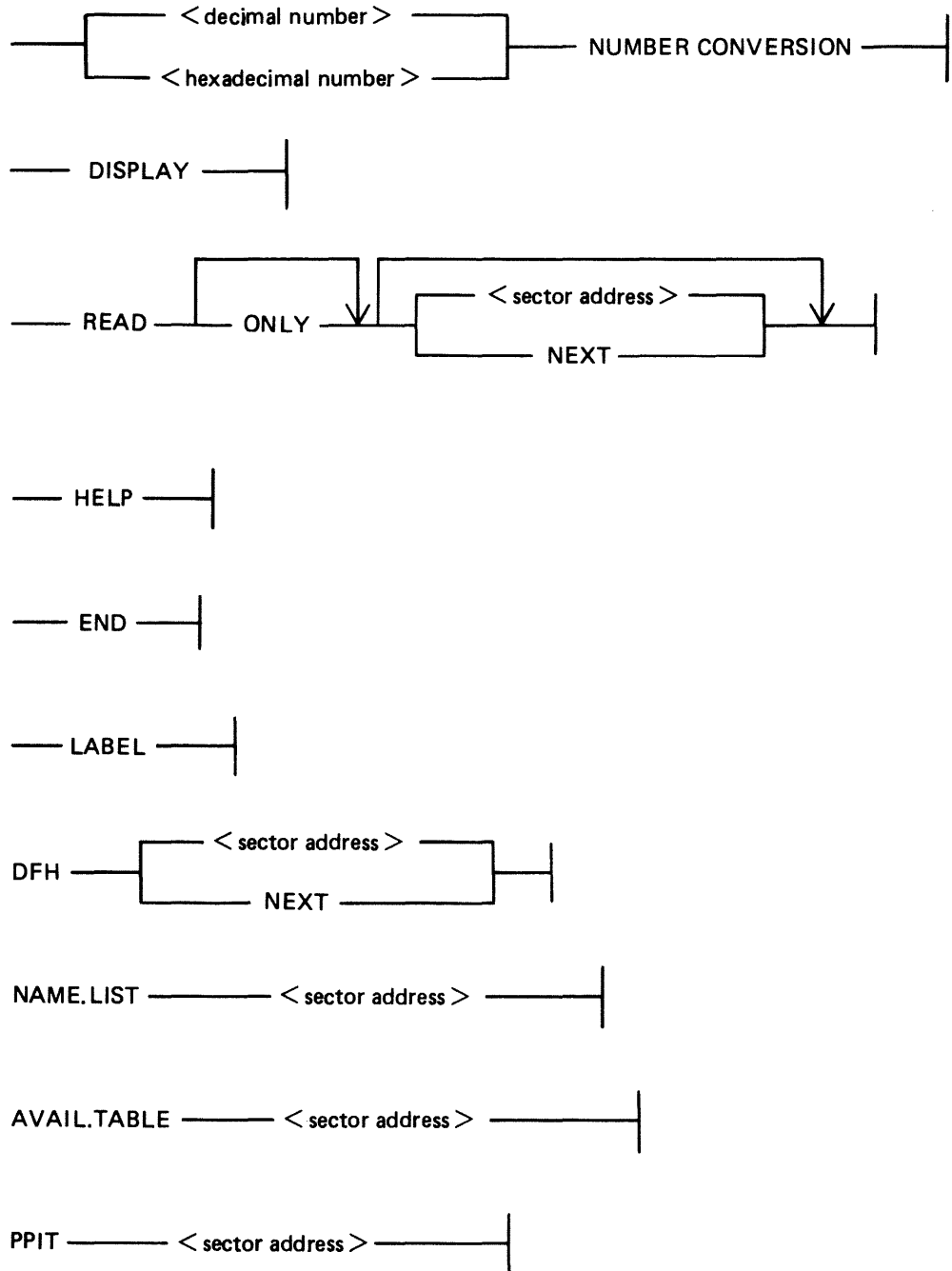
The initiating message for analyzing non CMS standard disks is:

Railroad diagram:



Command options available when analyzing a non CMS standard disk:

Railroad diagram:



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:

2015228

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:

Railroad diagram:

DISK — < disk-name > —

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
AVAILABLE	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.
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 DCL OF disk-name	An error was encountered while reading sector zero of the specified disk.	None.
PARAMETER REQUIRED	The option which has been specified is incomplete.	Re-input the complete option.

Message	(continued) Possible Causes	Suggested Action
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.
ERROR ON READ ADDRESS:--	The disk parity error has been encountered while reading a record in disk 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.
MTR MAY BE RUN ON THIS DISK	The disk has sectors reserved in the available table for MTR purposes.	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.
SYSTEM DISK LOCKED	The initiating message specified that the system disk, or a file residing on the the system disk, is to be analyzed, but the 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.

DSKUTL

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 not implemented on B 90 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.

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@

s = SDI and c = caelus in the above table

** = B900 ONLY

NOTE

On SDI type disks (BSMII and 211), an execution of the IN function is equivalent to the already implemented Reformat Function, RF, with the IGNORE option specified.

Bootstrap Table

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/ CAELUS	2A80-2ABF 2B00-2B3F 5540-557F	MTR/ CAELUS	2A80-2ABF 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
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) necessary to support a system without a console printer.

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 LIVM 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 and BSMII mini) 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 greater.

If the <COLDSTART> option has been specified in the initiating message of RF or IN, the following messages may be displayed. For messages produced by the BOOT option, see subsection DSKUTL BOOT.

Message	Possible Cause	Suggested Action
COLDSTART REQUEST IGNORED – VALID ONLY ON B 90 SYSTEM	On a B 900 system, a BDS bootstrap is selected for the disk.	None.
<file-name> DOES NOT CONTAIN COLDSTART BOOTSTRAP followed by DSKUTL <IN or RF> ABORTED	The bootstrap file is in CMS format but does not contain the COLDSTART bootstrap. The utility goes to end of job.	Make available CMSBOOT version 3.04 or greater.
Dxy IS NOT A MINI DISK – CANNOT BE USED FOR COLDSTART followed by DSKUTL <IN or RF> ABORTED	A disk or type other than BSMII or 1 MB mini has been AD'ed to the utility. The utility goes to end of job.	Use mini disk with this option.

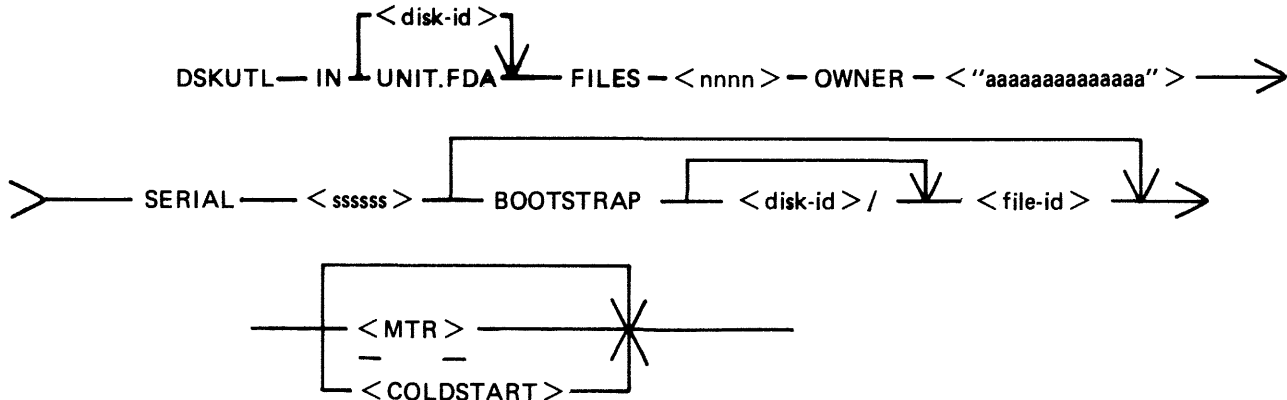
2015228

Format 1: IN

To initialize a disk using DSKUTL the following steps are required.

The following format provides the initializing (IN) function:

Railroad diagram:



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 "aaaaaaaaaaaa"	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 (").

	(continued)	
Message	Possible Cause	Suggested Action
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.	

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:

READY AND RESERVE REQUIRED DRIVE FOR DISK INITIALIZE

followed by the MCP prompt:

<mix number>/DSKUTL <28> WAITING INITIALIZE DK
DEVICE ASSIGNMENT REQUIRED

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 may be necessary to ready (RY) the drive after it has been reserved.

If Test Patterns have been specified and the disk AD'ed is either a BSMII or 211, the utility displays the following message and continues executing:

TEST.PATTERN<S> NOT REQUIRED FOR (BSMII OR 211) DISK

If the disk label is already in CMS format, the utility displays the following prompt:

Dxy IS CMS DISK <disk-name> – ENTER OK TO CONFIRM <IN> REQUIRED

followed by the MCP prompt:

<mix number>/DSKUTL ACCEPT

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 "DSKUTL <IN or RF> CONTINUING – READ ERROR IGNORED".

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 PHYSICAL UNIT <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:

```
DSKUTL <IN> - <bootstrap used> BOOTSTRAP VERSION <version number>
              USED ON Dxy DISK <disk-name>
              <sectors> SECTORS ARE UNAVAILABLE
              ON Dxy DISK <disk-name>
```

where - <bootstrap used> = CMS/WARMSTART if the CMS bootstrap was used,
or
B 900 SYSTEM if the BDS 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.

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.

DSKUTL IN Error Messages

The following messages may be displayed by the IN function of this utility before the utility terminates with the following message:

```
DSKUTL <IN> ABORTED
```

2015228

Error Message	Description
PARAMETER SPECIFICATION FOR <IN> INCOMPLETE	At least one of the messages which must be specified for the initialize function is missing – check initiating message. The utility goes to End of Job.
NO PPIT ON SYSTEM	UNIT.FDA has been specified and the system running is not a Fixed Disk Assemblage.
BOOTSTRAP FILE <file-name> NOT CMS STANDARD	The checkstrings and/or attributes of the specified or default bootstrap file are incorrect. Copy the bootstrap file from a good backup copy.
READ ERROR ON BOOTSTRAP FILE <file-name>	A read error has been encountered on the file containing the bootstrap code while verifying check strings or copying to track 0 of the disk. Copy the bootstrap file from a good backup copy.
NO UNIT SLOTS AVAILABLE IN SYSTEM PPIT	UNIT.FDA was specified. The utility has found all physical units listed in the System PPIT of the Fixed Disk System to be on line and there are no available physical unit slots in the System PPIT.
READ ERROR ON PPIT OF <disk-id>	A read error has been encountered on the System PPIT while scanning for physical unit disk-name.
OPEN DISK UNLABELLED NOT IMPLEMENTED	The utility has been unable to invoke an unlabelled disk open as the facility is not supported by the MCP.
CAPACITY OF D _{xy} DISK NOT RECOGNIZED – <no. sectors> SECTORS.	The disk size reported by the MCP <no. sectors> does not match that of any disk type supported by this utility.
D _{xy} IS NOT A FIXED DISK	UNIT.FDA was specified and a disk drive other than a 201-I or a 211 (DF _y) has been AD'ed to the utility.
DSKUTL <IN> OF D _{xy} NOT CONFIRMED	The initialize option has not been confirmed.
INIT.TRACK COMMUNICATE NOT SUPPORTED	A Caelus type disk has been AD'ed to the utility and the communicate which is required for initialization is not supported by the MCP.
INITIALIZATION ERROR ON D _{xy} AT SECTOR z	An error condition has been returned by the INIT.TRACK communicate while initializing the sector z of a Caelus type disk.
TRACK 0 OF D _{xy} BAD	The utility is unable to read or write to one or more of sectors 0-31. Replace the disk.
BAD MTR SECTOR z ON D _{xy}	The utility is unable to read or write to an MTR sector (see IN Bootstrap table). Replace the disk.

(continued)

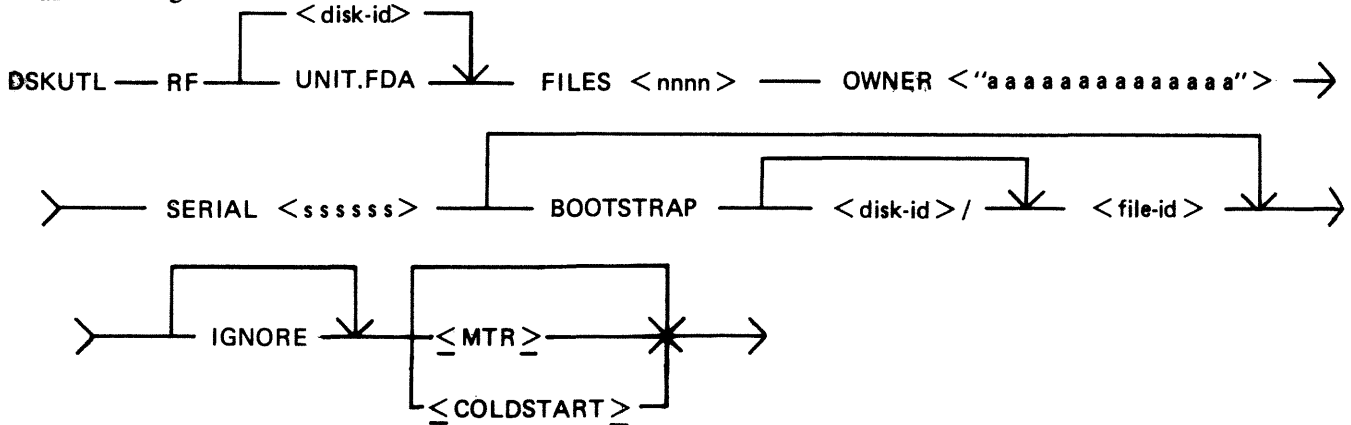
Error Message	Description
CANNOT ACCOMMODATE DIRECTORY FOR Dxy	The utility cannot find an available area large enough to take the new disk directory. Reduce the number of files allowed on the disk or replace the disk.
AVAILABLE TABLE TOO SMALL FOR Dxy	The new Available Table is too small to accommodate all the bad sectors found by the utility and/or reserved MTR sectors. Replace disk.
WRITE ERROR ENCOUNTERED ON Dxy AT SECTOR z	The utility has encountered a write error while writing test patterns, the directory, the Bootstrap code to the disk, or MTR patterns to a BSMDII disk. The utility continues if the error was encountered while writing test patterns or the directory and the sector is denoted as bad in the available table. Otherwise the utility goes to End Of Job.
MORE THAN x BAD SECTORS FOUND ON Dxy	The utility has encountered more than 31 sectors with read or write errors while trying to write the disk directories of a 1 MB mini disk or more than 49 errors on any other type disk. Replace the disk.

Format 2: RF

NOTE

The diamond braces must be included in the initiating message for MTR tracks to be reserved. Similarly, the diamond braces must be included around COLDSTART.

Railroad diagram:



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	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 "aaaaaaaaaaaa"	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.
<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.

Operation

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 the bootstrap file is copied to the disk.

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 "READY AND RESERVE REQUIRED DRIVE FOR DISK REFORMAT"
"<mix number>/DSKUTL <14> WAITING UNLAB REFORMAT DK DEVICE RE-
QUIRED"

Enter "RD DMA"

System displays "DMA RESERVED 0 FILES OPEN"

Enter "AD <mix number> DMA"

System displays "DMA IS CMS DISK TESTA - CONFIRM <RF> REQUIRED"

Enter "AX <mix-number> OK"

System displays:

"CMS BOOTSTRAP VERSION XX.XX.XX USED ON DMA DISK DISKA"

"0 SECTORS ARE UNAVAILABLE ON DMA DISK DISKA"

"DSKUTL <RF> OF DMB DISK <16672> AS DISK DISKA ON B90 COMPLETED"

Enter "RY DMA"

2015228

The reformatted disk can now be accessed as normal.

DSKUTL RF Output Messages

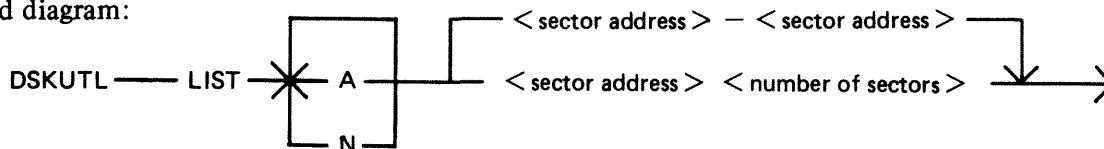
Message	Possible Causes	Suggested Action
PARAMETER SPECIFICATION FOR <RF> INCOMPLETE	At least one of the parameters required in the initiating message for the reformat option is missing.	Check initiating message and re-input.
NO PPIT ON SYSTEM	UNIT.FDA was specified in the initiating message and the system running is not a fixed disk assemblage.	Check system configuration and initiating message.
BOOTSTRAP FILE <file-name> NOT CMS STANDARD	The checkstring of the specified or default BOOTSTRAP file is incorrect.	Check specified BOOTSTRAP file for integrity. Copy a new version if necessary.
READ ERROR ON BOOTSTRAP FILE <file-name>	A read error has been encountered on the specified or default BOOTSTRAP file.	Copy a new BOOTSTRAP file to the system.
NO UNIT SLOTS AVAILABLE IN SYSTEM PPIT	UNIT.FDA was specified in the initiating message. The utility has found all physical units listed in the system PPIT to be on-line, and there are no available physical unit slots in the system PPIT.	Check initiating message and system configuration.
READ ERROR ON PPIT OF <disk-name>	A read error has been encountered on the system PPIT while scanning for physical unit disk-name.	Re-initialize disk.
READY AND RESERVE REQUIRED DRIVE FOR DISK REFORMAT	This message will be displayed by the utility before the MCP prompts the operator to AD the required disk drive.	Ensure disk being reformatted is reserved (RD).
OPEN DISK UNLABELLED NOT SUPPORTED	The utility is not able to open unlabelled the disk as the level of MCP in use does not support this feature.	Check version number of MCP.
CAPACITY OF Dxy NOT RECOGNIZED – <number of sectors> SECTORS	The disk-size <number of sectors> returned by the MCP does not match any of the disk types supported by the utility.	Check the system configuration.
Dxy IS NOT A FIXED DISK	The disk assigned for RF UNIT.FDA is not a 2011 or 211 fixed disk.	Check device assigned.
CANNOT READ SECTOR 0 OF Dxy	The utility has opened Dxy successfully but is unable to	If the disk is a Caelus disk, ensure that it has been

Message	(continued) Possible Causes successfully read sector 0.	Suggested Action previously initialized. If this is the case, the disk will require re-initializing using SAU IN. For SDI disks, use the IGNORE option.
DISK LABEL OF Dxy NOT IN CMS FORMAT	IGNORE was not specified in the initiating message and the disk being reformatted does not contain a CMS format label in sector 0.	Re-enter initiating message using the IGNORE option.
Dxy IS CMS DISK <disk-id> – CONFIRM RF REQUIRED	The disk label is already in CMS format.	Enter AX <mix-number> OK to continue.
DSKUTL <RF> OF Dxy NOT CONFIRMED	The utility has gone to EOJ without completing the reformat function. OK not entered as above.	Re-run the utility.
TRACK 0 OF Dxy BAD	The utility is unable to read from or write to one or more sectors of track 0.	Check integrity of media and drive. Replace media if necessary.
BAD MTR SECTOR <z> ON Dxy	The utility is unable to read from or write to one or more of the disk MTR sectors.	Check integrity of media and drive. Replace media if necessary.
DFy WILL BE REFORMATTED AS PHYSICAL UNIT <disk-id>	UNIT.FDA was specified in the initiating message and this disk name will be assigned to the fixed disk being reformatted.	Information only.
READ ERROR ON OLD DIRECTORY AT SECTOR <z> OF Dxy	The utility has encountered a read error while reading the directory of the disk being reformatted.	Re-enter the initiating message using the IGNORE option. WARNING – This should be used with caution for Caelus disks.
OLD DIRECTORY OF Dxy IS CORRUPT	The utility has encountered an invalid entry while scanning the available table for bad sector entries.	Re-enter the initiating message using the IGNORE option. WARNING – This option must be used with caution for Caelus disks.
TOO MANY BAD ENTRIES IN THE OLD AVAILABLE TABLE OF Dxy	More than 90 bad entries have been found in the available table of the disk being reformatted.	Use IGNORE option. If the disk is Caelus disk, it must be initialized with SAU.
CANNOT ACCOMMODATE DIRECTORY FOR Dxy	The utility cannot find an available area large enough to accommodate the directory as specified.	Check and re-enter initiating message with smaller number of files to reduce size of new directory. Use IGNORE if SDI disk being used.

Message	(continued) Possible Causes	Suggested Action
AVAILABLE TABLE TOO SMALL FOR Dxy	The new available table is not large enough to accommodate all required entries. Bad entries maintained from old available table, MTR sectors, and any bad sectors found during RF will not fit in area.	Re-enter initiating message with a larger number of files to increase size of new available table.
WRITE ERROR ENCOUNTERED ON Dxy AT SECTOR <z>	The utility has encountered a write error while writing Bootstrap records MTR patterns Available table Directory. If the error is encountered while writing the directory, DSKUTL will mark these as bad in the available table and continues execution. For all other write errors it will abort.	Check integrity of disk drive and media. Replace media if necessary.
MORE THAN <z> BAD SECTORS FOUND ON DISK Dxy	More than 31 read/write errors encountered during reformatting of a 1MB disk, or more than 49 errors for all other disk types.	Check integrity of disk drive and media. Replace media if necessary.
DSKUTL <RF> ABORTED	Utility has terminated prematurely.	See message displayed immediately prior to abort message for reason.
<file-name> BOOTSTRAP VERSION <version-num> USED ON Dxy DISK <disk-id>	Confirmation of BOOTSTRAP file used during reformat.	Information only.
<z> SECTORS ARE UNAVAILABLE ON Dxy DISK <disk-id>	Confirmation of number of bad sector entries placed in the available table of the reformatted disk excluding MTR sectors.	Information only.
PLEASE RUN CHECK.DISK ON Dxy DISK <disk-id>	For 1MB disk only.	It is recommended that CHECK.DISK is run on all reformatted 1MB disks to ensure that surface integrity checks are performed.
DSKUTL <RF> OF Dxy DISK <disk-size> AS <disk-id> ON <system-name> COMPLETED	Normal termination on successful EOJ.	Information only. Reformatted disk must be RY'ed before it can be accessed normally.

Format 3: LIST

Railroad diagram:



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:

"PRESENT DISK FOR LISTING NOW"

"<mix-number>/DSKUTL <14> WAITING UNLAB LIST DK DEVICE REQUIRED"

Enter "RD DMA"

The system displays "DMA RESERVED 0 FILES OPEN"
 Enter "AD <mix-number> DMA"

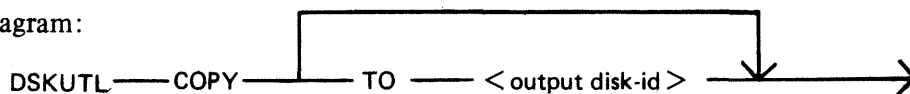
The utility now lists the specified sector numbers and goes to EOJ.

DSKUTL LIST Output Messages

Message	Possible Causes	Suggested Action
PRESENT DISK FOR LISTING PLEASE	The utility displays this message prior to the MCP. Prompts the user to RD and AD the required disk.	RD and RY the required disk.
OPEN DISK UNLABELLED NOT SUPPORTED	The utility is unable to open unlabelled the disk as the MCP being used does not support this feature.	Check version number of MCP in use.
ILLEGAL SECTOR ADDRESS <sector-address> FOR Dxy	The sector addresses specified in the initiating message are outside the physical capacity of the disk.	Check initiating message and re-enter.
Dxy EXHAUSTED DURING SECTOR RANGE <sector-address - sector-address>	The last physical sector address of the disk in drive Dxy has been read. The utility will continue with the next specified sector range.	Check initiating message.
READ ERROR ENCOUNTERED ON Dxy AT SECTOR <sector-address>	A read error was encountered at the specified address. The utility will continue.	Information only. Check KA listing of the disk for listed BAD sectors.
DSKUTL LIST ABORTED	The utility has prematurely terminated.	See message displayed immediately prior to abort message for reason.

Format 4: COPY

Railroad diagram:



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:
 "PRESENT INPUT DISK FIRST PLEASE"

The operator must now reserve and assign the disk drive required for input. The utility displays the message:

“PRESENT OUTPUT DISK NOW PLEASE”

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 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.

Dxy DEVICE TYPE NOT SUPPORTED
DSKUTL <COPY> ABORTED

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 BAD SECTORS FOUND ON DMx
DSKUTL <COPY> ABORTED

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 and incorrupt information.

Before the utility copies the label of the input disk (this is the last sector to be copied), the utility deals with each sector error encountered (the address of which was noted during the disk copy) in the following way:

Error Detected on INPUT Disk	Action by Utility on OUTPUT Disk
Sector 1-31	Write binary zeros to all sectors 1-31 and display the message: BOOTSTRAP CODE NOT COPIED TO OUTPUT DMx DISK <disk-id>
MTR Sector	Reset MTR Flag in Disk Label. MTR areas remain denoted as “BAD” in the Available Table. Message displayed: CANNOT RUN MTR ROUTINES ON OUTPUT DMx DISK <disk-id>
Available Table	Write new sector with 29 unused entries and one sterile entry. Available areas and/or the ghost entry may be lost. *

(continued)
Action by Utility on OUTPUT Disk

**Error Detected on
INPUT Disk**

Name List	Build and write new name list sector with file-ids traced from associated DFH sectors. Remember that a file entry in the Name List is in the same ordinal position in the File Header list.
1 st File Header "SYSTEMEM"	Build and write new DFH for SYSTEMEM.
File Header	Write new DFH with @80@ (available) in the file-id field, trace corresponding Name List entry and if used, display the message*: <file-name> NOT COPIED TO OUTPUT DMy DISK <disk-id>
File Area	Alter the file-id fields in both the DFH and Name List entries of the relevant file to @81@ (temporary); set integrity flag in disk label to @FF@ to invoke MCP AVR disk clean routine when the disk is readied; this returns any disk areas allocated to this file to the available table and alters the Name List entry to unused. Message displayed : <file-name> NOT COPIED TO OUTPUT DMx DISK <disk-id>

* Note

The message "AREAS POSSIBLY MISSING FROM Dxy DISK <disk-id>" is also displayed when all Sector Addresses in Error have been scanned.

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"

The system displays "PRESENT DISK FOR INPUT FIRST PLEASE"

"<mix-number>/DSKUTL <14> WAITING UNLAB COPY DK DEVICE REQUIRED"

Enter "RD DMA"

The system displays "DMA RESERVED 0 FILES OPEN"

enter "AD <mix-number> DMA"

The system displays "PRESENT DISK FOR OUTPUT NOW PLEASE"

"<mix-number>/DSKUTL <14> WAITING UNLAB COPY DK DEVICE REQUIRED"

Enter "RD DMB"

The system displays "DMB RESERVED 0 FILES OPEN"

Enter "AD <mix-number> DMB"

The utility now copies the disk and goes to EOJ.

DSKUTL COPY Output Messages

Message	Possible Causes	Suggested Action
PRESENT INPUT DISK FIRST PLEASE	The utility will display this message just prior to the MCP prompting the operator to RD and AD the input disk.	RD and AD the disk required for input.

Message	(continued) Possible Causes	Suggested Action
PRESENT OUTPUT DISK NOW PLEASE	The utility will display this message just prior to the MCP prompting the operator to RD and AD the output disk.	RD and AD the disk required for output.
OPEN DISK UNLABELLED NOT SUPPORTED	The utility is unable to open the disk as unlabelled, since the level of MCP in use does not support this feature.	Check the version number of the MCP.
Dxy AND Dxz HAVE DIFFERENT CAPACITY	The two specified disks are incompatible for this function.	Check correct devices have been AD'ed. Only like disk types can be COPIED.
TRACK 0 OF Dxy BAD	The utility is unable to read from or write to one or more sectors of track 0.	Check integrity of media and drive. Replace media if necessary.
Dxy DEVICE TYPE NOT SUPPORTED	An attempt has been made to copy a disk which is not an SDI type disk.	Check AD command for correct device type and re-execute utility.
DSKUTL <COPY> ABORTED	The utility has terminated prematurely.	See message displayed immediately prior to abort message for reason.
SPECIFIED DISK-ID <output-diskname> FOR COPY IGNORED>. INPUT Dfy <disk-name> HAS A PPIT	The output disk will be assigned the same disk-name as the input disk. The utility will continue.	Information only.
SPECIFIED DISK-ID <output disk-name> FOR COPY IGNORED. Dxy DISK NOT IN CMS FORMAT	The input disk is not in CMS format. The specified output disk-name is ignored but the utility will continue.	Information only.
READ ERROR ENCOUNTERED ON Dxy AT SECTOR <z>	A read error has been encountered on the input disk. The utility will continue.	See recommendations for read/write errors below.
WRITE ERROR ENCOUNTERED ON Dxy AT SECTOR <z>	A write error has been encountered on the output disk. The utility will terminate.	See recommendations for read/write errors below.
DSKUTL COPY FROM Dxy TO Dxz COMPLETED	Successful terminating message when the input disk is not in CMS format.	PO disk and remove from drive.
DSKUTL COPY FROM Dxy DISK <disk-name> TO Dxz DISK <disk-name> COMPLETED	Successful terminating message when the input disk is in CMS format.	RY disk to enable normal access.

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.

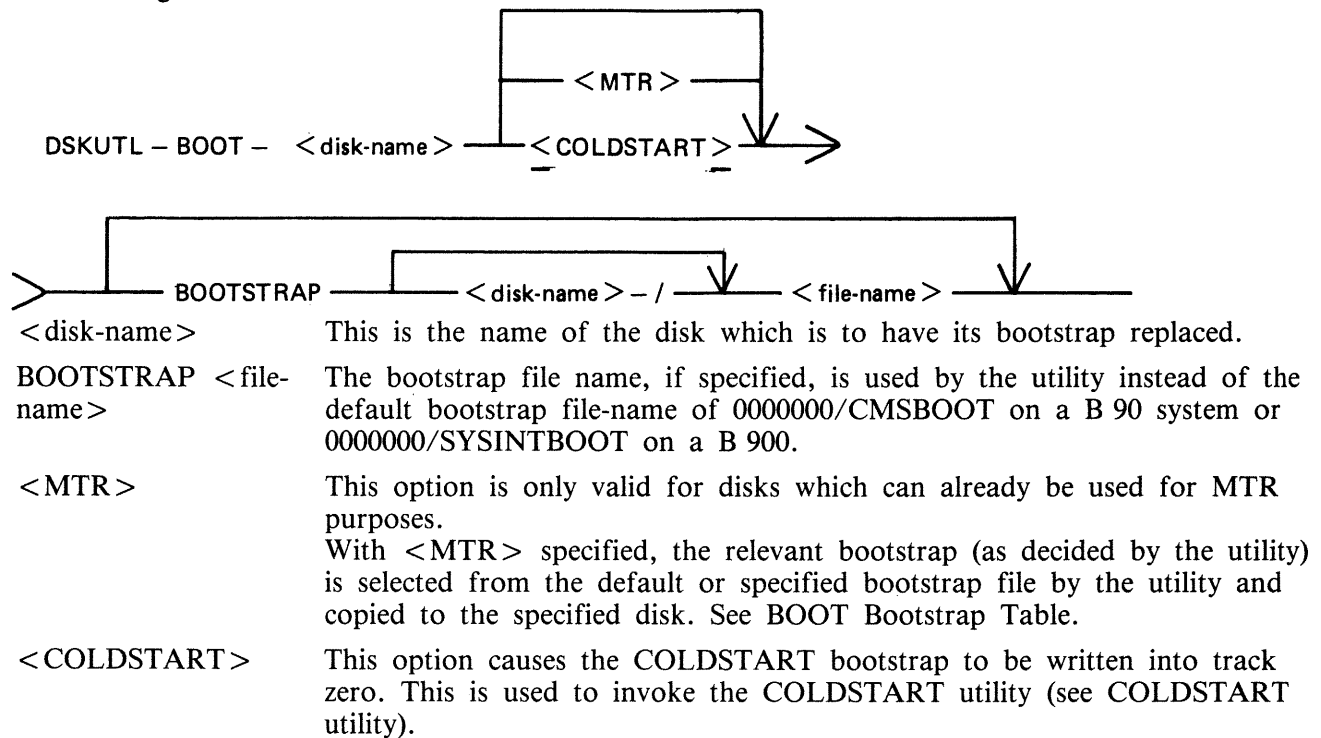
If the read/write error occurs in the disk directory area, the disk should not be used without re-initialization (or re-formatting if it is an SDI disk).

Format 5: BOOT

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:

Railroad diagram:



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:

```
DSKUTL <BOOT> - <bootstrap used>
      BOOTSTRAP VERSION <version number>
      COPIED TO DISK <disk-name>
```

where - <bootstrap used> = CMS/WARMSTART if the CMS bootstrap was used,
or
B900SYSTEM if the BDS 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 is replaced by either CMS or BDS bootstrap, it is impossible to replace this bootstrap with an MTR bootstrap. The disk must be reformatted or re-initialized specifically for MTR purposes. Any areas reserved for MTR purposes remain unavailable.

Table 4-4A. 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 disks, reserved disk areas are checked for readability before the bootstrap is copied.

Table 4-4B. DSKUTL BOOT Bootstrap Table If COLDSTART Specified

Disk Type	Bootstrap Selected	
	B 90	B 900
	Default bootstrap file is 0000000/CMSBOOT	Default bootstrap file is 0000000/SYSINTBOOT
BSMII Mini and 1MB Mini	COLDSTART	Not Supported

DSKUTL BOOT Error Messages

If an error is encountered by the utility, one of the following error messages is displayed before the utility terminates, together with the message :

BOOTSTRAP REPLACEMENT ABORTED

Error Message	Description
FUNCTION VALID ONLY ON B90 AND B900 SYSTEMS	The BOOT utility only functions on B 90 and B 900 systems.
COLDSTART OPTION IS VALID ONLY ON B90 SYSTEM	The <COLDSTART> option can only be used on B 90 systems.
<file-name> DOES NOT CONTAIN COLDSTART BOOTSTRAP	The <COLDSTART> option has been selected and the Bootstrap file with <file-name> does not contain the code for the COLDSTART bootstrap.
BOOTSTRAP FILE <file-name> IS NOT CMS STANDARD	The checkstrings and/or attributes of the specified or default B 90 or B 900 Bootstrap file are incorrect. Check the Bootstrap file used.
<disk-name> IS NOT A MINI DISK – CANNOT BE USED FOR COLDSTART	On a B 90 system, the <COLDSTART> option has been specified but the AD'ed disk was not a 1 MB mini disk or BSMII disk. See COLDSTART utility.
<disk-name> IS NOT AVAILABLE	The specified disk is not recognized as being on the system.
<disk-name> IS LOCKED	The requested disk has been locked and is being used by another utility.
<disk-name> IS A PSEUDO DISK	The specified disk is not a physical disk.
DISK <disk-name> WAS NOT INITIALIZED ON B90/B900 SYSTEM	<disk name> was not initialized or reformatted on the same type of system as the utility is running on.
DISK <disk-name> CANNOT BE USED FOR MTR PURPOSES	The option <MTR> has been specified in the initiating message and <disk-name> has not been initialized or reformatted as an MTR disk.
TRACK 0 OF <disk-name> IS BAD	A write error has been encountered in track zero while copying the bootstrap code. This disk is unable to be used.
READ ERROR ON BOOTSTRAP <file-name>	A read error has been encountered while reading the bootstrap code. <file-name> should be replaced with a good backup copy.

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 DISK NAME <UP TO 7 LEGAL CHARACTERS> OR 'UNIT.FDA' IF DISK FOR FIXED ASSEMBLAGE	The user is required to AX the disk name. If the system is running on a Fixed Disk Assemblage, the additional prompt to input UNIT.FDA is displayed.
ENTER NO. OF FILES <NUMBER LESS THAN 2805>	The user is requested to AX the number of files to reside on the disk to less than 2805.
ENTER SERIAL NO. <6 DIGITS>	The user is requested to AX a six digit serial number.
ENTER OWNERS NAME <UP TO 14 CHARACTERS>	The user is requested to AX up to 14 alphanumeric characters.
IS THIS DISK FOR MTR USE? -	The user is requested to AX "Y" if the disk is to be used as an MTR disk. Otherwise AX "N".
IS THIS DISK FOR COLDSTART USE? - <Y OR N>	This prompt is only displayed if "N" was AX'ed to the previous prompt. The user is required to AX "Y" if the disk is to be used as a coldstart disk. Otherwise AX "N". Note, a coldstart disk can only be a BSMII or 1 MB mini disk and the system the utility is running on must be a B 90.
ENTER REQUIRED BOOTSTRAP FILE- NAME OR NULL IF <disk-name>/ <bootstrap-name> IS TO BE USED	The user is requested to AX the file name of the file containing the required Bootstrap code. If the default Bootstrap file name displayed in the prompt, (for example, CMSBOOT on a B 90 or SYSINITBOOT on a B 900) is the required Bootstrap file, only a space " " need be: AX'ed to the utility.
DSKUTL <IN> - ENTER NO. OF TEST PATTERNS TO BE SPECIFIED <NUMBER 0-4>	This prompt is only displayed when the IN function has been requested. The user is required to AX the number of test patterns to be written to a Caelus type disk (1 MB, cartridge), 0-4. See Test Patterns for Caelus Type Disks. For disks other than Caelus type disks 0 should be AX'ed.
ENTER nth TEST PATTERN <20 CHARACTER MAX. DE-LIMIT WITH @ IF HEX, " IF ASCII>	This prompt is only displayed when the number of test patterns specified is greater than 0; for the IN function prompt ENTER NUMBER OF TEST PATTERNS The user is required to AX test pattern n. See Test Patterns for Caelus type Disks. This prompt is issued n times, n being the number of patterns requested.

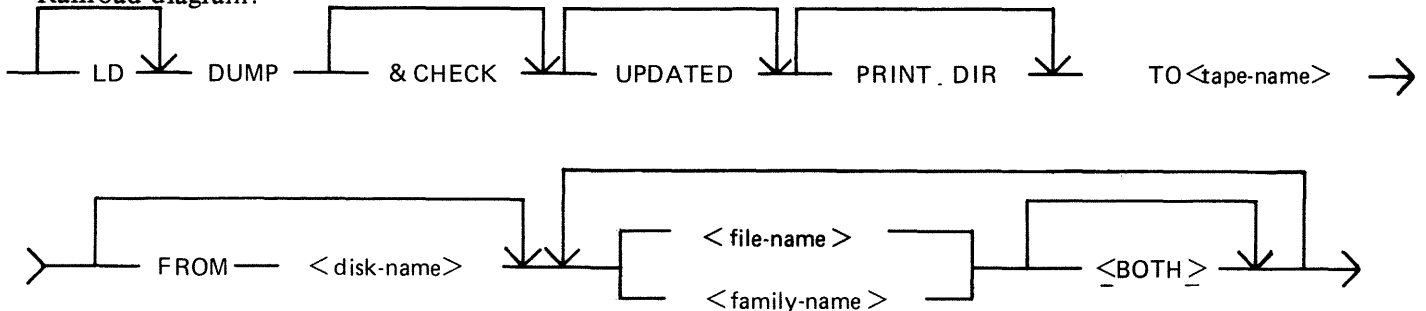
Prompt	Description
IS THE DIRECTORY ON THIS DISK TO BE IGNORED? - <Y OR N>	This prompt is only issued if the RF function has been requested. The user is required to AX "Y" if the IGNORE option is to be used. See RF IGNORE option. Otherwise "N" should be AX'ed.
** INVALID SELECTION - RETRY **	A response other than "Y" or "N" has been AX'ed to a prompt. The prompt is reissued.

DUMP (Dump Files from Disk to a Library Tape)

(LD – TAPE LIBRARY UTILITY)

This function is part of the utility LD. DUMP 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 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 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 ARCFIL1 and ARCFIL2 from the disk named ARCDISK to the tape named ARCTAPE, the following initiating message is used:

```
LD DUMP & CHECK TO ARCTAPE FROM ARCDISK ARCFIL1,  
ARCFIL2
```

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

Message	Description
NO FILE <file-name> ON DISK <tape-name> FOR DUMP	This file is not present on the disk. The utility continues to look for other file(s) to dump, if specified.
NO FILES IN THE FAMILY <family-name> ON DISK <disk-name> FOR DUMP	No files have been found by DUMP in the family <family-name>. The utility continues to look for other files to dump, if specified.
NO FILES TO DUMP	The utility has found that none of the files specified will be dumped. The utility goes to End of Job. Check that the disk has the required file(s).
<file-name> DUMP DISCREPANCY	The end of file has been reached sooner than expected, and implies that there is erroneous

(continued)

Message	Description
<file name> DUMPED	information in the Disk File Header for that file. The utility continues trying to dump the rest of the file(s) specified. This file has been dumped onto the specified tape.
<file-name> NOT FOUND	The file containing the <parameter-list> (star file) is not present. The utility goes to End of Job. Make the star file available or specify the parameters in the initiating message.
DISK <disk-name> NOT AVAILABLE or with a pseudo-disk arrangement PSEUDO-DISK 'pseudo-disk-id' ON DISK <disk-name> NOT AVAILABLE and if the pseudo-disk is unrestricted, then FILE LIST FOR <family-name> MAY BE INCOMPLETE or FILE LIST MAY BE INCOMPLETE	The specified disk is either not on the system or not ready. The utility goes to End of Job. Ensure that the required disk is on the system and ready before re-running the utility.
<file-name> NOT DUMPED - DATA FILE NOT ON LINE	The data file for the given key file is not present on the disk. The data file is not dumped. Note that the key file has been dumped. The utility continues. Retry the utility with a disk which contains both data and key files.
UTILITY LIMIT REACHED AT FILE <file-name>	The dumping is from an unrestricted pseudo-pack, the number of files to be dumped has reached the limit of 2804 files. The utility now stops collecting any more file names and commences the actual dumping.
INVALID CHARACTER IN IDENTIFIER <file-name> - WARNING	The utility has generated a file name which contains invalid characters. The utility continues.
<file-name> NOT DUMPED - <reason> where <reason> can be: IN OUTPUT USE or HAS BEEN REMOVED or HAS BEEN ALTERED or IS INCOMPLETE or REASON UNKNOWN or DISK WAS LOCKED	The requested file could not be dumped. The reason for this is explained in the message. If the reason is that the disk was LOCKED, that is, unavailable to LD, then the utility goes to End of Job as there are no further files on the tape.
In this case, the additional message below is displayed LOAD ABANDONED - ERROR IN DUMP	For any other reason, the utility continues trying to load any further requested files.
DUPLICATE - <file-name> ALREADY BEING DUMPED	More than one request has been made to dump this file in the initiating message. The utility continues. Note this is usually caused by specifying individual files and the family group containing the files.
COMMUNICATE FAILURE DUMP ABANDONED - TAPE SHOULD	LD is running on a B 90 with an MCP version equal to or greater than 3.04. The utility has

(continued)

Message	Description
BE PURGED	requested to select the speed at which it uses the tape drive. The MCP has reported an error to the utility. The utility terminates.
ZIP FAILURE WHEN ZIPPING CHECKADUMP FROM LD. ENTER "Y" TO RETRY ZIP, OR "N" TO TERMINATE.	" & CHECK " was specified in the initiating message, and LD has attempted to zip CHECKADUMP at the end of the dump operation. Check that CHECKADUMP is on the system disk; if it is not, copy it to the system disk and AX "Y" to LD to retry the zip of CHECKADUMP; otherwise, AX "N" to terminate the 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 contained 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 201I 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.

Format:

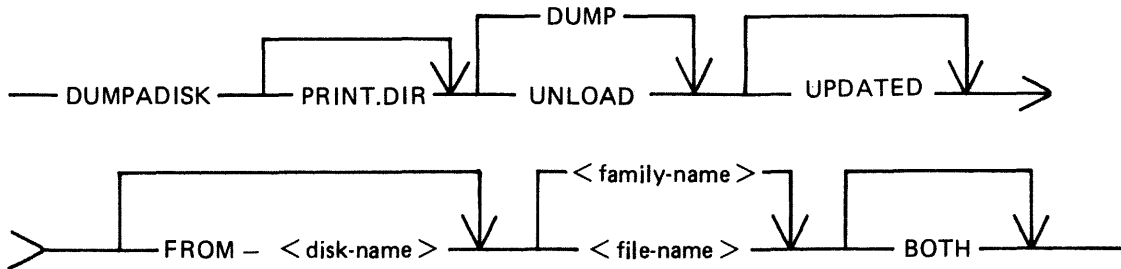
Format 1

DUMPADISK PRINT.DIR

This function lists the disk directory.

Format 2

Railroad diagram:



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) dumped.

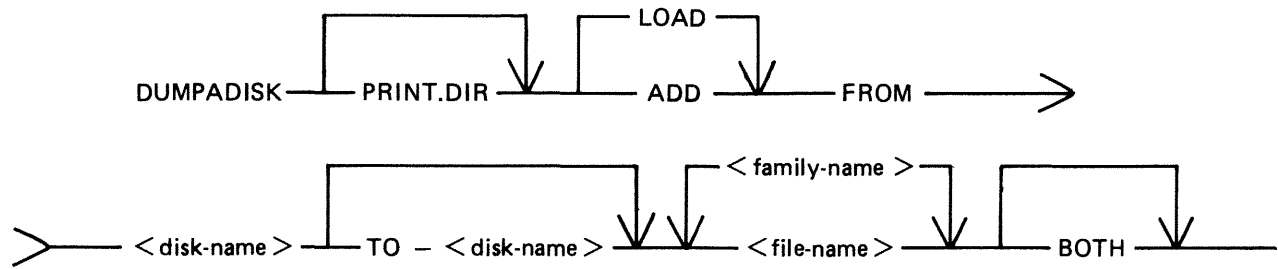
NOTE

See Archiving Using UPDATE Facility for a fuller explanation.

The use of DUMP option will copy file or files to backup medium only. Using UNLOAD option will copy file or files to backup medium and remove the original copy from disk.

Format 3

Railroad diagram:



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 by DUMPADISK previously 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.

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:

“ENTER A FOUR CHARACTER DUMP IDENTIFICATION PREFIX”

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:

“OVERFLOW DISK REQUIRED TO CONTINUE 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 then the dumping of the current file is terminated (see the error messages listed below). 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
<disk-id> IS NOT A RECOGNIZED BACK-UP DISK	The disk being accessed for a LOAD or ADD is not a disk produced by DUMPADISK.	Check that the correct disk was inserted in the drive.
<file-name> NOT LOADED - ALREADY ON DISK	The initiating message specified files to be ADDED which were already resident on the output disk.	RM existing files and re-run the ADD.
ALTHOUGH WITH DIFFERENT ATTRIBUTES	This message may follow and qualify the above message.	See above message.
<file-name> LOAD DISCREPANCY or <file-name> DUMP DISCREPANCY	DFH file-size and number of records DUMPed or LOADED are not equal.	Investigate discrepancy and re-execute the utility.
NO FILES TO DUMP	The utility is not able to find any of the files specified in the initiating message.	Check initiating message. If correct execute PD and check existence of files on the disk being backed-up.
FILE <file-name> NOT DUMPED followed by - HAS BEEN REMOVED - IN OUTPUT USE - HAS BEEN ALTERED	The utility is unable to DUMP the specified file for the reasons stated.	Check status of files not dumped, and re-execute the utility.
(NO FILES IN THE FAMILY) <file> or (FILE) ON (BACKUP) or (DISK) FOR (LOAD) or (ADD) or (DUMP) or (UNLOAD)	This generalized message format is used to inform the operator when files are not found during DUMP or UNLOAD, LOAD or ADD.	Check initiating message for correct file-names and re-run utility.
DUPLICATE <file-name> ALREADY BEING (DUMPED) or (LOADED)	The initiating message contained multiple requests to DUMP or LOAD the same file.	None. The utility will continue to execute. Only one copy of these <file-names> will be DUMPed or LOADED.
NO FILES TO LOAD	The utility is unable to find any of the specified files on the back-up disk or sequence of back-up disks specified.	Check initiating message and disk-id of back-up disk being used.
<file-name> NOT DUMPED - DATAFILE NOT ON LINE	The data-file of an indexed pair specified in the DUMP initiating message cannot be found. The utility will continue. The key-file is not DUMPed.	Check the input disk for the presence of the specified files. Re-run the utility.

Message	(continued) Possible Cause	Suggested Action
<file-name> - DATAFILE NOT FOUND ON DISK FOR LOAD	The utility cannot find the data-file on the back-up disk for the specified keyfile. The utility will continue but the keyfile is not LOADED.	Check the initiating message and print the back-up disk directory.
DISK <disk-id> NOT AVAILABLE or DISK <disk-id> LOCKED	Disk specified is either not on-line or is in use by another utility using SYSMEM LOCK on this disk.	Check <disk-id> of back-up disk in use or wait until other utility has gone to EOJ.
FILE LIST FOR <family- name> MAY BE INCOMPLETE or FILE LIST MAY BE INCOMPLETE	If DUMPing from an unrestricted pseudo-disk and a disk belonging to this pseudo-pack is not found, then the files DUMPed may not be a complete list.	Check PRINT.DIR listing and re-run utility if required.
UTILITY LIMIT REACHED AT FILE <file-name>	When DUMPing from an unrestricted pseudo-pack the number of files has reached the maximum, 2804. DUMPing will now commence.	Information only.
OK TO PO DRIVE Dxy or OK TO PO DISK <disk-id>	Informs the operator that it is safe to PO a disk when DUMPing or LOADING and the next disk in a sequence is required.	PO specified disk.
OVERFLOW DISK REQUIRED TO CONTINUE DUMP	Follows first of above pair of messages to denote that another disk is required to continue the DUMP in progress.	Select required disk and continue.
DISK <disk-id> REQUIRED TO CONTINUE LOAD	Informs operator of the next disk in sequence required to continue the LOAD.	Select the required disk and continue.
OPEN DISK UNLABELLED NOT IMPLEMENTED	The utility is unable to open the disk unlabelled as the level of MCP in use does not support this function.	Check version number of MCP in use.
DISK <disk-name> IS NOT REQUIRED FOR THIS LOAD	The files to be LOADED do not reside on this disk.	See next message.
MAKE DISK <disk-id> AVAILABLE AND ENTER "OK" or MAKE ALTERNATE DISK AVAILABLE AND ENTER NAME	Follows the above message to inform operator that an alternate disk is required to continue the LOAD.	Select appropriate disk and continue.

Message	(continued) Possible Cause	Suggested Action
ENTER A FOUR CHARACTER DUMP IDENTIFICATION PREFIX	Informs the operator to enter the four character disk-id prefix before starting the DUMP.	Enter the required four character prefix using the AX command.
DISK LABEL OF Dxy NOT IN CMS FORMAT	Sector 0 of the disk specified for DUMPing to is not in CMS format.	Check device assigned or check the disk located in that device.
(WRITE) or (READ) ERROR ENCOUNTERED ON DRIVE Dxy RANGE <hex> - <hex>	The utility has encountered a read/write error while DUMPing or LOADING.	See next message.
CONTINUING PROCESS	The utility has been able to recover.	No action necessary.
I/O ERROR DETECTED IN WRITE TO FILE <file-name> or I/O ERROR DETECTED IN READ FROM FILE <file-name>	The utility has detected an irrecoverable I/O error. The LOAD or DUMP of this file will terminate but the utility will continue with the next file.	At EOJ, check the file in error and re-run the utility.
ERROR AT RECORD NUMBER <decimal-number> <file-name> NOT (DUMPED) or (LOADED) DUE TO I/O ERROR	Follows the above message to inform operator of the record number in error. Follows above messages to inform the operator that the DUMP or LOAD is incomplete due to an I/O error.	Check missing files and LOAD or DUMP from previous level of back-up.
DIRECTORY ERROR ON DISK Dxy < disk-name >	The utility has detected an I/O error in the directory of a back-up disk during LOAD or DUMP. The utility terminates.	Replace back-up disk if DUMPing. If LOADING, the previous level of back-up disk will be required.
READ ERROR ON PPIT OF Dxy < disk-name >	A read error has been encountered in the PPIT of a pseudo-pack. The utility will terminate without LOADING or DUMPing.	RF or IN pseudo-pack if necessary.
DISK <disk-id> DOES NOT BELONG TO THIS LOAD	A disk from the wrong DUMP sequence has been used. The utility will terminate.	Select the correct disk and re-run the utility from the disk just lected.
FILE <file-name> NOT LOADED - (ABNORMAL TERMINATION) or (IN USE) or (REMOVED) or (INPUT PARITY)	During a LOAD the utility is unable to LOAD a file. The back-up disk does not contain the specified file. During the DUMP which created the back-up disk in use, the	If a copy of the file in question is required, it will be necessary to LOAD from an earlier level of back-up.

Message
or (OUTPUT PARITY)
or (ALTERED) WHEN
DUMPING

(continued)
Possible Cause
DUMP of that file was
terminated because of the
reason given.

Suggested Action

ECMA.LD (Load/Dump of ECMA Tape Files)

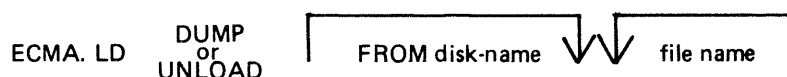
This utility allows the operator to structure tape files according to ECMA BASIC and ECMA COMPACT systems as specified in the STANDARD ECMA-41 publication.

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)

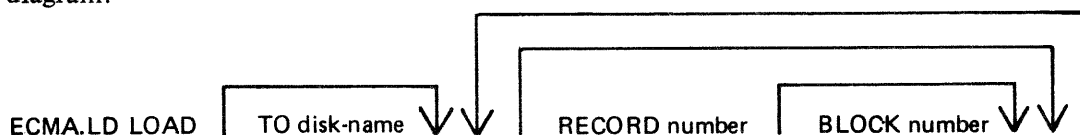
Format diagram:



The files are copied from the disk to a purged tape. If the option UNLOAD is specified then the files copied are removed from the disk.

Format (Tape to Disk copy)

Format diagram:



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 then 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 then 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)

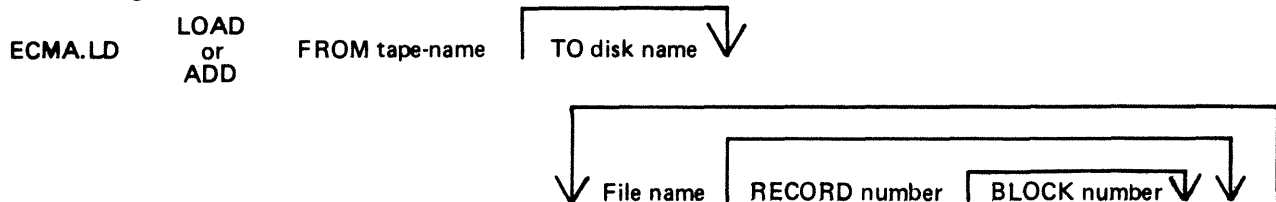
Format diagram:



The files are copied from the disk to tape. If the option UNLOAD is used then files copied are removed from the disk.

Format (Tape to Disk copy)

Format diagram:



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 from tape FRED on CTA which is created in ECMA COMPACT format files IN001, IN002 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 RE-  
QUIRED
```

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 group-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

Message	(continued) Possible Causes	Suggested Action
file-name DUMPED	This message is displayed for each file dumped or unloaded	none
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 initiating message for FILEUTL is :

RN FILEUTL

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).

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; unlike CANDE, no work file is created.

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 #22 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 |
| Screen 9/11. | 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 4-5.

Key to Screen Formats

ffffffffff	File identifier
dddddd	Disk identifier
ssssss	System disk identifier
rrrr	Record size
bbbb	Block size
nnnnnn	Actual no. of records in file
mmmmmm	Maximum possible no. of records in file
xxxxxx	Current record no.
yyyyyy	Current record no. + 1
zzzzzz	Actual no. of records in file + 1
aaaaaaa	Data from record (ASCII)
hhhhhhh	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

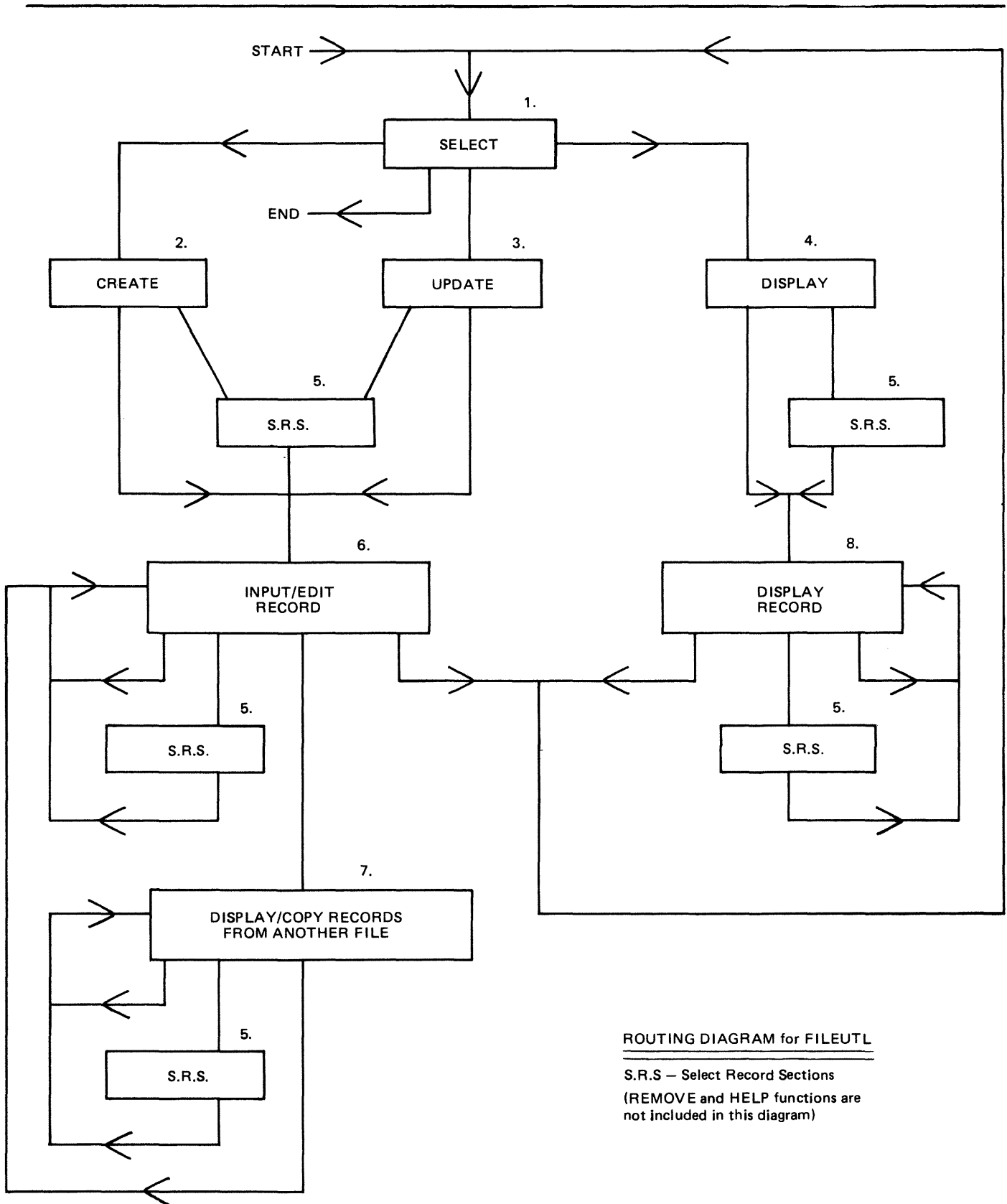
1. The data, as displayed in any unprotected data fields preceding the field where the cursor is currently situated, is saved.
2. If the cursor is in the first character position of an entry field, the data in this field as originally displayed is saved. If not, the contents of the field up to the cursor are saved and the rest of that field is assumed to be blank filled.
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 the Screen 6 section.

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 identify-



ROUTING DIAGRAM for FILEUTL

S.R.S – Select Record Sections
 (REMOVE and HELP functions are not included in this diagram)

Figure 4-4. Routing Diagram for FILEUTL

ing 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 11 – HELP 3.

Screen 1 – Select

```
-----  
CMS UTILITY - FILEUTL      VERSION 3.04.nn      date.and.time  
  
ENTER "1" TO CREATE A NEW FILE  
      "2" TO UPDATE AN EXISTING FILE  
      "3" TO DISPLAY AN EXISTING FILE  
      "4" TO REMOVE AN EXISTING FILE  
      "5" IF HELP IS REQUIRED  
      "6" TO FINISH THIS PROGRAM RUN          > <  
  
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  
      "2" FOR DATA - INPUT IN ASCII  
      "3" FOR DATA - INPUT IN HEXADECIMAL    >1<  
  
ENTER FILE IDENTIFIER          >          <  
  
ENTER NAME OF DISK ON WHICH FILE RESIDES OR WILL RESIDE >ssssss<  
  
.....  
-----
```

If function 1, 2 or 3 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 9, 10 and 11).

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                ffffffff  
  
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 - ASCII IF ONE CHARACTER  
                                HEX IF TWO CHARACTERS >00<  
  
ENTER "1" TO COMMENCE FILE CREATION  
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 11 – 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 – Display

```
-----  
FILE IDENTIFIER                ffffffff  
  
NAME OF DISK ON WHICH FILE RESIDES  dddddd  
  
CREATION DATE/LAST ACCESS DATE OF FILE  11 JUL 81/14 AUG 81  
  
RECORD SIZE/BLOCK SIZE OF FILE      rrrr/bbbbb  
  
NO. OF RECORDS IN FILE              nnnnnn  
  
MAXIMUM NO. OF RECORDS FILE CAN CONTAIN  #####  
  
FILE TYPE                        <00> DATA  
  
ENTER "1" TO COMMENCE FILE MODIFICATION  
      "2" TO SELECT RECORD SECTIONS  
      "3" TO RETURN TO FUNCTION SELECTION  >1<  
  
ENTER "1" FOR ASCII INPUT MODE  
      "2" FOR HEXADECIMAL INPUT MODE      >1<  
  
ENTER RECORD FILLER - ASCII IF ONE CHARACTER  
                      HEX IF TWO CHARACTERS  >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 – Update

```
-----  
FILE IDENTIFIER                ffffffff  
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 FILE DISPLAY  
      "2" TO SELECT RECORD SECTIONS  
      "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
      "2" TO DISPLAY COMPLETE RECORD   >1<

IF "1" IS NOT ENTERED, ALL OTHER INPUT FROM THIS SCREEN WILL BE IGNORED.

UP TO 13 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'S > <
START POSITION OR THE END OF THE RECORD. IF THIS NUMBER OF > <
CHARACTERS EXCEEDS yy, 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.

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.

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  
      "2" TO RE-SELECT RECORD SECTIONS          >1<  
      "3" IF NO OTHER FILES ARE REQUIRED  
ENTER IDENTIFIER OF FILE CONTAINING THE REQUIRED RECORDS >      <  
ENTER NAME OF DISK ON WHICH THIS FILE RESIDES          >ssssss<  
ENTER REQUIRED RECORD NUMBER, LAST IN FILE IS 111111    >1    <  
  1  >aaaaaaaaaaaaaaaaaaaaa<  
 21  >aaaaaaaaa<  
 31  >aaaaaaaaaaaaaaaaaaaaa<  
 56  >aaaaaaaaaaaaaaaaaaaaa<  
 81  >aaaaaaaaaaaaaaaaaaaaa<  
 99  >aa<
```

```
TO COPY THIS REC. OR RANGE FROM THIS REC, ENTER END REC. NO. >      <  
FILE ffffffff      MAX. POSSIBLE SIZE IS tttttt RECORDS  
DISK ddddddd      SIZE OF FILE NOW IS tttttt 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 ffffffff          MAX. POSSIBLE SIZE OF FILE IS ##### RECORDS
DISK ddddddd          ACTUAL SIZE OF FILE IS nnnnnnn RECORDS

ENTER "1" TO DISPLAY RECORD
      "2" TO RE-SELECT RECORD SECTIONS           >1<
      "3" IF NO MORE RECORDS ARE TO BE DISPLAYED
IF "1" IS NOT ENTERED, THE NEXT RECORD NUMBER WILL BE IGNORED.

CURRENT RECORD NUMBER IS xxxxxxx
  1 >aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa<
 21 >aaaaaaaaaaaa<
 26 >hh<
 27 >aaaaaaaaaaaa<
 32 >hh<
 33 >aaaaaaaaaaaa<
 38 >aaaaaaa<
 41 >hh<
 42 >hh<
 43 >hh<
 44 >aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa<
 60 >hh<

NEXT RECORD NUMBER TO BE DISPLAYED IS >yyyyyyy< - ALTER IF NECESSARY
.....
-----
```

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 "SYSTEMEM".

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 MODIFIED. 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.

TRANSMIT TO CONTINUE

"RECORD FILLER" FIELD - WHEN CREATING OR UPDATING A DATA TYPE FILE,
THE USER MAY ENTER A CHARACTER WHICH WILL FILL RECORD ENTRY FIELDS IN
THE 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 HEX. 20-5D/60-7F, 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 MODIFY 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. AN UNDERSCORE OR UP-ARROW CHARACTER WILL BE CONVERTED TO A ?

TRANSMIT TO CONTINUE

Screen 11 – 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, HIT 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 DISPLAYED 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.

HIT "HOME" KEY, ENTER "0", HIT "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/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.

TRANSMIT TO RETURN TO FUNCTION SELECTION

If the HELP function is selected in screen 1, the utility displays screens 9, 10 and 11 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	1-8
2. ERROR – HIGHLIGHTED ENTRY MUST NOT BE BLANK	1-8
3. ERROR – HIGHLIGHTED ENTRY MUST BE NUMERIC	1-8
4. ERROR – HIGHLIGHTED ENTRY SHOULD NOT BE GREATER THAN xxx	1-8
5. ERROR – HIGHLIGHTED ENTRY SHOULD NOT BE LESS THAN xxx	7
6. ERROR – HIGHLIGHTED ENTRY MUST NOT BE ZERO	1-8

(continued)

Field Entry Error Messages	May be displayed on screens:
7. ERROR – BLOCK SIZE MUST BE A MULTIPLE OF RECORD SIZE	2
8. ERROR – INVALID CHARACTER IN IDENTIFIER	1,7
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

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.

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 ffffffff ON DISK ddddd	1,7
2. FILE ffffffff ON DISK ddddd IS INCOMPLETE	1
3. FILE ffffffff ON DISK ddddd IS IN USE	1,7
4. DISK ddddd IS NOT AVAILABLE	1,7
5. DISK ddddd IS LOCKED	1,7
6. FILE ffffffff ALREADY EXISTS ON DISK ddddd	1
7. FILE ffffffff ON DISK ddddd IS NOT SOURCE OR DATA TYPE	1
8. FILE ffffffff ON DISK ddddd IS NOT SOURCE TYPE	7
9. FILE ffffffff ON DISK ddddd IS NOT DATA TYPE	7
10. RECORD SIZE OF FILE ffffffff ON DISK ddddd IS NOT rrr	7
11. FILE ffffffff ON DISK ddddd HAS NO RECORDS	1,7
12. RECORD SIZE OF FILE ffffffff ON DISK ddddd IS GREATER THAN 2000	1
13. DIRECTORY OF DISK ddddd IS FULL	1

Remove Function Message

1. FILE ffffffff REMOVED FROM DISK ddddd

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 ffffffff ON ddddd STARTING AT REC.
yyyyyy

Copy Error Messages

1. NO RECORDS COPIED-BAD RECORD xxxxxxx IN FILE ffffffff ON ddddd
2. ONLY xxxxxxx RECORD<S> COPIED-BAD RECORD yyyyyy IN FILE ffffffff ON
dddddd
3. RECORD xxxxxxx IN FILE ffffffff ON ddddd 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 ffffffff ON ddddd CANNOT BE EX-
CEEDED

No Disk Space Message

1. >>>>> WAITING FOR SPACE FOR FILE ffffffff ON DISK ddddd.....

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 ffffffff ON DISK ddddddd.

RECORD xxxxxxxx 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.

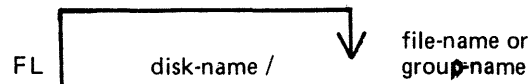
THE NEXT SCREEN TO BE DISPLAYED IS FOR FUNCTION SELECTION.

The utility then displays Screen 1.

FL (Display File Attributes on Self-Scan)

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.

Format diagram:



The utility uses the following PKs when more than one file is specified:

PK1	PK2	PK3	PK4	PK4	PK6
page to next screen	-	-	-	-	EOJ

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. FL uses the 1920 character screen in this six-line format:

```
Line 1 FILE ddddddd/ffffffffffff filetype @xy@
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 : ***** UNIT : uuuuuuu
Line 6 OVERFLOW DISK : ooooooo FILE INCOMPLETE
UNRESTRICTED PSEUDO
```

The 256 character screen is used in the following manner:

```
Line 1 FILE NAME: ddddddd/ffffffffffff
Line 2 FILE TYPE: filetype @xy@
```

2015228

Line 3 SIZE: ACTUAL xxxxxxxx, MAX xxxxxxxx
 Line 4 CREATED ACCESSED REC.SZ RECS/BLK
 Line 5 yyddd yyddd xxxxx xxxxx
 Line 6 AREA MAP: *****
 Line 7 OVERFLOW DISK: ooooooo (or UNRESTRICTED PSEUDO)
 Line 8 FILE INCOMPLETE UNIT: uuuuuuu

Note that the OVERFLOW ON DISK will not be displayed if the file has no overflow areas allocated.
 The first line contains disk name specified by DDDDDDD on which the file specified by FFFFFFFFFF 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:	
	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

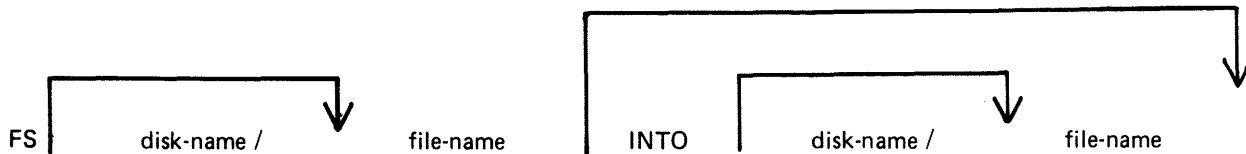
Message	Possible Causes	Suggested Action
FILE NAME = disk-name/ file-name NOT FOUND IN DISK DIRECTORY	Specified file not found on this disk.	Check input and re-input if necessary. Check for correct disk.
disk-name/file-name NO FILES FOUND IN DIRECTORY FOR FAMILY	Specified group was not found on this disk.	Check input and re-input if necessary. Check for correct disk.

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@ are 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 diagram:



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.

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 Causes	Suggested Action
file-name SQUASHED FROM n RECORDS TO m RECORDS	FS successful. Original and resulting filesizes of the data file indicated. “n” and “m” are decimal numbers up to 7 digits each.	None

Message	(continued) Possible Causes	Suggested Action
KEYFILE file-name RECONSTRUCTED	“file-name” indicated by operator input identified by keyfile and FS has successfully squashed the data file and reconstructed the keyfile.	None
KEYFILE SORT FAILURE	SORT utility was not able to construct a keyfile properly.	None
RECORD SIZE TOO GREAT	An attempt has been made to squash a file with a record size greater than 10000 bytes.	None
ZIP TO SORT FAILURE – END OF JOB	When attempting to reconstruct the keyfile, the utility failed to execute SORT.	Ensure that SORT and SORTINTRINS are present on the disk, and re-input.

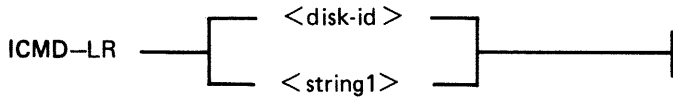
Note: Refer to “Command Utility Output Messages” for additional aid.

ICMD

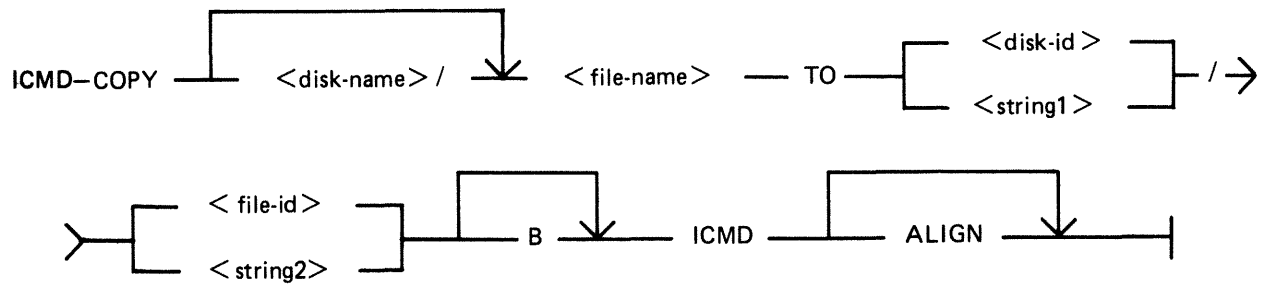
The ICMD utility allows the operator to access industry compatible mini disks.

Railroad diagram:

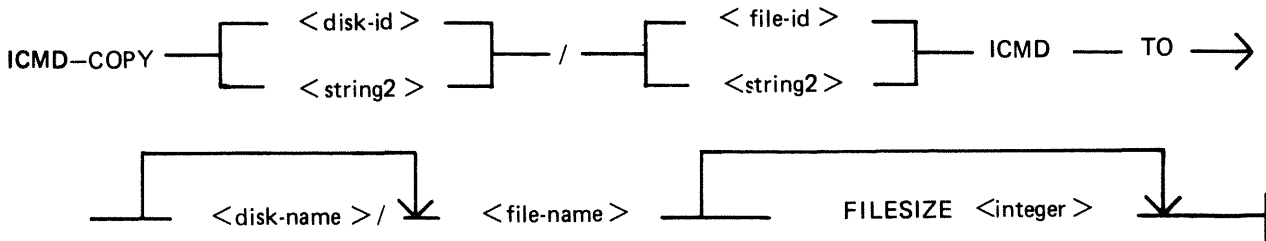
Version 1



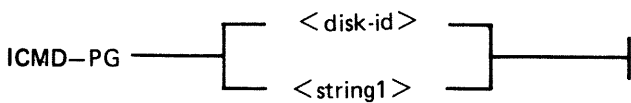
Version 2



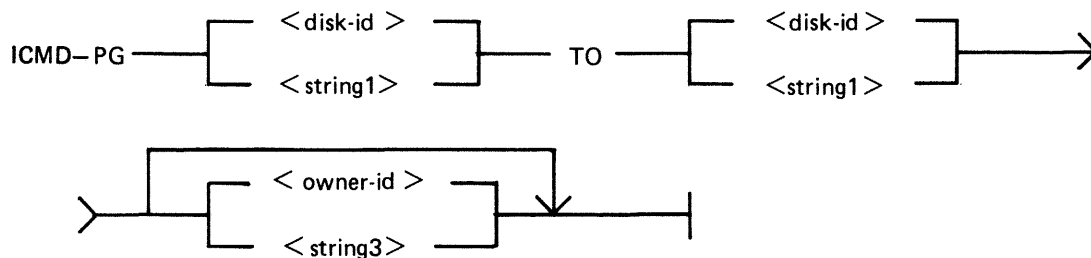
Version 3



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
ENTER VOL-ID FOR "file-name" n	Program is about to display an ACPT requesting the next disk-name containing the Multi-Volume File "file-name". "n" is the sequence number to be used, or blank if none.	Enter appropriate disk-name.
VOLUME n OF "file-name" OUT OF SEQUENCE	Sequence number "n" of "file-name" was not expected at this point. (Ex: program was expecting disk with sequence number of 2 and operator supplied disk with sequence number 4).	Remove inappropriate disk from disk drive (PO cannot be used with ICMD). Replace with disk having correct sequential number for this file.
DUPLICATE FILE "file-name"	File cannot be copied to ICMD as a file of the same name already exists. Disk cannot contain 2 files with the same name.	Copy this file while changing its name. For example, if file-name MYFILE is already on the ICMD, enter ICMD COPY DISK/MYFILE TO ICDISK/YOURFILE ICMD.
NO FREE LABEL ON "disk-name"	File cannot be copied to ICMD as its disk directory is full.	Replace ICMD with another ICMD having directory space. Re-attempt ICMD utility;
NO SPACE FOUND ON "disk-name"	File cannot be copied to ICMD as no unused space could be found.	PG ICMD and re-attempt copy; or select another disk on which to copy.
Note: if any of the above messages (including "disk-name" NOT FOUND and "file-name" NOT FOUND) is output in response to the initiating message, the utility will go to End of Job. If in response to an ACPT, the utility will repeat the message: ENTER VOL-ID FOR "file-name" "n".		
BAD INDEX TRACK ON disk-name SECTOR number	ICMD has an error on track 0 and cannot be used by this utility. Utility goes to End of Job.	Select another ICMD for this function.
RECORD SIZE OF file-name EXCEEDS 128	The file cannot be copied to the ICMD as its record size is more than 128, the maximum	Use MCP COPY utility to change record/block sizes. (Ex: COPY CMSFILE TO CMSFILE

	for an ICMD. Utility goes to End of Job.	RECORD 128, BLOCK 128) Then use ICMD COPY to complete transfer.
NEXT SEQUENCE NUMBER OF file-name EXCEEDS 99	No more disks may be used, as the maximum for an ICMD is 99. Utility goes to End of Job.	Use MCP COPY utility to break file (see COPY utility). Then use ICMD COPY to complete transfer.
INPUT ERROR ON file-name SECTOR number	Irrecoverable error was found on reading sector "number" from the ICMD. Utility goes to End of Job.	File cannot be copied from the ICMD; use backup ICMD if available.
OUTPUT ERROR ON file-name SECTOR number	Irrecoverable error was found on writing sector "number" to the ICMD.	Disk cannot be used. Use another ICMD.
file-name TO file-name COPIED	ICMD COPY function successful.	None.
disk-name PURGED	ICMD PG function successful.	None.
file-name IS NOT TYPE SOURCE OR DATA	The requested file has the wrong file type for this use.	None.
file-name REMOVED	The CMS file was removed by the utility to make way for another file being copied from an ICMD.	None.
INVALID CHARACTER IN IDENTIFIER <invalid string> - WARNING	This is a warning that the identifier <invalid string> does not comply with ICMD standards.	None.
FILESIZE REQUIRED FOR <file-name>	The attribute FILESIZE must be specified when copying a multi-volume ICMD file if it was created without specifying B.	Re-enter with filesize option in initiating message.
CANNOT CREATE A FILE 'DATA' ON ICMD <ICMD-name> - RESERVED NAME	The file name DATA is a reserved name.	None.
FILESIZE OF DISK FILE <file-name> TOO SMALL	The filesize specified in the FILESIZE attribute is too small. Increase filesize accordingly.	Re-enter with larger filesize.

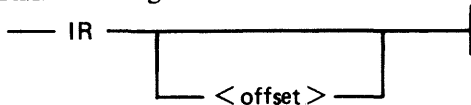
Refer to Common Utility Output Messages for additional aid.

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.

Railroad diagram:



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

Output Messages

Message	Possible Causes	Suggested Action
CANNOT LOCATE DESIRED LOG ENTRY	The log files have been transferred using TL utility. Offset is greater than number of entries in log file.	None Decrease the value of offset and re-enter.

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.

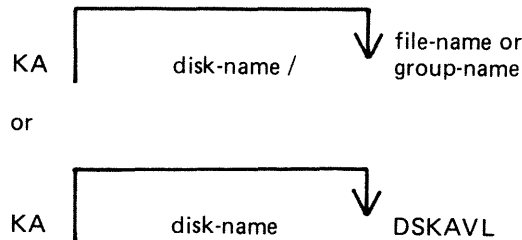
If files are created, extended or deleted by the system during the processing of KA, the map will not be accurate. It is therefore necessary that KA be run only when no other programs are in the mix.

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:

Format diagram:



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
AREA APPARENTLY ASSIGNED TWICE	Area is contained partly or completely in an area previously listed.	Notify field engineering. DO NOT USE DISK. It may be possible to recover any file not involved, using COPY.
AREA ASSIGNED BEYOND MAXIMUM ADDRESS	Area is assigned beyond the addressable space.	Notify field engineering. DO NOT USE DISK.
NOTE: OUTPUT FILES ON DISK WERE OPEN DURING THIS EXECUTION OF KA	Files were open on disk while KA was processing. Other programs may have been in the mix when KA was processing.	KA printout may not be completely accurate. Begin KA when no other programs are in the mix.
input NOT FOUND IN DIRECTORY ON THIS DISK	Specified file is not on disk.	Check input and re-input if necessary; Check for correct disk.
input NO FILES IN DIRECTORY FOR THIS FAMILY	Group of files specified is not on disk.	Check input and re-input if necessary; Check for correct disk.
TABLE SIZE EXCEEDED	Number of lines of output required is greater than KA permits. Maximum permitted is about 141 pages of output. KA ends.	None.
NO OUTPUT GENERATED BY KA	Disk directory contains no file names to print.	Check for correct disk; Check input and re-enter if necessary.

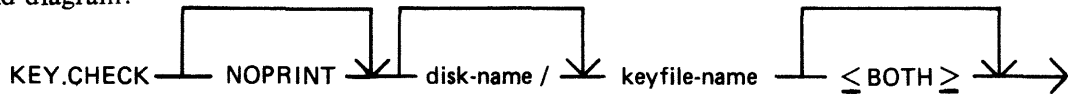
Note: Refer to "Common Utility Output Messages" for additional aid.

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:

Railroad diagram:



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 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
file-name IN USE	Another program is using the keyfile.	Wait until job using keyfile has closed file, then re-input.
file-name IS NOT A KEYFILE	Self-explanatory.	Check the file-name and re-input.
file-name IS A NULL-KEY KEYFILE	The specified file is a tag-file.	See SOG section on the SORT for meanings; this utility cannot check tagfiles.
file-name <DATAFILE> NOT FOUND	The data file corresponding to the keyfile cannot be found.	<ol style="list-style-type: none">1. Enter LR disk-name/keyfile-name.2. Check the disk name of the data file in LR listing.3. If the disk file name is the same as the keyfile, then copy data file from backup and re-input.4. If the disk name is different, then load that disk and re-input.

Message	(continued) Possible Causes	Suggested Action
file-name <DATAFILE> IN USE	The corresponding data file is being used by a program.	Wait until the other job has closed the data file, then re-input.
file-name INCOMPATIBLE WITH KEYFILE	The key-offset plus key-length in the KFPB of the specified keyfile is greater than the record size of the data file.	None.
READ ERROR ON KFPB OF file-name	Parity errors have been encountered: the utility cannot continue.	1. Attempt to copy files to another disk. 2. Run CHECK.DISK to check the integrity of the disk.
READ ERROR ON KEYFILE file-name	Same as above.	Same as above.
UNABLE TO OPEN file-name <INDEXFILE>	<BOTH> has been given in the initiating message and the utility is unable to open the keyfile and data file as an indexed pair after completion of the keyfile to datafile check.	None.
=INVALID KEY= KEY IN DATA RECORD: invalid key RECORD datafile record number	A data record has occurred with an invalid key.	None.
file-name FILETYPE IS NOT DATA	The file identified in the KFPB of the specified keyfile as the associated datafile does not have a filetype @00@.	Check the file-name and re-enter.

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

If NOPRINT is specified in the initiating message, the following messages can be displayed by the utility:

NOTE

As the following messages are for information only, no suggested corrective action is given.

Message	Possible Causes
KEYFILE keyfile-name PERTAINS TO DATAFILE file-name	Shows which data file is related to which keyfile.
WARNING – GENERATION NO. MISMATCH BETWEEN KEYFILE keyfile-name AND DATAFILE datafile-name	The generation numbers of the file are inconsistent.
FIRST DISCREPANCY FOUND BETWEEN KEYFILE keyfile-name DATAFILE datafile-name	The first error has been encountered between the key and data files.

(continued)

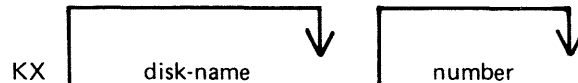
Message	Possible Causes
x DISCREPANCIES FOUND WITH KEYFILE keyfile-name	The check on the keyfile has been completed and x number of discrepancies have been found between the keyfile and its associated data file.
KEY.CHECK ON keyfile-name COMPLETED	<BOTH> was not specified in the initiating message. No checking will be invoked on the data file and the utility will go to End of Job.
KEY.CHECK COMMENCING ON DATAFILE file-name	<BOTH> was specified in the initiating message and the datafile check has been invoked.
y DISCREPANCIES FOUND WITH DATAFILE file-name	The check on the data file in relation to the keyfile has been completed and y discrepancies have been found.
KEYFILE keyfile-name IS CONSISTENT WITH DATAFILE file-name	No discrepancies have been found on either the datafile or keyfile. If the generation numbers of the files did not match that of the datafile, then either “GENERATION NO. OF keyfile-name MODIFIED TO Z” or “UNABLE TO MODIFY GENERATION NUMBER OF KEYFILE keyfile-name” will be displayed and the utility will go to End of Job.
KEYFILE keyfile-name IS NOT CONSISTENT WITH DATAFILE datafile- name	Discrepancies have been found on one or both files. The utility will go to End of Job.

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.

Format diagram:



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.

Examples:

To display the name of the first file on the system disk whose size is equal to or greater than 250 sectors:

```
KX 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

See SYS-SUPERUTL messages also.

Message	Possible Causes	Suggested Action
END KX	KX successful	None
filename – number SECTORS; AV. number, TEMP (or TEMPORARY) number	Normal KA display showing current number of sectors, available and temporary on disk.	None
input IS AN UNACCEPTABLE RESPONSE FOR KX	Typing error	Check input and re-enter

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.

Railroad diagram:

— LB ————|

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
IR REQUIRED BEFORE LB CAN BE DONE	IR function was not initiated before LB.	Initiate IR then re-input LB.
CANNOT RECALL LB BEYOND THIS POINT	LB has reached the beginning of this log file.	None

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 as regards 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.

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.

Railroad diagram:

— LF ————|

Example:

To look forward from last recall and display messages:

LF

LF can be initiated only after IR, LB, and LF.

Output Messages

Message	Possible Causes	Suggested Action
IR REQUIRED BEFORE LF CAN BE DONE	IR was not initiated before LF	Initiate IR and then re-enter LF
CANNOT RECALL LF BEYOND THIS POINT	LF has displayed the most recent entry in the log file	None

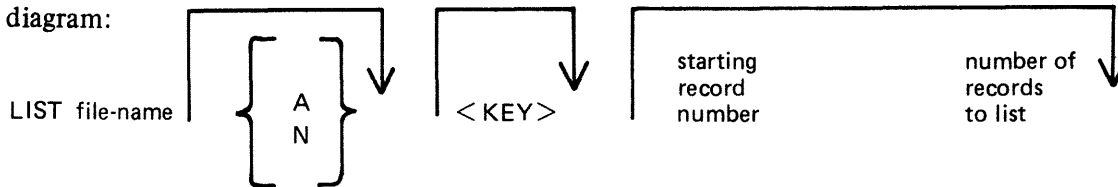
LIST (File List)

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.

Format:

Format diagram:



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 are 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

Further features of this utility are summarized in the railroad chart given in figure 4-5, which gives the complete input specifications.

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
PTP – paper tape

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. Upon 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.

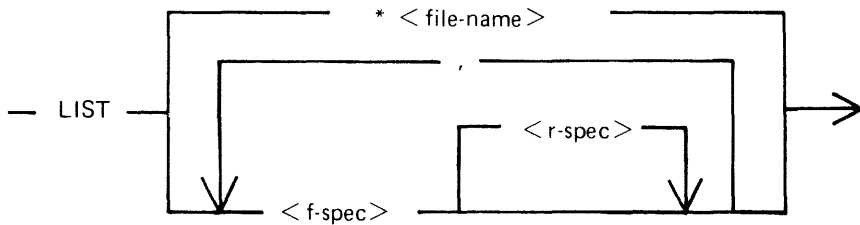
If the RECORD size is not 180 bytes, refer to the section on Record/Block modifications.

Example:

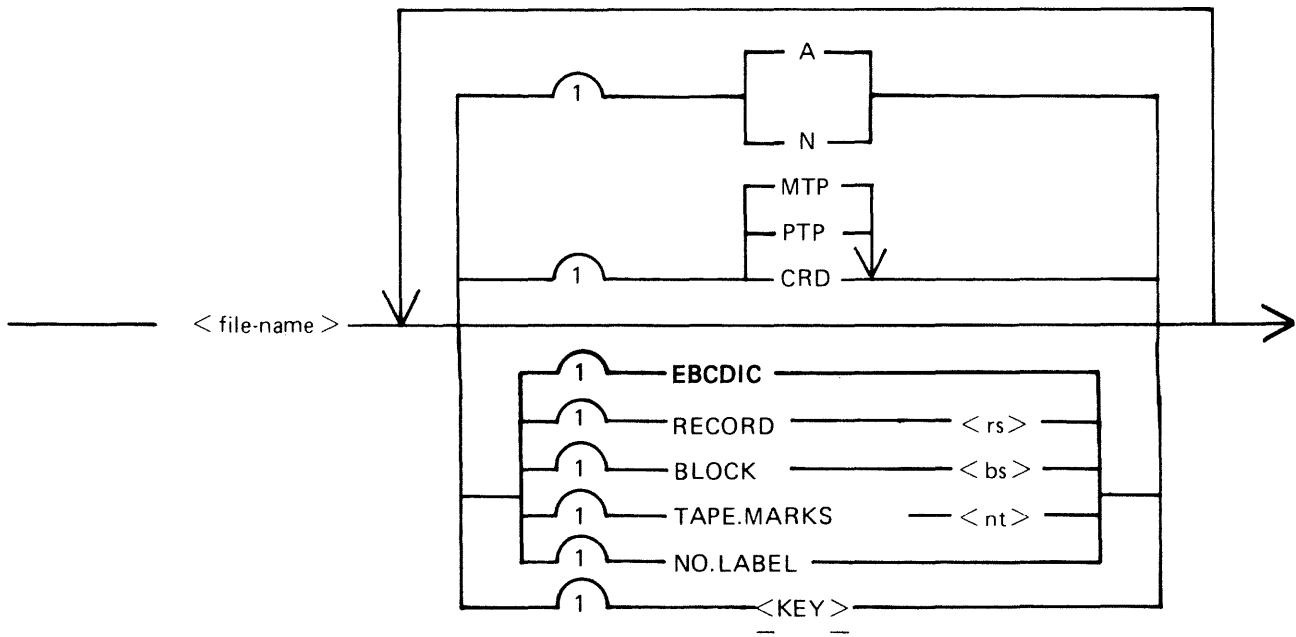
To list the first file of a magnetic tape with non-standard label (the format being: label, tapemark, data, tapemark):

```
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.



< f-spec > is defined as :



< r-spec > is defined as :

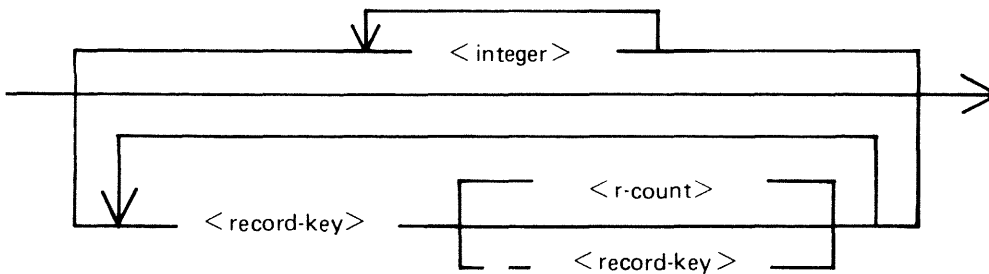


Figure 4-5. Railroad Chart for LIST Utility

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 2 options: the number of records can be specified or an ending key value.

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:

2015228

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).

Output Messages

Message	Possible Causes	Suggested Action
number number IN filename NOT LISTED	Error found in pair of numbers for selected record list option. One record number in the pair indicates a section of the file at a lower file address than a previously specified section. This pair is ignored by LIST.	Check input and re-enter.
filename EXHAUSTED DURING number number	End-of-file was encountered while the section of the file indicated was being copied.	Check input and re-enter if necessary.
filename EXHAUSTED DURING RANGE record-key record-key or filename EXHAUSTED DURING RANGE record key number	End-of-file encountered while section of file indicated is being listed; or no records were found in range of the 2 record keys.	Check input and re-enter if necessary.
NO RECORDS FOR LISTING FROM filename	Specified file contains no records to list.	Check for correct filename.
filename NOT ACCEPTABLE – RECORD SIZE OF number EXCEEDS MAXIMUM FOR THIS RUN – RESUBMIT	LIST has encountered a file within a record size greater than expected. This can happen if a magnetic tape file with a record size greater than 1024 characters is submitted to the utility without the record size being properly specified in the initial input.	Check input and re-enter.
KEYFILE filename OR DATA FILE NOT FOUND	Utility cannot locate keyfile or data file pertaining to the keyfile specifications.	Check for correct medium.
BAD ATTRIBUTE SPECIFIED	The number specified after BLOCK is not an integer multiple of the number specified after RECORD.	Correct the input and re-enter.

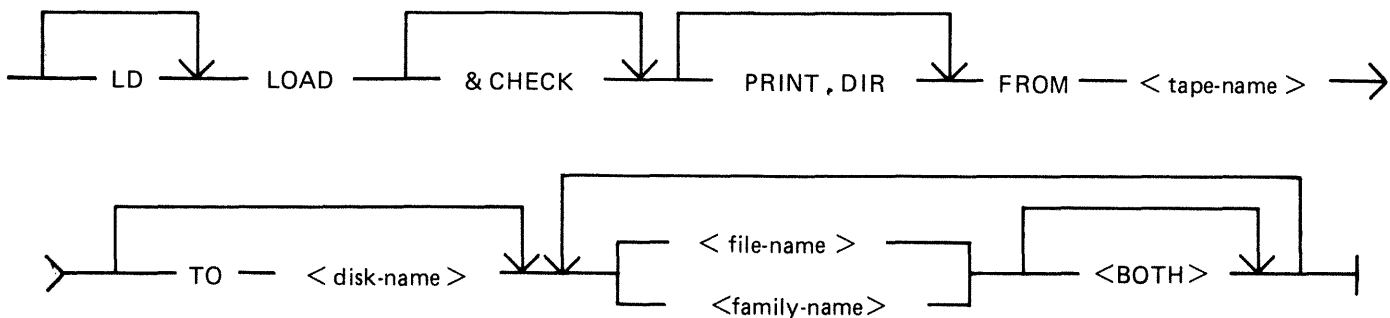
Message	(continued) Possible Causes	Suggested Action
PERMANENT ERROR ON INPUT FILE filename (NO. number or KEY key)	The irrecoverable error, e.g. parity error has been found.	None.
SELECTIONS OF RECORDS FROM number IGNORED	The file reached end before the record selected.	None.
RECORD SIZE TOO GREAT	The specified record size is greater than 10000.	None.
INVALID CHAR. IN IDENTIFIER filename – WARNING	A tape with an illegal name is specified.	Correct the name and re-input.
KEYFILE filename OR DATA FILE IN USE	The specified keyfile and its data file are on line, but one or both are in use by a task which will not permit sharing.	Wait until the task using the keyfile or data file goes to end of job.

Note: Refer to “Common Utility Output Messages” for additional aid.

LOAD (LD – TAPE LIBRARY UTILITY)

This function is part of the utility LD. LOAD 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.

Railroad diagram:



& 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 “,” or a space “ ”.

< 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

Message	Description
<file-name> NOT A RECOGNIZED DUMP TAPE	A tape which has been provided for loading does not have a recognizable format for the LD utility. The utility will go to End of Job. Re-execute the utility with a valid LD format tape.
NO FILE <file-name> ON TAPE <tape-name> FOR LOAD	This file is not present on the tape. The utility continues to look for other files to load, if specified.
NO FILES IN THE FAMILY <family-name> ON TAPE <tape-name> FOR LOAD	No files have been found by LOAD in the family <family-name>. The utility continues to look for other files to load, if specified.
NO FILES TO LOAD	The utility has found that none of the files specified will be loaded. The utility goes to End of Job. Check that the tape has the required file(s).
<file-name> LOAD DISCREPANCY	The end of the file has been reached sooner than expected and implies that there is erroneous information in the Disk File Header for that file. The utility will continue trying to load the rest of the file(s) specified. Try loading this file using another tape unit.
<file-name> REMOVED ON DISK	This file is already present on the specified disk. It was removed after the file was copied from the tape.

(continued)

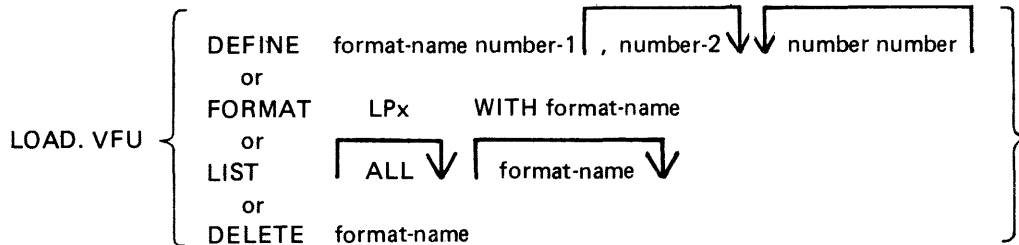
Message	Description
<file name> LOADED	This file has been loaded onto the specified disk.
<file-name> NOT FOUND	The file containing the <parameter-list> (star file) is not present. The utility goes to End of Job. Make the star file available or specify the parameters in the initiating message.
DISK <disk-name> NOT AVAILABLE or with a pseudo-disk arrangement PSEUDO-DISK 'pseudo-disk-id' ON DISK <disk-name> NOT AVAILABLE and if the pseudo-disk is unrestricted, then FILE LIST FOR <family-name> MAY BE INCOMPLETE or FILE LIST MAY BE INCOMPLETE	The specified disk is either not on the system or not ready. The utility will go to End of Job. Ensure that the required disk is on the system and ready before re-running the utility.
<file-name> - DATA FILE NOT FOUND ON TAPE FOR LOAD	The data file for the given key file does not follow on the tape. The data file is not loaded. Note that the key file has been loaded. Check that the data file is on the tape. Reload the data file and SORT it to produce the correct keyfile.
LOAD ABANDONED - ERROR IN DUMP TAPE SHOULD HAVE BEEN PURGED	The tape should have been purged at time. All the files loaded so far will remain, except the file that the utility was attempting to load when the tape error was encountered. The utility goes to End of Job.
INVALID CHARACTER IN IDENTIFIER <file-name> - WARNING	The utility has generated a file name which contains invalid characters. The utility continues.
FILE NOT LOADED - NOT ON TAPE <reason> AT DUMP TIME where <reason> can be: WAS IN USE or HAD BEEN REMOVED or HAD BEEN ALTERED or WAS INCOMPLETE or REASON UNKNOWN or DISK WAS LOCKED In this case, the additional message below will be displayed	The requested file could not be loaded as it was not dumped. The reason for this is explained in the message. If the reason is that the disk was LOCKED, that is, unavailable to LD, then the utility goes to End of Job as there are no further files on the tape. For any other reason, the utility continues trying to load any further requested file(s).
LOAD ABANDONED - ERROR IN DUMP DUPLICATE - <file-name> ALREADY BEING LOADED	More than one request has been made to load this file in the initiating message. The utility will continue. Note this is usually caused by specifying individual files and a family including these files.
ZIP FAILURE WHEN ZIPPING CHECKADUMP FROM LD. ENTER "Y" TO RETRY ZIP, or "N" TO TERMINATE.	"& CHECK" was specified in the initiating message, and LD has attempted to zip CHECKADUMP at the end of the dump operation. Check that CHECKADUMP is on the

Message	(continued)	Description
		system disk; if not, copy it to the system disk and AX "Y" to LD to retry the zip of CHECKADUMP; otherwise, AX "N" to terminate the utility.

LOAD.VFU (Load Vertical Format Unit)

This utility allows the operator to define the page format on a line printer that contains a soft vertical format unit.

Format diagram:



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). The utility zips the VF intrinsic which performs the actual loading of a format string.

When using the DEFINE function, if the file SYSVFU.LIB does not already exist, the utility will create this file and an entry will be made for the format being defined. Otherwise, the file will be updated to contain the newly defined format. The number-1 is the page height in lines and number-2 is the end of page line. The pair of numbers consists of a channel number (2-11) followed by a 'skip to line' number (not greater than the page height).

Example:

```
LOAD.VFU DEFINE PAYROLL 66, 60, 2 10, 4 20
```

This format is equivalent to:

- Page Height = 66
- End of Page = 60
- Channel 2 has associated line number 10
- Channel 4 has associated line number 20

This format is also equivalent to figure 4-6, which illustrates a paper vertical format tape on other line printers.

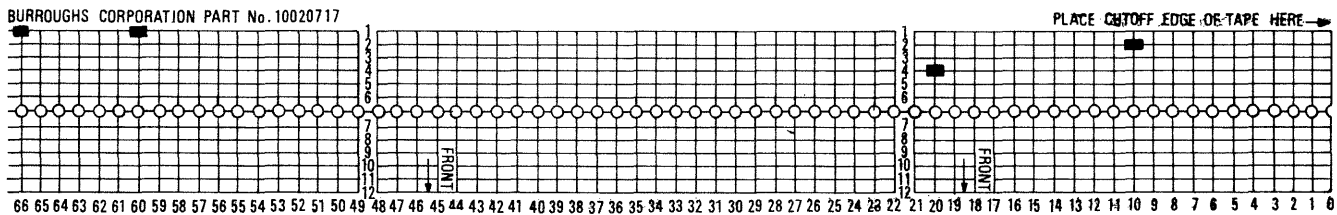


Figure 4-6. Paper Vertical Format Tape

Limitations

The utility does not check the following:

1. For each channel, no more than 4 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.

Example:

To load the format PAYROLL defined earlier on LPA:

```
LOAD.VFU FORMAT LPA WITH PAYROLL
```

The LIST function has three possible options:

```
LOAD.VFU LIST
will list all format-names defined in SYSVFU.LIB;
```

```
LOAD.VFU LIST format-name
will list the format string of specified format-name as specified in SYSVFU.LIB;
```

```
LOAD.VFU LIST ALL
will list all format-ids and strings as defined in SYSVFU.LIB.
```

The DELETE function is used to delete a predefined VFU format from the SYSVFU.LIB file. If the format being deleted is the only defined format in SYSVFU.LIB, then the SYSVFU.LIB file will be removed.

Example:

To delete predefined VFU format PAYROLL from the SYSVFU.LIB file:

```
LOAD.VFU DELETE PAYROLL
```

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 Causes	Suggested Action
SYSVFU.LIB NOT FOUND	Self-explanatory	Use DEFINE function to create library file.
SYSVFU.LIB IN USE	Another copy of LOAD.VFU is running.	Use only one copy of the utility.
READ ERROR ON SYSVFU.LIB	Parity error on the library file.	
WRITE ERROR ON SYSVFU.LIB	Parity error on the library file.	

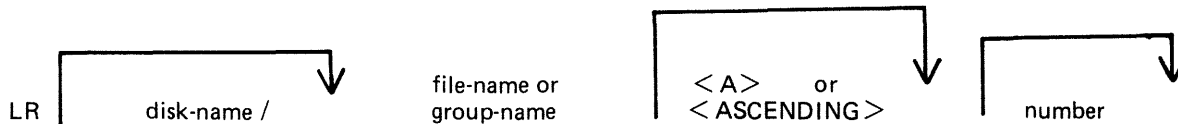
Message	(continued) Possible Causes	Suggested Action
format-name NOT FOUND IN SYSVFU.LIB	Requested DELETE, FORMAT or LIST cannot find specified format-name.	Check input and re-enter if necessary: use DEFINE function to create format-name.
format-name ALREADY DEFINED IN SYSVFU.LIB	Cannot DEFINE two formats of the same name.	Re-input using different format- name.
format-name DEFINED IN SYSVFU.LIB	DEFINE function has been successful.	None
format-name DELETED FROM SYSVFU.LIB	DELETE function has been successful	None
SYSVFU.LIB DELETED	A DELETE function has deleted the only remaining format in the library file, and the file has been removed.	None
PRINTER peripheral FORMATTED WITH format- name	FORMAT function has been successful.	None
PRINTER peripheral NOT FORMATTED WITH format- name	The utility has found the specified format in the library, but the zip of the VF intrinsic was unsuccessful (this will be preceded by output from VF).	Refer to the VF error message.

LR (List Directory)

This utility allows the operator to print detailed information about particular files or groups of files on disk.

Format:

Format diagram:



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 “number” 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 “number”; this will be followed by a listing, with totals, of all available and temporary areas on the disk.

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).
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.

Heading	(continued) Value	Significance
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 displayed with type identification as follows:

Filetype	Entry in Listing
@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@

(YYMMDD = the compile date and SYSDATA could be SYSCONFIG or SYSVFU.LIB)

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.

Output Messages

Message	Possible Causes	Suggested Action
TABLE SIZE EXCEEDED	The number of filenames to be sorted by LR in an ASCENDING or A request has exceeded the maximum permitted (254).	None.

Message	(continued) Possible Causes	Suggested Action
input – NOT FOUND IN DIRECTORY OF THIS DISK	File specified is not on the disk.	Check the correct disk and re- enter if necessary.
input – NO FILES IN DIRECTORY FOR THIS FAMILY	Group of files specified is not on disk.	Check input and re-enter if necessary. Check for correct disk.

Note: 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.

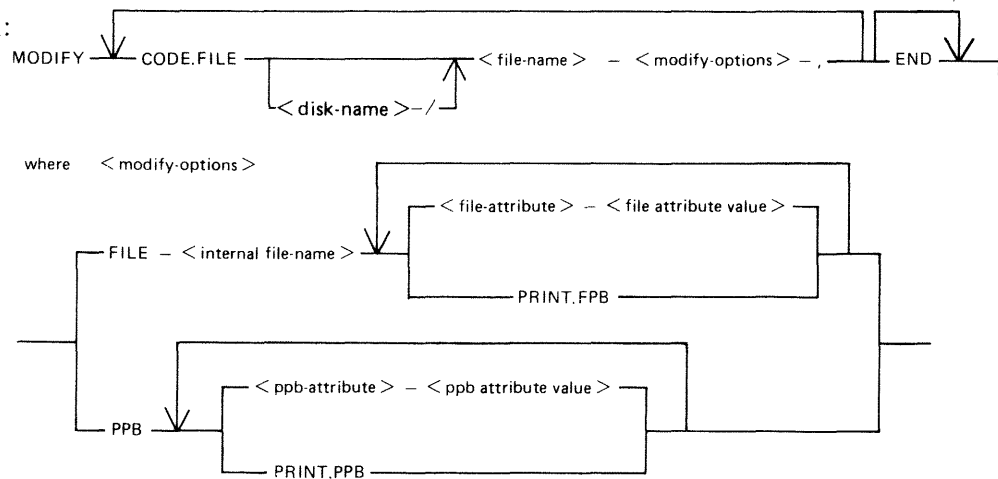
The utility operates in an interactive manner using a console file if no further information is provided when initiating the utility, thus:

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

Railroad diagram:



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:

The second byte of the field in the PPB is machine-dependent and is therefore not printed.

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

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

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	EOJ

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 4-5. The utility displays the current value, then requests
NEW VALUE

Enter the new value required. The utility then returns to the select-function loop.

File Attributes

Table 4-5 gives the keywords for each file attribute that can be changed by the MODIFY utility, together with allowable values for each attribute. Table 4-6 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.
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 4-5 and 4-6 and re-input.
FILE NOT SPECIFIED	Missing keyword “FILE”	Re-input.

Message	(continued) Possible Cause	Suggested Action
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.
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 4-5. 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 4-7.
RECORD	1-5 decimal digits in range 0-65535
BUFFER	1-5 decimal digits in range 0-65535

Table 4-5. File Attributes Accessible by MODIFY
(continued)

File Attribute Name	Allowable Values
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
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.

Table 4-6. 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 4-7. Mnemonics for Device Attribute for MODIFY

Mnemonics	Meaning
PR	Any printer.
KP	Keyboard printer.
KD	Keyboard display.
KB	Keyboard any output.
SP	Serial printer.
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.
PTR	Paper tape reader.
PTP	Paper tape 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.

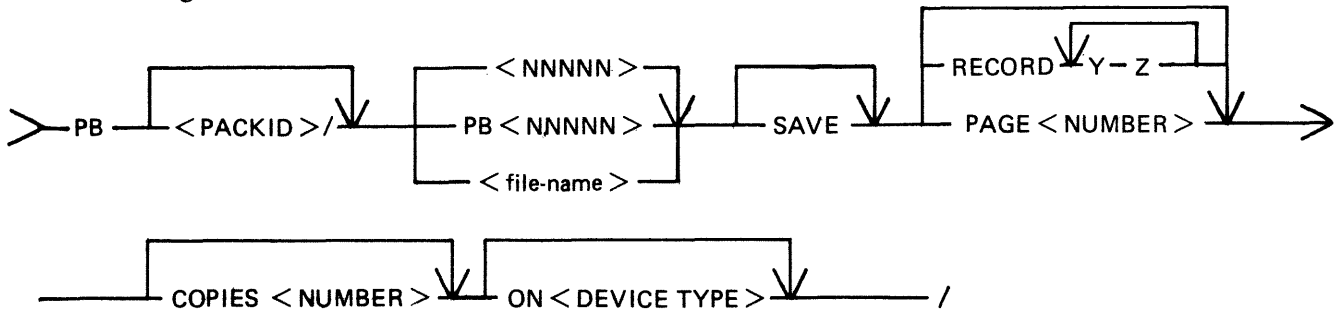
PB (List Printer Backup Files)

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 syntax of the PB initiating message is:

Format 1:

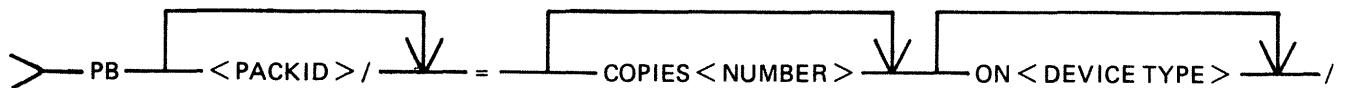
Railroad diagram:



Note: The options "SAVE", "RECORD", "PAGE", "COPIES", "ON" can be entered in any order.

Format 2:

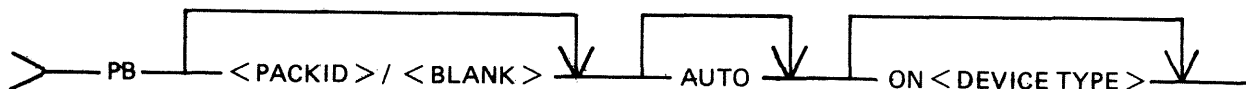
Railroad diagram:



Note: The options "COPIES", "ON" can be entered in any order.

Format 3:

Railroad diagram:



PB Initiating Message Parameters

<PACKID>/ Identification of pack on which the backup file resides. Contains one through seven characters. If no <packid> 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 <packid> 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 the family of 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.
<file-name>	Filename of the named printer backup file to be printed.
SAVE	Prevents the removal of the file from disk after printing is completed. If "SAVE" is not specified, removal occurs.
COPIES	Specifies a non-zero decimal number containing at most two digits. "COPIES cc" specifies the number of copies to be printed. The default value is one.
RECORD	Selects specific parts of the file for printing by giving a starting record number (that is, "Y") and the number of records to be printed (that is, "Z"). More than one range of records may be specified, and overlapping of record number pairs may exist.
	NOTE: When there is no quantity (that is, "Z") 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.
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.
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.
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 AP300XAx. Their use is described in DATACOMM PB. If no ON option is entered, default is to the line printer.
LIST	Causes the printing of a list of all printer backup files of the form "PBnnnnn". The following information is printed for each file: <ol style="list-style-type: none"> 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:
 - a. "Forms" or "No forms"
 - b. "Labelled" or "Unlabelled"
 - c. "Translation" or "No translation"
 13. "Last record" (followed by the number of the last record printed).

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 station name assigned to the printer is DCPRINTxx, where the last two characters are arbitrary(numbers only).
4. 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 be declared in the NDL as AP300XAx where the last two characters are arbitrary(numbers only).

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 back 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 (5) 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 (3) 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 (10) 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 (1) 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

Thirty (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 MYDISK/LIST

Produces a listing which identifies all "PBnnnnn" files residing on the disk "MYDISK". Lists the basic characteristics of each backup file.

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 REQUIRED FOR <disk-name>/<file-name>  
CREATED AS <disk-name>/<file-name> <PRINT FILE NAME> BY <program-name>
```

The user must perform an "AX" to inform the utility whether to continue execution. The user enters:

```
AX <MIX NUMBER> Y  
or  
AX <MIX NUMBER> N
```

If the user enters "AX <MIX> Y", printing is performed. If the user enters "AX <MIX> N", that particular PB file is not printed. Entering anything else in the "AX" message causes the PB utility to send an error message to the user:

```
AX INVALID REQUIRES Y OR N
```

The user then resubmits the correct response.

After printing a file requiring special forms, the utility displays the message:

```
REMOVE FORMS
```

The user enters another "AX" to inform the system to continue:

AX <MIX NUMBER> OK

Entering anything else in the "AX" message causes the PB utility to respond:

AX INVALID REQUIRES OK

The user then resubmits the correct response.

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

Error	Possible Causes	Suggested Action
AX INVALID REQUIRES Y OR N	A "Y" or "N" was not entered in response to AX SCL.	Resubmit correct response.
AX INVALID REQUIRES OK	An "OK" was not entered in response to AX SCL.	Resubmit correct response.
<file-name> I/O ERROR	While reading or writing the named file, an error condition was detected.	Request technical assistance.
INVALID FILE NAME	This message appears if a "Star File" has been used for the initiating message. One of the following conditions may exist: 1. The packid portion of a filename is greater than 7 characters in length. 2. The fileid portion of a filename is greater than 12 characters or is empty. 3. The filename contains invalid characters.	Check "Star File" input and re-initiate PB.
INVALID PACKID	1. The packid specification is empty (that is, blank characters immediately preceding a slash). 2. The packid contains invalid characters.	Check input and re-enter.
FIELD TOO LARGE	1. The number of digits in a number following the page option is greater than 7. 2. The number of digits in a number in a record pair following the record option is greater than 7.	Check input and re-enter.

Error	(continued) Possible Causes	Suggested Action
NUMERIC FIELD NOT NUMERIC	While syntaxing a numeric field, a non-numeric character which is not a blank or a binary zero has been detected.	Check input and re-enter.
INVALID START	An error condition is detected when executing a conditional start.	Check record parameters in initiating message; re-enter.
EOF <file-name> ENCOUNTERED BEFORE PAGE(S) SKIPPED	When attempting to position to the page option. An End Of File condition is detected before the specified page has been reached.	Determine actual filesize and re-enter input.
<file-name> NOT SUCCESSFULLY OPENED	The conditional open on the printer backup file failed.	Re-enter input when PB is no longer in use.
<file-name> NOT AVAILABLE – IN USE	The fetch value returned on a file open indicates the open was not successful and the file was in use.	Re-enter input when PB is no longer in use.
<file-name> FILE TYPE NOT PB	The file type of the printer backup file opened was not “@A0@”.	None. The file is not a printer backup file.
READ ERROR	While attempting to read the header record, an error condition was detected.	Request technical assistance.
<file-name> HEADER RECORD NOT ASCII	The first 36 characters of the header record were not ASCII.	None. The file is not a printer backup file.
<file-name> INVALID BLOCK SIZE	The block size of the printer backup file is greater than 132 or equal to zero.	None. The file is not a printer backup file.
<file-name> INVALID RECORD SIZE	The record size of the printer backup file is either zero or greater than 132.	None. File corrupt.
<file-name> HEADER RECORD PB NUMBER GREATER THAN 65535 COPIES PARAMETER EQUALS ZERO	The number of the PB file as specified in the header record is greater than 65535. The number of copies specified in the initiating message is zero.	None. File corrupt. Check input and re-enter.
DUPLICATE OPTION	The option has been specified more than once in the initiating message.	Check input and re-enter.
PAGE PARAMETER EQUALS ZERO	The page number specified in the initiating message is zero.	Check input and re-enter.

Error	(continued) Possible Causes	Suggested Action
TOO MANY START LENGTH PAIRS HAVE BEEN SPECIFIED	More than 100 record pairs have been specified in the record option.	Check input and re-enter.
RECORD PARAMETER EQUALS ZERO	One of the numbers in the record pair is zero.	Check input and re-enter.
PB NUMBER GREATER THAN 65535	The PB number specified in the initiating message is greater than 65535.	Check input and re-enter.
INVALID EQUALS PARAMETER	The option following "=" is not COPIES or ON.	Check input and re-enter.
INVALID AUTO PARAMETER	The option following AUTO is not ON.	Check input and re-enter.
AUTO OPTION REQUIRES CLOCK ON SYSTEM	The AUTO option is not valid for systems which do not have a real-time clock.	Re-enter input without AUTO option.
INVALID ON PARAMETER	The parameter following ON is not LP, SP, AP or <station-id>.	Check input and re-enter.
INVALID PB RECORD FORMAT	The current record in the printer backup file does not adhere to the required format of printer backup media.	None. File corrupt.
INVALID OPTION	The option requested is not supported by the PB utility.	Check input and syntax diagrams and re-enter.
INITIATING MESSAGE TOO LONG	The length of the initiating message exceeds 401 characters.	Check input and re-enter.
INVALID PB PARAMETER	A parameter which is not valid has been found for a valid option.	Check input and re-enter.
INVALID LIST PARAMETER	The LIST option is followed by an invalid parameter.	Check input and re-enter.
MCS/DATA COMM SUBSYSTEM NOT AVAILABLE	A station-id was requested for a datacomm printer without GEMCOS running.	Check that the datacomm network is up and running and re-enter.
INVALID STATION NAME	The station name of the datacomm printer is invalid.	Check that the station name is of the format stated in DATACOMM PB.
PRINTER DEVICE SPECIFIED IS NOT AVAILABLE	Self-explanatory.	
AP JOURNAL PRINTER NEEDS OPERATOR ATTENTION	AP300 Journal printer status indicates operator attention required.	Check printer and clear fault.

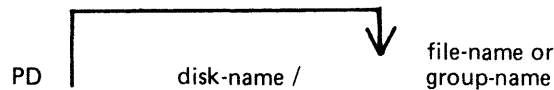
Error	(continued) Possible Causes	Suggested Action
DATACOMM ERROR – PB TERMINATING	The datacomm link with the printer is in error.	
ERROR CONDITION REPORTED BY <station- name>	The AP300 Journal printer status indicates an error has occurred.	Check printer.
PRINTER DEVICE <station- name> IS NOT AVAILABLE	The printer specified is unavailable.	Ready printer and re-enter.
<file-name> EOF ENCOUNTERED BEFORE number RECORDS SKIPPED	The end of the printer backup file has been reached before the specified number of records has been skipped.	Check file size and re-enter.

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 diagram:



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.
NO FILES FOUND IN THE FAMILY "group-name"	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 OVERFLOW DISK "disk-name"	Remainder of specified file resides on another disk. PD cannot complete until appropriate disk is supplied.	Supply appropriate disk.
<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.

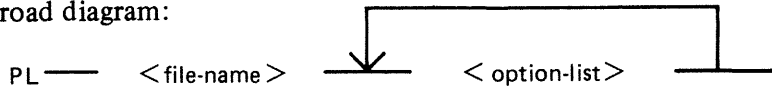
PL (Print Log Files)

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.

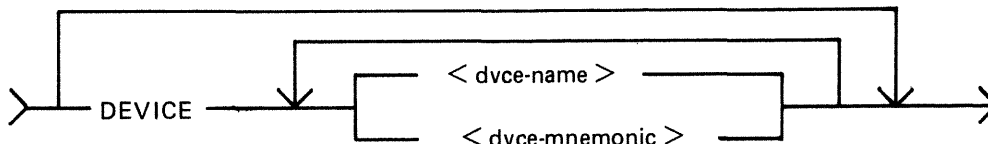
Format

Railroad diagram:



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
 Format for DEVICE option:

Railroad diagram:



where:

<dvce-id> is of the form DFA CTB etc.
 <dvce-mnemonic> is of the form DF CT etc.

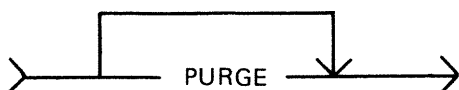
NOTE

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.

PURGE

Format for PURGE option:

Railroad diagram:



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 with 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
NO SPECIFICATION GIVEN	The initiating message contained no parameters.	Re-enter correctly.
NO FILE NAME FOUND IN PARAMETERS	The first parameter found in the initiating message was not the log file name.	Check input and re-enter correctly.
ILLEGAL OPTION <parameter>	The parameter displayed is not a valid one.	Check input and re-enter correctly.
ILLEGAL VALUE <integer>	The integer entry after either ENTRY or MIX is not within the allowable range of values.	Check input and re-enter correctly.
NO TIME SPECIFICATION AFTER <time option>	Input after either "TO" or "FROM" is incorrect or inappropriate.	Check input and re-enter correctly.
NO <param> NUMBER GIVEN	A number was expected after "MIX" but was not found.	Check input and re-enter correctly.
<option> OPTION ALREADY SET	One of the parameters set contradicts another.	Check input and re-enter correctly.
ILLEGAL <date/time> SPECIFICATION <time-spec>	The <time-spec> is not of the correct format. <time> comes before <date> if both are specified.	Check input and re-enter correctly.

Message	(continued) Possible Causes	Suggested Action
FILE <file-name> NOT FOUND or, FILE <file-name> IN USE	The desired file is not on the specified disk, or is not available.	Make file available and re-enter.
<file-name> IS NOT A LOG FILE	The file being processed is not of log-file type or format.	
ILLEGAL PARAMETER LIST <param>	A parameter error has been found. The parameter in error is displayed.	Check input and re-enter correctly.
ILLEGAL FILE NAME <file-name>	The file-name does not conform to CMS format.	Check input and re-enter correctly.
INVALID CHARACTER IN IDENTIFIER <identifier>		Check input and re-enter correctly.
DISK <pack-id> NOT AVAILABLE	The given <pack-id> is not on-line.	Put the desired disk on-line and re-enter correctly.
**** UTILITY LIMIT REACHED ****	Invalid information contained in the log file.	None.
UNRECOGNIZED DEVICE IN ERROR ENTRY	No device can be found in the Hardware Configuration Table which corresponds to the device type in the error descriptor and hence, no error counts can be updated from this information.	None.
NO ENTRIES FOUND WITHIN PARAMETERS	End of File has been reached without any entries being found and printed.	None.
DISK <disk-id> LOCKED	The file requested is on a disk for which a utility is using SYSMEM LOCK.	Re-enter when the utility has finished.
SCREEN SIZE TOO SMALL TO RUN UTILITY	When using the DISPLAY option, the utility found that the SPO screen size was less than 80 characters wide and 10 characters deep.	Re-input without the DISPLAY option.
ENTER END TO FINISH OR NEXT TO CONTINUE	After displaying a screenful, the utility will request whether or not more information is required.	Enter NEXT or END.
INVALID REQUEST <param>	A response other than the expected END or NEXT was entered. The utility will again request whether or not more information is required.	Check input and re-enter correctly.

Message	(continued) Possible Causes	Suggested Action
OUTPUT FILE <file-name> TOO SMALL	When using the BACKUP option, the utility reached the end of the output file.	None – the file will be closed and the utility will go to EOJ.
<decimal no> RECORDS WRITTEN TO BACKUP FILE (file-name)	Writing to the backup file has been completed and the file has been closed.	None.
WARNING – ENTRY OPTION NOT ALLOWED FOR FILE <file-name>	An attempt has been made to use the ENTRY option on a file containing only maintenance entries. Processing will continue but the ENTRY option will be ignored.	None.
DEVICE <device spec> HAS BEEN SELECTED ALREADY	A duplicate device selection request has been found. No further processing will be done.	Re-enter.
UNRECOGNIZED DEVICE IDENTIFICATION <device entry>	The parameters in DEVICE option contain an invalid <device-entry>. No further processing will be done.	Check input and re-enter.
NO DEVICE(S) FOLLOWING DEVICE SPECIFICATION	No <device-entry> was discovered following the keyword DEVICE. No further processing will be done.	Check input and re-enter.
DEVICE <device-entry> NOT FOUND IN HARDWARE CONFIGURATION AREA	The <device-entry> specified in the DEVICE option does not have an entry in the first hardware configuration area encountered.	Check input and re-enter.
WARNING – FILE <file- name> MAY NOT BE PURGED	The utility has been requested to purge a file which appears to be a primary log file. PL will process the file but will ignore the request to PURGE it.	None.
<file-name> REMOVED	The specified field file has been successfully removed.	None.

As well as these messages, a check on parity errors is made during the processing of the file. If a read/write parity error is incurred, then the message:

I/O ERROR DETECTED IN READ FROM <file-name>

or

I/O ERROR DETECTED IN WRITE TO FILE <file-name>

is displayed, followed by the record number at which the error occurred. The program will either attempt to continue processing or will go to End of Job, depending on where the error occurred. If it does continue, then “CONTINUING PROCESS” will be displayed.

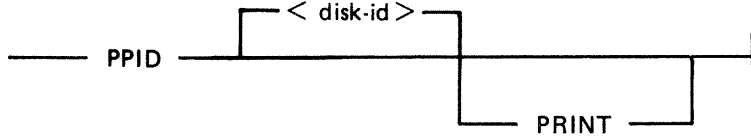
However, if the ENTRY option has been used, and the error occurs while processing this command, then the results might not be those requested.

On reaching the end of the processing, the utility goes to End of Job and closes all files, leaving the input file in exactly the same state as it was at the start.

PPID (PSEUDO PACK IDENTIFIER DISPLAY)

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:

Railroad diagram:



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.UNIT xx/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 physical 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.

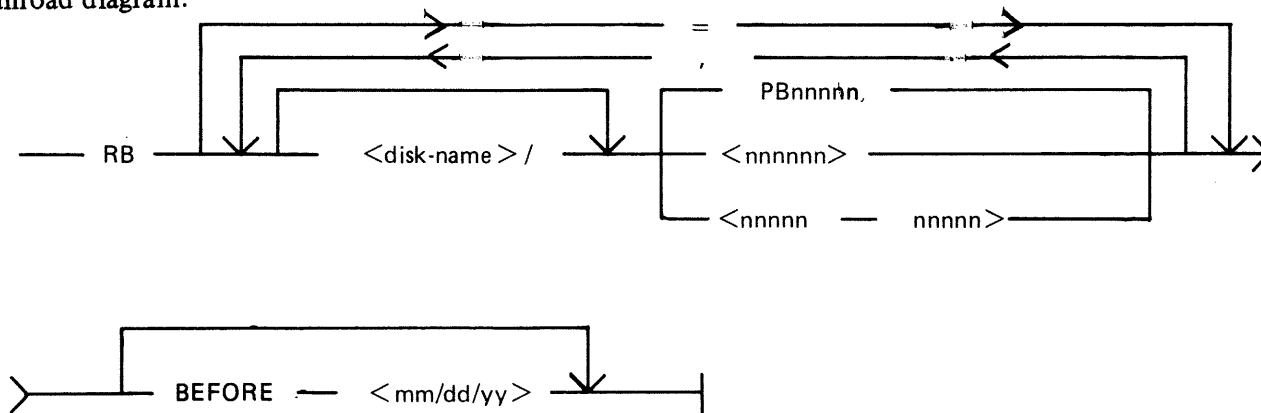
Output Messages

Message	Possible Causes	Suggested Action
NO PPIT ON SYSTEM	No pseudo pack identifier table was found on the system disk.	None.
READ ERROR ON PPIT OF “DISKID”	An error was encountered while reading the PPIT.	None. Irrecoverable error.

RB (Remove Printer Backup Files)

This utility enables the user to remove printer backup files from one or more disks. It can remove backup files singly, by family, or by age.

Railroad diagram:



Where:

- <disk-name> This is the name of the disk containing the required printer backup file(s); the default is the printer backup disk designated in CONFIGURER.
- PB<nnnnn> This is the printer backup file name. Where <nnnnn> is a five digit number in the range 00001 to 65535.
- <nnnnn> Specifies the backup file PB<nnnnn>. Leading zeros may be omitted.
- <mmmmm - nnnnn> Specifies all backup files from PB<mmmmm> to PB<nnnnn>. Where <mmmmm> and <nnnnn> are numbers in the range 1 to 65535. Leading zeros may be omitted.
- BEFORE <mm/dd/yy> This option causes all backup files, with a file name of the format PB<nnnnn>, created before the specified date to be removed. Where mm and dd (month and day respectively) may be either 1 or 2 digits, while yy must be 2 digits.

Examples

To remove the file PB00038 from the system disk:

```
RB 38
```

To remove the file PB00017, and all backup files in the family PR=, from the disk USDSK:

```
RB FROM USDSK 17, PR=
```

To remove PB00032 and PB00034 from the system disk, all the backup files named PBxxxxx from the disk TASK, and all backup files in the range PB00011 to PB00019 from the disk ARDSK, each one of these files subject to the date given:

```
RB 32,34 ; FROM TASK = ; FROM ARDSK 11-19 BEFORE 3/3/81
```

Messages

If syntax errors occur, the utility displays messages that are self-explanatory, and goes to End of Job.

If a disk cannot be opened, or no files are found in a family, RB continues execution after displaying the appropriate messages.

The user is also informed when a file is removed, or when it is not removed (because it is still in use, or because it was created on or after the "BEFORE" date).

After completion, RB issues the message:

```
** RB COMPLETE **
```

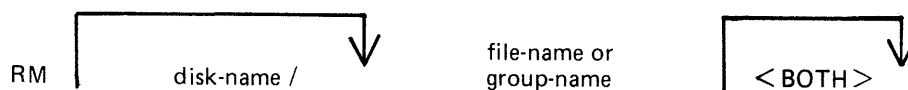
RM (Remove Files from Disk)

(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 diagram:



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/RM filename OK (mix-number is the mix number of RM)
```

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/RM NDL.INTERP OK

Output Messages

Message	Possible Causes	Suggested Action
“filename” REMOVED	File was removed.	None.
“filename” NOT REMOVED NOT FOUND	Specified file was not removed.	Check input (re-input if necessary); check for correct disk.
“filename” NOT REMOVED – IN USE	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, etc).	If a file is to be removed, type “AX mix-number/RM filename OK”. If file is not to be removed, type AX mix/RM NO.
INDEXED PAIR “filename” “filename” REMOVED	Keyfile and associated data file were removed.	None.
INDEXED PAIR “filename” “filename” NOT REMOVED	Keyfile and associated data file were not removed. This message is followed by the reason.	Check input (re-input if necessary) or check for correct disk.
INDEXED PAIR “filename” “filename” NOT REMOVED – IN USE	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 used, then re-input.
END RM	Normal EOJ message.	None.

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 are 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:

“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.	
AX <mix-no> <test list>	INVALID INPUT CHARACTER <PROMPT 1> SCR INITIATING MESSAGE ERROR DUPLICATE SCR SUBTEST SCR ABORTED DUPLICATE TEST NO. <PROMPT 2> <PROMPT 3> <PROMPT 4> <PROMPT 5>	Test list input did not consist of valid test numbers or "ALL". Self explanatory. A digit has been repeated in the test list entered. Test 1 (DCR) or "ALL" has been selected. Test 2 (CPU.IO) or "ALL" has been selected. Test 3 (MT.IO) or "ALL" has been selected. Test 4 (OTHER.IO) or "ALL" has been selected.	Re-input test list, or terminate the utility. Re-start SCR utility, taking care to only enter one of each test number. Re-enter test list, or terminate the utility. Refer to possible replies to Prompt 2. Refer to possible replies to Prompt 3. Refer to possible replies to Prompt 4. 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.	

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.	
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>.	
	FETCH VALUE = <hex-no.> READ ERROR ON <disk-name> LABEL DCR ABORTED	MCP input error on reading the disk label.	
	ILLEGAL PARAMETER LIST 0000000 <PROMPT 2.1>	The default System Disk identifier has been used.	To use the System Disk, the disk name of the actual disk where the system resides.
	DISK <disk-name> IS A PSEUDO-DISK <PROMPT 2.1> <PROMPT 2.2>	DCR does not use PSEUDO-DISK disk-names. <disk-name> has been opened successfully and disk-label has been read.	Re-enter a non pseudo-disk disk-name. Proceed to the instructions for PROMPT 2.2.

PROMPT 2.2:

“SECOND DISK TO BE TESTED? <Y or N>”

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.

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.

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.	
AX <mix-r.o> <disk-name>	FETCH VALUE = <hex-no.> DISK <disk-name> NOT AVAILABLE DCR ABORTED	The MCP has been unable to open <disk-name>.	
	FETCH VALUE = <hex-no.> READ ERROR ON <disk-name> LABEL DCR ABORTED	MCP input error on reading the disk label.	
	ILLEGAL PARAMETER LIST 0000000 <PROMPT 2.1>	The default System Disk identifier has been used.	To use the System Disk, the disk name of the actual disk where the system resides.
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.	
	<PROMPT 2.3>	Disk capacities have been compared and are equal, disk-name has been successfully	Proceed to the operating instructions for PROMPT 2.3.

Possible Reply	Possible Resulting Output	(continued)	
		Possible Cause	Suggested Action
		opened and the disk-label has been successfully read.	

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 <PROMPT 2.4>	Self-explanatory.	Re-start SCR utility, taking care to only enter one of each test number.
		Test 1 or “ALL” tests have been selected.	Proceed to input for PROMPT 2.4.

Possible Reply	(continued)		Suggested Action
	Possible Resulting Output	Possible Cause	
	<PROMPT 2.5>	Test 4 or "ALL" tests have been selected.	Proceed to input for PROMPT 2.5.
	<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> <neither 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.	
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.	

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.		

(continued)			
Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
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.
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	None

Message	(continued) Possible Cause	Suggested Action
ERROR ON <disk-name> MULTI-SECTOR READ	(for both disks, if two are used) will be via the single-sector buffers. 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 (FDNL) 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 being 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.	
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-	DCR reads back index files in the order in which they were	

(continued)		
Message	Possible Causes	Suggested Action
value >	written. If the MCP reports 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 substest requested.	
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.	
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.	
READING RANDOM SECTORS ABORTED	If more than 300 errors are reported from a disk, DCR displays this message and abandons the test.	
<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	

Message	(continued) Possible Cause	Suggested Action
READING RANDOM SECTORS COMPLETED	<p>have been read from each disk being tested, this message – a summary of the error error count n – is output.</p> <p>When the total number of requested sectors have been read from each disk, this message is displayed.</p>	
<n> BAD SECTORS READ ON <disk-name>	Total number of bad sectors read.	

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.	
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.	
	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	Zero devices (magnetic tape reels and	Re-enter the parameters for MT and CS devices,

(continued)

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
	DEVICES = 0 <PROMPT 4>	cassettes) were explicitly requested.	taking care that the 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> <PROMPT 5.1>	The input string contains a non-decimal character.	Re-input the required no. of iterations correctly.
		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.”

Execution Details

“SCR” is followed by an initiating message. This message can take two forms:

Possible Reply	Possible Resulting Output	Possible Cause	Suggested Action
AX <mix-no> Y	CARD PERIPHERALS NOT IMPLEMENTED	No card tests will be performed.	
AX <mix-no> N		No card tests will be performed.	
AX <mix-no.> <not "Y" or "N">	REPLY NOT "Y" OR "N" <PROMPT 5.2>	Self explanatory	Re-enter either "Y" or "N".

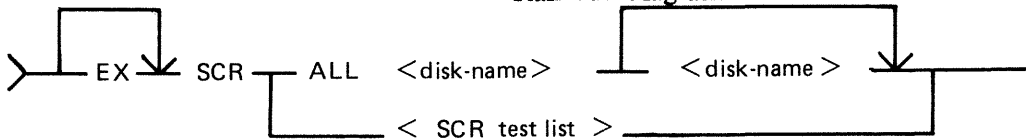
1. SCR parameters (Example 2), or
2. names of star-files containing SCR parameters (Example 3).

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.

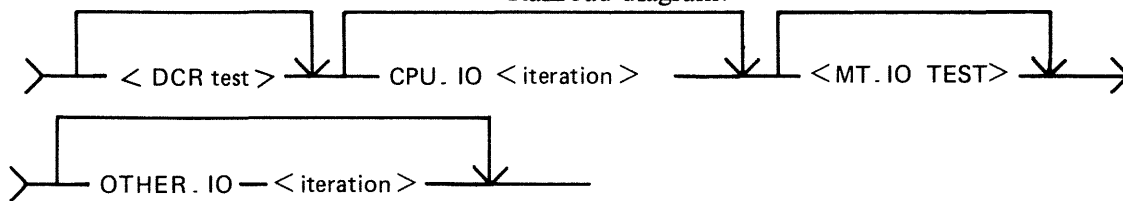
Railroad diagram:



iteration = the number of iterations of the subsection.

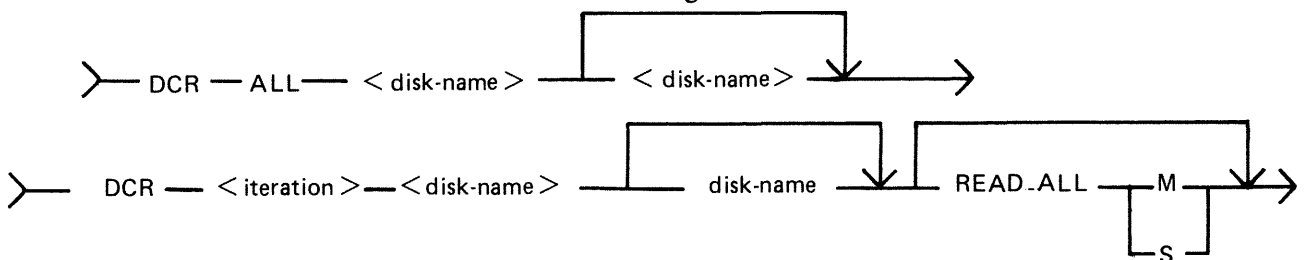
Where SCR test list is:

Railroad diagram:



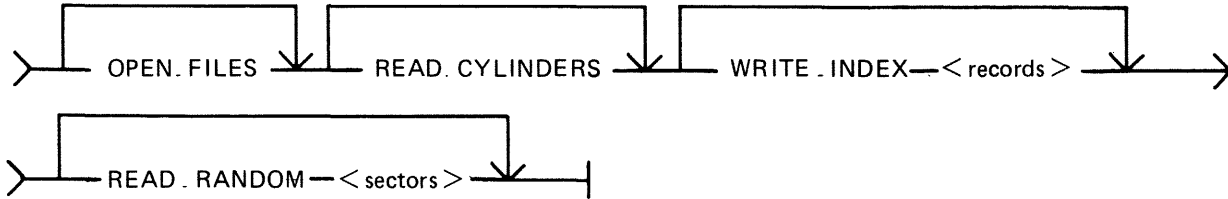
Where DCR test is:

Railroad diagram:



or

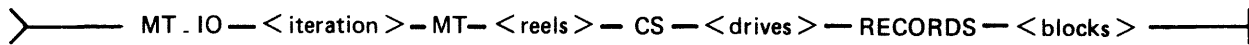
Railroad diagram:



records = The number of (data) records in the index file.
sectors = The number of sectors to be read randomly.

Where MT.IO test is:

Railroad diagram:



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.COBOLE 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.COBOLE". Messages 12 and 13 do not apply to SCR.MPL or SCR.COBOLE, as the ZIP is without PAUSE.

As SCR.MPL and SCR.COBOLE 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.COBOLE, 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.

SCR.COBOL

Using the initial values:

A = 10
B = 1
C = .1

D = .01
E = .2

SCR.COBOLE 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.COBOLE 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.COBOLE 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 4-7.

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.

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7.....5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
-----
X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X X

X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X

X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X

X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X

X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X.....X X X X X X X X X X X X X X X X

      " .....
      " .....
      " .....
      " .....
      "
      V

                X X X X X X X X X
                X X X X X X X X X
                  X X X X X X X X
                   X X X X X X X X

                      X X X X X X
                      X X X X X X

                          X X X X
                          X X X X

                              X X X
                              X X X

                                  X X
                                  X X

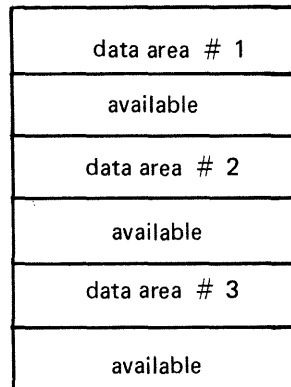
                                      X
                                      X

```

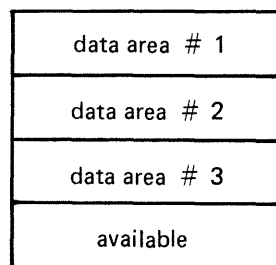
Figure 4-7. OTHER.IO Sample Printer Output

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 USER DISK" failures and extra time 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" 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 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

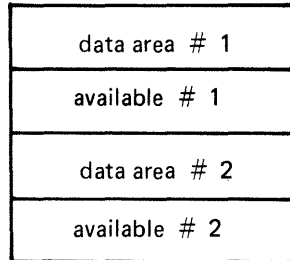
Only data areas within a default section of the disk are moved to lower addresses within the section.

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:

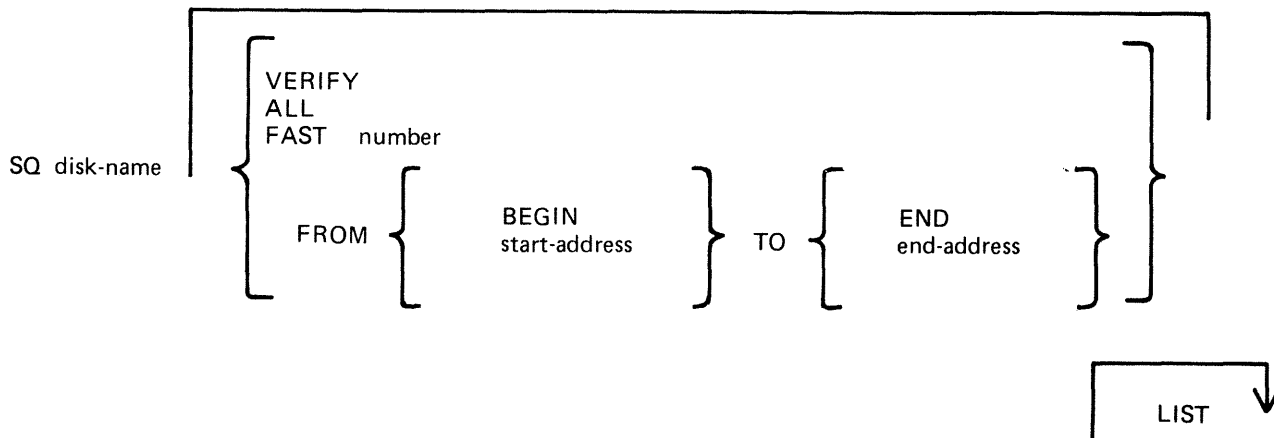


where data area #1 is 100 units, data area #2 is 200 units and both available areas are 1 unit each. If available areas are merged the available area gained would be 2 units. However, to acquire these 2 units, the 200 units of data area #2 would have to be moved. Therefore, an "economic squash" would 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:

Format diagram:



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.

If a printer is available, SQ proceeds normally, sending list information to that printer. If no printer is available, SQ prompts the operator with the following message:

```
SQ NEEDS A PRINTER - DO YOU WANT TO CONTINUE WITH NO PRINTOUT <Y OR N>
```

The operator has two options:

1. If a printout is not required, "Y" should be AXed to the utility. SQ then continues without producing a "LIST".
2. If a printout is required, the operator must make available an on-line ready printer, then "N" should be AXed to the utility. SQ then looks for a printer for output.

The utility will only accept "N" or "Y" as input. If some other input is AXed to the utility, the prompt will be repeated.

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 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. In addition, SQ can only be run in a suitable mix, as defined by the MCP to safeguard the integrity of the disk. No user program can be run with SQ. During execution of SQ the MCP will reject any attempt to execute any utility or user program.

Output Messages

Message	Possible Causes	Suggested Action
LARGEST AVAILABLE AREA IS number SECTORS. TOTAL AVAILABLE IS number SECTORS IN	Given on successful completion of SQ.	None.

Message number AREA(S). ***SQ COMPLETED***	(continued) Possible Causes	Suggested Action
NON-FILE DIRECTORY FULL – “PARTIAL” SQUASH REQUIRED	Display during verification phase if there are no free entries when attempting to add entries to the available table if missing areas are detected.	Run SQ with VERIFY LIST options. Examine the disk map to discover a section of the disk that can be squashed to create free entries in the available table. Example: the pattern File Area Available File Area Available if squashed will create one free entry in the available table. In general the # of free entries to be created equals the # of missing areas that are not contiguous with any available area.
SQ INVALID – NO INITIATING MESSAGE	Self-explanatory.	Re-input.
SQ INVALID – INVALID PACK-ID	Pack-id specified is longer than 7 characters.	Re-input.
SQ INVALID – INVALID SYNTAX	Self-explanatory.	Re-input.
SQ INVALID – INTEGER MUST NOT BE GREATER THAN 65535.	The maximum # of sectors requested for the FAST option is 65535.	Re-input.
SQ INVALID – INVALID ADDRESS	No address or an invalid address (for example, 004G15) was specified for the FROM or TO address with partial squash.	Re-input.
*** BAD SECTOR AND filename OVERLAP – NO WAY TO SEPARATE THEM. SAVE AND/OR PURGE AND AT LEAST RETURN SQ VERIFY	Area marked in available table as “bad” overlaps with area allocated to file.	SQ cannot resolve this. Integrity of file is suspect. Remove the file after copying it to another disk for examination if necessary and run SQ VERIFY.
***filename AND filename OVERLAP – THERE IS NO WAY TO SEPARATE THEM. SAVE ONE OR BOTH, PURGE AND AT LEAST RERUN SQ VERIFY.	Area allocated to file overlaps with area allocated to another file.	The two areas cannot be separated. Copy each file individually to another disk for later examination if required, remove them both, re-run SQ VERIFY.

Message	(continued) Possible Causes	Suggested Action
SQ ABORTED – REQUESTED AREA ALREADY EXISTS	Request was made with FAST option for an available area which already exists.	None.
*** SQ ABORTED – INVALID DISK ALLOC. UNIT = 0	Disk label is probably corrupted.	Disk must be assumed useless and should be re-initialized.
*** DIRECTORY FID NEQ HEADER FID FOR FILE filename. CORRECT USING CH AND RESTART SQ	Name of file in file directory name list does not match disk file header. The filename displayed is that in the same list.	Enter “CH <FILE-ID> TO <FILE-ID> to correct the anomaly (this re-writes the disk file header).
SQ INVALID – SPECIFIED DISK NOT AVAILABLE	Self-explanatory.	Make disk present and re-run SQ; check input.
*** SQ ABORTED NOTHING TO SQUASH IN THAT AREA	A partial squash was requested and SQ found nothing to do in that section.	None.
SQ ABORTED – NO WAY TO GET REQUIRED AREA	Area size specified in the FAST option cannot be obtained either because it is larger than the total available space or because certain areas cannot be moved to release available space. For example, Area # 1 100 units Available # 1 1000 units Area # 2 100 units Available # 1 cannot be used if areas # 1 and # 2 are in use or system log files because areas # 1 and # 2 cannot then be moved.	Attempt to remove unwanted user files: re-input SQ.
*** MEMORY INCONSISTENCY OR SOME OTHER IRRECOVERABLE PROBLEMS – RERUN SQ	Internal work-tables in memory (used by SQ) are corrupted.	Re-run SQ VERIFY. If problem persists, request technical assistance.
*** ADDRESS MISMATCH – SAVE AND RE- INITIATE THE DISK	Some addresses in disk directory are probably corrupted.	Try to dump or copy files from the disk. Disk must be re- initialized before re-use.
*** IRRECOVERABLE ERROR ON DISK – SAVE AND/OR RE-INITIALIZE	Disk is corrupt.	Try to dump or copy files from the disk. Disk must be re- initialized before re-use.
*** TOO MANY FILES OPEN AND/OR BAD AREAS – NO WAY TO SQUASH THE DISK	Self-explanatory.	Save required files from disk, then re-initialize the disk.

Message	(continued) Possible Causes	Suggested Action
*** DISK INTEGRITY SUSPECT – USAGE MAP OF THE DISK WILL BE PRINTED (IF LIST IS NOT SET)	SQ has detected and resolved an overlap situation; but disk remains suspect.	Scrutinize disk map, request technical assistance.
*** AREAS STILL MISSING – RERUN SQ VERIFY	A FAST squash has detected that areas of disk are not accounted for.	SQ VERIFY will return missing areas to the available table.
*** LAST SQ EXECUTION WAS ABNORMALLY TERMINATED WHEN MOVING FILE – filename. INTEGRITY OF FILE SUSPECT. EXECUTION CONTINUES	System crash occurred while file areas were being moved.	Remove suspect file.
*** SQ ABORTED # I/O ERROR AT DISK ADDRESS @NNNNNN@	Hard error on disk persists after 10 retries.	If address is outside directory area, remove offending area with XD utility.
*** I/O ERROR ON filename FILE SKIPPED – EXECUTION CONTINUES	Hard disk error encountered. File is not moved.	None.
*** WRITE ERROR IN NEW LOCATION WHEN MOVING FILE AREA. DISK ADDRESS – @NNNNNN@. THIS AREA SKIPPED – EXECUTION CONTINUING	Hard error encountered when moving file area to available area.	Available area is left as available and should be XD'ed after squash EOJ.

NOTE

Error messages marked with “****” indicate that a hardware or system software error has occurred or that the disk itself is suspect. If these persist, request technical assistance.

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.

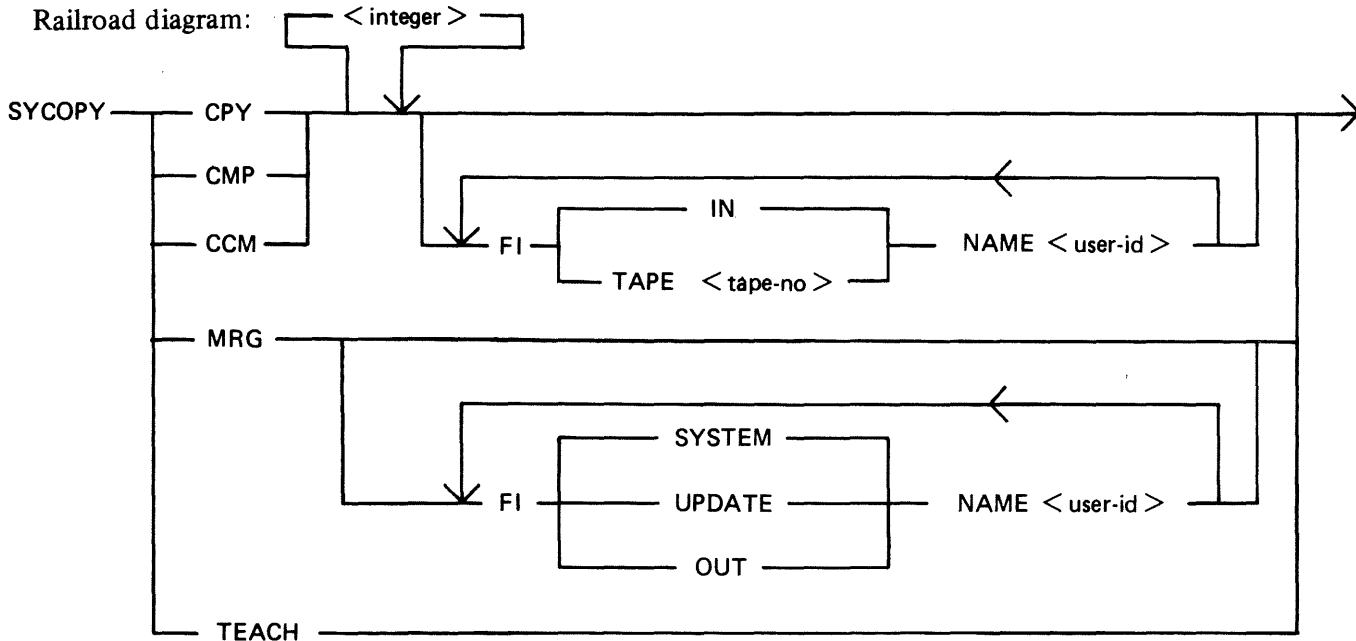
“SQ VERIFY” is a means of checking the integrity of any disk and if run on a regular basis may help pinpoint sooner rather than later any degradation in hardware performance or system software

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.

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 $\langle \text{integer} \rangle = 1-7$ (defaults to 1)

The meanings of these options are:

file-equate is defined as one or more of the following entries:

FI default-name NAME actual-name

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 $\langle \text{integer} \rangle$ 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 $\langle \text{integer} \rangle$ 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 CMP 1 FI 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 names “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 number TAPE(S) DELETED	One or more, but not all, of the output tapes has been deleted.	None.
number TAPE(S) COPIED AND COMPARED number TAPE(S) DELETED	One or more but not all of the output tapes has been deleted.	None.
INVALID SYNTAX – NO INITIATING MESSAGE		
INVALID SYNTAX – INIT MSG NOT COMPLETE	Specified invalid specification in initiating message.	Check input and re-enter if necessary.
INVALID SYNTAX – INVALID PARAMETER LIST <string>		
INVALID SYNTAX – TOKEN TOO LONG <string>		
COMPARISON ERROR ON <tape.id>/<file.id> – RECORD NB <number>		
COMPARISON ERROR ON <tape.id>/<file.id> – NOT END OF FILE	Compare function failed, possibly due to a hardware malfunction.	Re-run SYCOPY with good tapes and drive.
COMPARISON ERROR ON <tape.id>/<file.id> – END OF FILE		

If all output tapes are deleted, the following message will be displayed before the utility is aborted:
TAPES NOT IDENTICAL – SYCOPY ABORTED

Other Messages:

NOT LIB TAPE <tape.id> – SYCOPY ABORTED
RECORD SIZE DIFFERENT ON <tape.id> (for CPY, CMP, CCM)
RECORD SIZE DIFFERENT – SYCOPY ABORTED (for MRG)
MK NUMBERS DIFFERENT – SYCOPY ABORTED (for MRG)

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(s)

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
library-tape-name NOT A RECOGNIZED DUMP TAPE	This tape was not created by either DUMP or UNLOAD functions of LD utility. It is ignored by the TAPELR 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(s)

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 is 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.

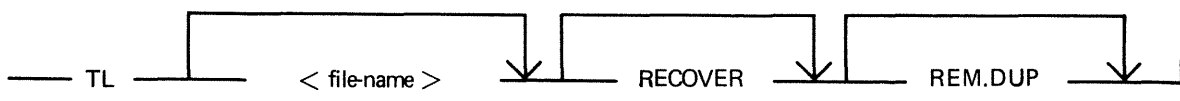
TL (Transfer Logs)

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.

Railroad diagram:



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 date, nn is a cyclic numeric progression in the range 00 to 99).

TL will also transfer system message to the secondary log file SYS-HLD jjjnn, 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.

Output Messages

Message	Description
<file-name> NOT FOUND	The utility has been unable to find the star file containing the parameters. Ensure the star file is present, then try again.
<file-name> NOT AVAILABLE – IN USE	The requested file is being used by another program.
NO LOG FILES FOUND FOR CONSOLIDATION	The utility was unable to find any files of the form “SYS-LOG-nn” where nn can take the values 01, 02, 03 and 04.
ILLEGAL USE OF RECOVER PARAMETER followed by ACTIVE FILE NOT CONSOLIDATED	The user has executed TL with the RECOVER option while logging is initiated on the system. The “ACTIVE” log file is being used by the MCP.
NO READY-TO-TRANSFER FILES FOUND	The utility was unable to find a “READY-TO-TRANSFER” log file. No consolidation will therefore occur.
NO ACTIVE LOG FILE FOUND	The option RECOVER was specified when the utility was zipped or initiated. There was no “ACTIVE” log file found. Complete consolidation of the log files has not taken place.
DISK <disk-name> LOCKED or SYSTEM DISK LOCKED or <file-name> NOT AVAILABLE – DISK LOCKED	The disk on which the file that TL is using is in use by another program.
INVALID CHARACTER IN IDENTIFIER <identifier>	The identifier specified contains an illegal character.
TRANSFER COMPLETED	The utility has transferred the “READY-TO-TRANSFER” log files.
DISK <disk-name> NOT AVAILABLE	This is displayed when the disk specified is not available. Generally, this will be the name of the backup disk specified in SYSCONFIG. In these circumstances, the transfer defaults to the SYSTEM disk.
SECONDARY LOG FILE <file-name> CREATED	A new backup file has been created.
INSUFFICIENT DISK SPACE TO CONSOLIDATE LOG FILES	The utility has detected a NO USER DISK state. It will close any files it has opened, display this message and go to EOJ.
<file-name> REMOVED	A duplicate file condition has arisen during consolidation, but the option “REM.DUP” was specified so the duplicate file has been removed.

Message	(continued)	Description
CANNOT OPEN FILE <file-name> – DIRECTORY FULL		The utility has attempted to create a new file, but the disk on which this file is to reside has reached its maximum file capacity.
INVALID IMPLEMENTATION LEVEL OF SYSCONFIG		While attempting to access information in SYSCONFIG, TL has discovered that this version is invalid for the logging system being used.
INVALID HARDWARE CONFIGURATION AREA IN FILE <file- name>		While TL was examining the hardware configuration area of the file specified, certain values were found to be invalid.
<option> OPTION ALREADY SET		An option has been specified more than once.
FILE <file-name> HAS BEEN ALTERED		The size of this file is less than the minimum size for a secondary log file. TL will create a new file with a progression value one greater than that of this file.

As well as these messages, checks are made for parity errors both on reading and writing. If any are detected, then a message is issued in the form:

either "I/O ERROR DETECTED IN READ OF FILE <file-name>"
or "I/O ERROR DETECTED ON WRITE TO FILE <file-name>"

for a read or write parity error.

Another message follows to indicate the record number at which the error occurred.

If the utility continues the processing, the message "CONTINUING PROCESS" is displayed, otherwise it goes to End of Job and leaves a partially consolidated file, depending on where the error occurred.

TRANSLATOR

CMS provides for a local language facility by releasing an English language directory file containing all SCL messages used by system software. This file is called SYSLANGUAGE. The TRANSLATOR utility is used to modify certain sections of the SCL message file allowing conversion of this file to any other language (provided that language has a horizontal, left to right presentation). The presence of this directory file on the system disk is required for a successful warmstart. The name of the file to be used is specified in the SYSCONFIG file which uses the default name of SYSLANGUAGE.

The language conversion is achieved by:

1. Permitting substitution of text for system words, for example, "READ", "OPEN", "READY", "EXECUTING" and so on.
2. Permitting choice of text for the message description.
3. Permitting restructuring of each message to allow semantically correct placement of text and parameters.

NOTE

When submitting evidence with a Field Communication Form which includes a log printout of any type, also include a copy of the SYSLANGUAGE used (if not the released version) on machine readable media.

Limitations

The SYSLANGUAGE file has an unique entry for each possible system message. Each entry consists of a combination of ASCII text and escape codes. Escape codes are requests to the MCP to execute intrinsic functions which fetch and format system parameters to be used in the display of the message. The major interface between the MCP and the directory file is the message number. The MCP is aware of which parameters are necessary in the display of a message and moves the required data to the appropriate field. Therefore, it is a limitation of the local language conversion process that no new escape code is introduced or an existing escape code removed from a message, since the appropriate parameter may be incorrectly set and cause a misleading message to be displayed.

Another limitation is that the total message length, when expanded may not exceed 300 characters.

One other limitation is that the total length of any request put to the utility may not exceed 120 characters. This therefore means that any text in a FIX command can be no longer than approximately 100 characters.

Operating Instructions

The TRANSLATOR utility enables the user to perform the required modification to the directory file (SYSLANGUAGE). This file is divided into six sections.

1. VERB list
2. WORD list
3. DEVICE list
4. MESSAGE DESCRIPTION list
5. UTILITY MESSAGE list
6. MCP MESSAGE list

The DEVICE list exists in the directory file but it is not accessible for text alteration as device mnemonics are an integral part of CMS.

The WORD section between £60 and £6F cannot be altered as these are utility names.

The VERB, WORD and MESSAGE DESCRIPTION lists contain words and phrases commonly used in MCP messages.

The UTILITY MESSAGE list makes up the actual message dictionary for the utility messages by formatting the parameters via escape codes. See UTILITY ESCAPE CODES.

The MCP MESSAGE list makes up the actual MCP message directory and references the sections VERB, WORD and MESSAGE DESCRIPTION lists of the file via the escape code values. See MCP ESCAPE CODES.

To help the user, a cross reference of the utility messages is included.

Initiation and Working Mode

The initiation of the TRANSLATOR utility and many of the functions used are similar to the operation of CMSCANDE.

If a datacomm system is present, the program may be executed from a TD 800 type terminal device. However, if the datacomm is not present, or access is denied by the MCS, the utility attempts to use console I/O. If this fails, Displays and Accepts are used. This same procedure is used if datacomm is being used, but the MCS is terminated during the execution.

If a datacomm 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 the TRANSLATOR utility are similar to those of CMSCANDE. The available functions are:

GET, FIND, FIX, NEXT, REMOVE, RECOVER, SAVE, LIST and BYE.

GET

Railroad diagram:

GET ——— <file-name> ———|

This command attempts to open the file named and use it as the Source Language file. If it fails to open the file, another GET of another file, or a BYE may be requested. When the file has been loaded, a new language type is requested (such as French, German and so on). Enter up to 24 non-blank characters.

RECOVER

Railroad diagram:

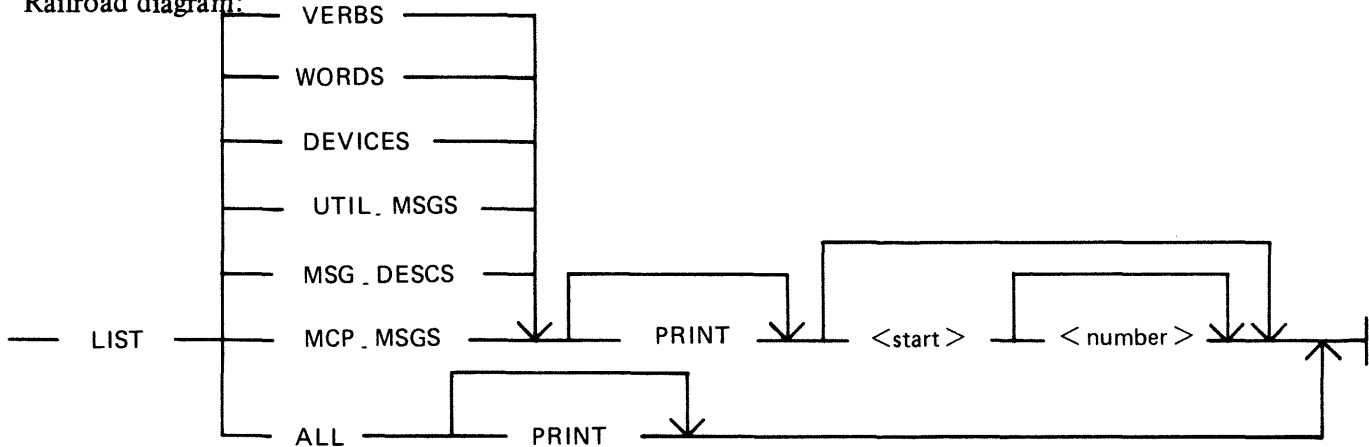
——— RECOVER ———|

The RECOVER command may only be executed when the utility has stated that work files are present and requests either a REMOVE or a RECOVER. After a successful RECOVER, the utility behaves as if a successful GET has been performed.

LIST

The function of the LIST command is to display or print one or all of the message files.

Railroad diagram:



If ALL is requested, the entire message file is displayed/printed.

By specifying PRINT, all output from this request is directed to a line printer; otherwise, the output is directed to the user.

The <start> parameter is not allowed with the ALL option.

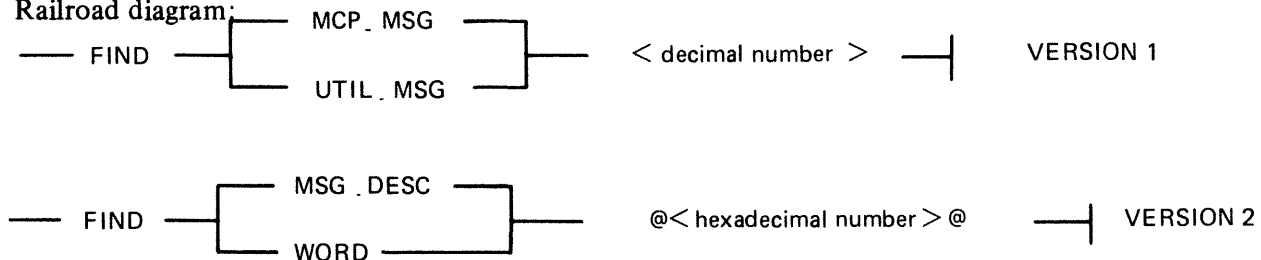
If a LIST has been requested and the user is using the datacomm system, it is possible to terminate this command while it is executing. This may be done by sending "BRK" to the utility. The utility responds with an appropriate message and terminates the LIST command.

While output is being directed to a screen device, a request to continue or to end is issued when either the screen is full, or the end of the current section has been reached and ALL has been selected.

FIND

The FIND command tries to locate the specified item within either the WORD, MESSAGE DESCRIPTION, UTILITY MESSAGE or MCP MESSAGE lists and displays it.

Railroad diagram:



The first version tries to locate the message given by the decimal number and displays it to the user.

The second version tries to locate the item given by the hexadecimal number.

In the case of the MCP.MSG option, any escape codes which can be, are expanded and enclosed between "++" before the message is displayed.

Example:

“FIND MCP.MSG 553” gives the display:

“553 @84@ FF11 FFEC++ PRINTER BACKUP FILE ++ FF36 OPENED”

the expansion of the escape code FFEC is “PRINTER BACKUP FILE” (see the message number @EC@ in the section, MESSAGE DESCRIPTION, in the SYSLANGUAGE file).

NEXT

Railroad diagram:



The NEXT command is used after a previous FIND and displays the next item in the section previously requested by the FIND. On reaching the end of the section, no further NEXTs are allowed until a further successful FIND has been requested.

FIX

The FIX command is used to modify an item in any one of the sections, except in the DEVICE section (as the device mnemonics are CMS defined values and cannot be altered) and the WORD section @60@ to @6F@ (as these are intrinsic names, are CMS defined and cannot be altered). The user specifies a section, an item and a text enclosed between delimiters (see Delimiters note); the utility then locates the item in the section and replaces its text by the one given by the user. There are three options for the FIX command depending on which section of the file is to be changed.

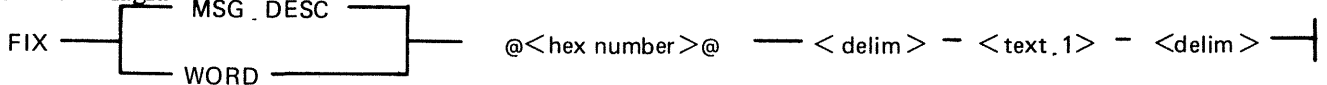
Version 1

Railroad diagram:



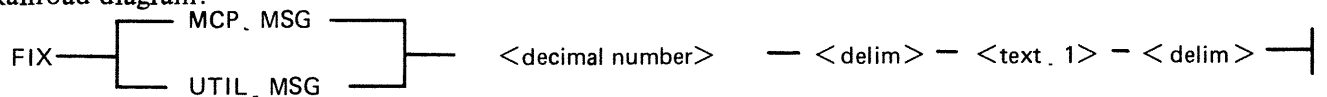
Version 2

Railroad diagram:



Version 3

Railroad diagram:



NOTE

A check is made on the number and value of escape codes given in text.1. and the requested item. These values must be identical for the FIX to be permitted. The order of the escape codes in text.1 may differ from the original text.

The first version scans the VERB section of the language file for all occurrences of text.1. It then replaces this with text.2. If text.1 contains more than one element, that is “SUPER ACCEPT”, the utility searches for the VERB “SUPER ACCEPT” in the verb section, irrespective of the number of spaces between the end of “SUPER” and the beginning of “ACCEPT”

that is, “SUPER ACCEPT” is treated the same as “ SUPER ACCEPT”

The second version of FIX tries to locate the item requested and replace it by text.1.

Example:

“FIX WORD @78@ THERE ARE @FF62@ FILES

changes the words “FF62 FILES OPEN7” to “THERE ARE FF62 FILES OPEN”

The third version is similar to the second, the only difference being that the FIX is on a complete message and not just a single item.

NOTE

When an item is displayed, either using FIND or LIST, the escape codes are displayed without surrounding “@” signs. These must be included around each escape code entered when executing a FIX, otherwise a mismatch in the number/value of escape codes may occur; or the message may not be exactly what is required. If an escape code is expanded in the displayed message (indicated by “+ +” either side of the text and immediately following the escape code), only the value of this escape code must be included in the FIX and not the text value it represents.

In the case of the MCP messages, a special field exists within the message which gives the MCP some indication of how this message should be routed. This field is displayed following the message number and bracketed by “@”. This value does not need to be entered in the new text; this is done by the utility.

Example:

Given the MCP message number 8, this message would be displayed as:

8 @08@ FF60 FF10 FF28 LOCKED

NOTE

A space is inserted between each escape code to make it more readable. It is not necessary to include this in the new text unless it is required. Spaces need only be included if there are two or more spaces between escape codes displayed.

To change LOCKED to UNLOCKED, the command would be:

FIX MCP.MSG 8

If the message is accepted, the utility displays this new text, expanding any escape codes where possible.

Delimiters

The value of “delim” is the first non-blank character the utility finds after the item number. The end of the text is the first character the utility finds which is the same as the first non-blank character found previously.

That is,

RTHIS IS AN EXAMPLER

would treat “R” as a delimiter.

Therefore the delimiter must be chosen carefully.

For example,

RTHIS IS ANOTHER EXAMPLER

would result in the text string

THIS IS ANOTHE

being accepted.

REMOVE

Railroad diagram:

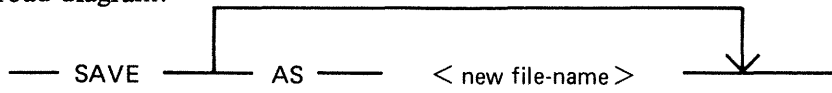


REMOVE removes the work files which have been created after a successful GET. While the utility has its work files present REMOVE may be used at any time.

After a REMOVE has been executed, only a GET or BYE is accepted.

SAVE

Railroad diagram:



This request can only be issued after a successful GET or RECOVER.

The SAVE command creates a new message file, incorporating all the alterations entered between the last GET (or RECOVER) and this current SAVE. This new file either has the same name as the source language file, or has the file name specified. The utility work file is removed once the SAVE has been completed.

Once the SAVE has finished, only a GET or BYE can be requested. Any work files that were present are removed, once the new file has been safely closed.

BYE

To terminate the run, enter BYE. However, this is only accepted if no FIXs have been entered since the last successful GET. If any FIXs have been entered, either REMOVE or a form of the SAVE command must be requested before BYE is allowed. This also applies after a successful RECOVER.

Output Format

Most of the messages which can be displayed are self-descriptive but some may need explanation.

1. **WORK FILES ARE PRESENT** followed by **RECOVER OR REMOVE REQUIRED**
This is displayed if, when executing the utility, files have been found with the same name as its work files. The user then has the choice of one of the displayed options.
2. **INVALID REQUEST <item>**
This is displayed if, for example, NEXT is requested and no previous successful FIND has been requested; or the last NEXT entered has reached the end of that section and has indicated that it has done so. (See "END <item>".)

3. ILLEGAL PARAMETER LIST <item>

The utility has found some illegal item in the request. If possible, the offending item is displayed.

4. FILE <file-name> IN USE or FILE <file-name> NOT FOUND

This is displayed if, when requesting GET, the utility cannot obtain the requested file for OUTPUT to, or the file is not there at all.

5. <file-name> IS NOT A RECOGNIZED SYSTEM FILE

This is displayed if the utility discovers that the requested file in a GET does not have the correct file-type or the correct record size, or that it contains no records.

6. FILE <file-name> HAS BEEN ALTERED followed by WORK FILE(S) BEING REMOVED

This may be displayed during the RECOVER request. The utility discovers that the source message file has had one or more of its sections altered since the current work files were last used. The utility then removes the current work files and releases the source message file.

7. UNABLE TO RECOVER followed by WORK FILE(S) BEING REMOVED

This is displayed if a request is made to RECOVER, but certain information necessary to make the RECOVERY successful is absent from the work files.

8. WORK FILE(S) ALREADY IN USE

The utility discovers work file(s) are present but at least one of them is already in use. The utility then goes to End of Job.

9. LAST OPERATION WAS <item>

This is displayed if RECOVERY is successful. It informs the user what the last operation was before the utility was abnormally terminated.

10. FILE <file-name> ALREADY PRESENT followed by SAVE OR REMOVE REQUIRED

The user has requested another message file but either there has been a successful RECOVERY, or at least one successful FIX, but no SAVE in the meantime.

11. MESSAGE FILE REQUIRED

The user has requested that a function is performed but no message file has been made present. Perform a MAKE or GET.

12. NO SPECIFICATION GIVEN

After a successful RECOVERY or GET, the language type of the message file is displayed.

14. INVALID CHARACTER IN IDENTIFIER <item>

15. UNKNOWN REQUEST <item>

The user has requested a function which is not recognized by the utility.

16. NO FILE-NAME IN PARAMETERS

After a request to GET a file, no <file-name> is included.

17. ENTER NEW LANGUAGE TYPE

After a successful GET, the language of the new file is required. This must be of the correct size (see "RESTRICTIONS OF USE").

18. LANGUAGE TYPE HAS TOO MANY CHARACTERS

The new language type does not follow the restrictions applied in "RESTRICTIONS OF USE".

19. END <item>

This is displayed after the last request has finished.

20. <item> VERB ENTRIES UPDATED

After doing a FIX of VERBS, the number of items translated is displayed.

21. INVALID DICTIONARY ENTRY

This is displayed if the user has attempted to FIX an item which does not exist.

22. THIS ITEM CANNOT BE TRANSLATED

The user has attempted to FIX an item which cannot be translated (see "RESTRICTIONS OF USE").

23. MIS-MATCH FOUND IN NEW TEXT

The user has requested a FIX to be made where a combination of the number and/or values of the escape codes does not match with the original version. No translation for this item occurs.

24. RECOVERY COMPLETED

After a successful RECOVERY, this message is displayed followed by "<item> WORK FILE LOADED" where this indicates the language type of the work file.

25. <item> RECORDS SAVED IN FILE <file-name>

This is displayed after completing a successful SAVE operation.

26. **** UTILITY LIMIT REACHED ****

The number of escape codes has exceeded the utility limit (see "RESTRICTIONS of USE").

27. COMMAND IN PROCESS ...

This is displayed at the start of a SAVE operation.

If a large number of translations has been done, especially in the MCP section, the SAVE operation may take several minutes to complete.

28. DISK <disk-name> NOT AVAILABLE

A request to GET or SAVE a file from/to a disk which the utility discovers is not present.

29. I/O ERROR DETECTED IN READ FROM FILE <file-name> or I/O ERROR DETECTED IN WRITE TO FILE <file-name> followed by ERROR AT RECORD NUMBER <item> and FUNCTION ABORTED

Some error occurred during a read/write operation, that is, a parity error from which the utility recovered. However, the requested function was not completed and must be tried again.

30. INVALID ESCAPE CODE IN FIX

This is displayed if, on requesting a FIX, an escape code is not of the correct format, that is, four hexadecimal characters bracketed by "@".

31. AU REVOIR UNTIL THE NEXT TIME

Normal terminating message.

32. TEXT LENGTH EXCEEDS UTILITY LIMIT

This is displayed when the user attempts to enter more characters, via the datacomm system, than the utility permits. (See "RESTRICTIONS OF USE".) The text must be re-entered in a shortened form.

33. REQUEST FOR INVALID RECORD FROM FILE <file-name>

The utility attempted to read a logical record number of zero or a record beyond the last entry in the file. This is usually because the first record of the message file has been set up incorrectly, or the section may be absent.

34. BREAK COMMAND RECEIVED

This is displayed if the user has requested a LIST option, is using the datacomm and has now sent "BRK" to the utility. This message is then followed by "FUNCTION ABORTED". The utility terminates the LIST command and waits for the next command.

35. DISK <disk id> LOCKED or <file name> NOT AVAILABLE – DISK LOCKED or WORK FILES NOT AVAILABLE – DISK LOCKED OR SYSTEM DISK IS LOCKED

Any one of these messages may be displayed if SYSMEM is locked on a disk which TRANSLATOR is attempting to access.

36. <file name> IS VERSION <version string>

This is displayed after a successful GET or RECOVER if the version field in the source file does not contain binary zeros.

System Parameters

The parameter fields, set up by the MCP and referenced in the SYSLANGUAGE file, are defined as follows:

1. MSGNUM Message number – used to index into the dictionary file.
2. EXECTASK Executing Task-id – used to indicate mix/program name of the task controlling this message display.
3. CPAPTR Communicate Parameter Area Pointer – used for accessing the specific communicate request and File Information Block/File Parameter Block fields.
4. DEVPTR Device Pointer – used for message about a particular peripheral.
5. DATE System maintained date.
6. TIME System maintained time, if the system has a real time clock.
7. SYSFILE System File Pointer – used for printing information about a task other than the task controlling the message display.
8. PARAM1 General purpose – intended to contain one byte parameters.
9. PARAM2 General purpose – intended to contain two byte parameters.
10. PARAM3 General purpose – intended to contain two byte parameters.
11. PARAM4 General purpose – intended to contain three byte parameters.

Escape Codes

Each escape code identifies an intrinsic function which will operate on the specified parameter to carry out the appropriate action. This action is mostly concerned with extracting information from the MCP or program structures and placing the information (or a more meaningful derivative of it) in the output message.

Escape codes are two byte fields. The first byte has a hexadecimal value of @FF@ to distinguish it from the ASCII text and the second byte identifies the function to be carried out.

The MCP and utilities use escape codes in different ways. The following subsections contain details of MCP and utility usage of escape codes and their meanings.

Escape Codes – MCP

Escape codes with the values @FF80@ – @FFFF@ do not identify intrinsic functions but are internal to the local language directory file. They are references from message entries to descriptive text in the file which is to be included in the message. In the meaning column for the following escape codes, reserved parameter fields are named; the section System Parameters gives a brief description of how the parameters are used. MFID stands for Multifile Identifier, that is, a disk name or a tape name.

Value	Meaning
FF00	Terminates system word or message description entry.
FF01	Terminates message.
FF08	Carriage return, line feed.
FF09	Machine type.
FF10	Print mix/program name for current task (EXECTASK).
FF11	Print mix/program name for task named in PARAM1.
FF12	Print mix/program name for task named in SYSFILE.
FF13	Print priority class of task named in PARAM1.
FF14	Print mix/program name for current task (EXECTASK) unless this task's priority class field indicates suppression.
FF18	Reprint the last SCL input up to the position indicated by PARAM2.
FF19	Print text from segment indicated by CPAPTR.
FF20	Print date as DD MMM YY (for example, 01 AUG 82).
FF21	Print date as YYDDD (for example, 81121).
FF22	Optionally print day of week (for example, MON).
FF23	Optionally print time as HHMM, if clock is present.
FF24	Print BCD contents of PARAM4 in the form NN.NN.NN.
FF28	Print device mnemonic from configuration table entry pointed to by DEVPTR.
FF29	Print file-name from configuration table entry indicated by DEVPTR followed by “/” if it is an MFID and followed by reel number if it is magnetic tape. If not named, suppress print.
FF2A	Print device mnemonic indicated by DEVPTR.
FF2B	Print serial number for the device indicated by DEVPTR if device is tape or disk.
FF30	Print segment number (in decimal) of File Information Block pointed to by CPAPTR.
FF31	Print segment number (in decimal) of File Parameter Block indicated by CPAPTR.
FF32	Print MFID/ from File Parameter Block indicated by CPAPTR.
FF33	Print [MFID/] file-name [reel number] from the File Parameter Block indicated by CPAPTR and the device kind in File Parameter Block.
FF34	Print device mnemonic from the File Parameter Block indicated by CPAPTR.
FF35	Print MFID/ indicated by SYSFILE.
FF36	Print [MFID/] file-name [reel number] pointed to by SYSFILE.
FF37	Print [MFID/] file-name [reel number] from File Parameter Block indicated by the actual Communicate Parameter Area of the mix defined by PARAM1.
FF38	Print device mnemonic from File Parameter Block indicated by the actual Communicate Parameter Area of the mix defined by PARAM1, using the device kind table.
FF39	Print the compilation date of the MCP in the format YYDDD.
FF40	Print verb indicated by CPAPTR using verb table.
FF41	Print verb indicated by PARAM1 using verb table.
FF42	Print hexadecimal value of verb indicated by CPAPTR.

(continued)

Value	Meaning
FF43	Print I/O function indicated by PARAM1 using system word table.
FF48	Print job state indicated by PARAM2 using system word table.
FF49	Print wait key indicated by PARAM2 using system word table.
FF4A	Optionally print [MFID/] file-name from SYSFILE using system word table.
FF4B	Print SCL function identity indicated by SYSFILE using system word table.
FF50	Print primary status from configuration table entry indicated by DEVPTR.
FF51	Print secondary status from configuration table entry indicated by DEVPTR.
FF52	Print number of files information, using system word table if configuration table entry indicated by DEVPTR describes a disk.
FF53	Print in use by information, from configuration table indicated by DEVPTR using system word table. If not in use, suppress print.
FF60	Print message number indicated by MSGNUM in decimal and enclosed within "<" and ">".
FF61	Print PARAM1 in decimal.
FF62	Print PARAM2 in decimal.
FF63	Print PARAM3 in decimal.
FF64	Print PARAM4 in decimal.
FF70	Optionally print device status information indicated by PARAM4 in hexadecimal.
FF71	Print PARAM1 in hexadecimal.
FF72	Print PARAM2 in hexadecimal.
FF73	Print PARAM3 in hexadecimal.
FF74	Print PARAM4 in hexadecimal.
FF75	Print memory where – PARAM1/PARAM4 is start address in slice/segment and offset format and PARAM2 is the length.
FF80 to FFFF	Use ((Opcode – @80@) * 2) and index into the message description table.

Utility Escape Codes

The escape codes have the format of @FFxy@ in hexadecimal, where x defines the function to be performed on parameter y (the first parameter is parameter 0) of the display procedure.

The functions available are as follows, with all values of x being hexadecimal.

1. Insert text, no alterations (x = 0 to 9).
2. Convert a 1, 2 or 3 byte binary value into decimal character string (x = A).
3. Insert the diskname and filename of the file whose file segment number is the supplied parameter in the format <disk-name>/<file-name>. If the diskname is '0000000', the diskname and "" are omitted (x = B).
4. Insert text, but truncate after last non-space character, that is, remove trailing spaces (x = C). For example, "THIS IS ANFFC0 EXAMPLE" would produce "THIS IS ANOTHER EXAMPLE", if the first parameter inserted was the text "OTHER".
5. Convert a binary value of between 1 and 127 bytes in length into a hexadecimal ASCII character string. For example, the hexadecimal number F3219 would be converted to "F3219". (x = D)

Utility Message Usage – SYSLANGUAGE Version 03.04.05

Message Numbers V Utilities

Message Number	Message Text	Utility Name
1	INVALID DICTIONARY ENTRY FFA0	ICMD TRANSLATOR COPY DUMPADISK LD LIST PL SYS-SUPERUTL TAPELR TAPEPD TL COMPARE LR XD KA FL CREATE AMEND UPDATE CHECK.DISK PPID LOAD.VFU DSKUTL RL
2	ILLEGAL PARAMETER LIST FFC0	ICMD SYCOPY BUILDTRANS CHECKADUMP COPY DUMPADISK FS LD LIST PL SYS-SUPERUTL TAPELR TAPEPD TL TRANSLATOR COMPARE CHECK.DISK LR XD KA CREATE AMEND

3 INVALID CHARACTER IN IDENTIFIER FFC0

UPDATE
FL
PPID
KEY.CHECK
LOAD.VFU
DSKUTL
RL
SAU.INIT
ICMD
BUILDTRANS
CHECKADUMP
COPY
DUMPADISK
FS
LD
LIST
PL
SYS-SUPERUTL
TAPELR
TAPEPD
TL
TRANSLATOR
COMPARE
CHECK.DISK
LR
XD
KA
CREATE
AMEND
UPDATE
FL
PPID
KEY.CHECK
LOAD.VFU
DSKUTL
RL
SAU.INIT
CONFIGURER
ICMD
BUILDTRANS
CHECKADUMP
COPY
DUMPADISK
FS
PL
LIST
TAPELR
TAPEPD
TL
TRANSLATOR

4 FFB0 NOT FOUND

5 FFB0 NOT AVAILABLE - IN USE

WL
COMPARE
LR
KA
CREATE
AMEND
UPDATE
FL
CHECK.DISK
KEY.CHECK
LOAD.VFU
DSKUTL
SAU.INIT
DA
ICMD
BUILDTRANS
CHECKADUMP
COPY
LIST
PL
DUMPADISK
FS
SYS-SUPERUTL
TAPELR
TAPEPD
TL
TRANSLATOR
WL
COMPARE
LR
KA
CREATE
AMEND
UPDATE
FL
KEY.CHECK
LOAD.VFU
DSKUTL
SAU.INIT
DA
PB
ICMD
BUILDTRANS
CHECKADUMP
COPY
DUMPADISK
FS
LD
LIST
PL

6 NO SPECIFICATION GIVEN

7 DISK FFC0 NOT AVAILABLE

8 FFC0 CONTAINS -
9 FFC0 ON FFC1 CONTAINS -
10 FFC0 FFC1 FFC2

SYS-SUPERUTL
TAPELR
TAPEPD
TRANSLATOR
COMPARE
LR
XD
SAU.INIT
CREATE
AMEND
UPDATE
FL
KA
KEY.CHECK
LOAD.VFU
DSKUTL
RL
ICMD
CHECKADUMP
COPY
DUMPADISK
FS
LD
LIST
PL
SYS-SUPERUTL
TL
TRANSLATOR
SCR
COMPARE
TAPELR
TAPEPD
CHECK.DISK
LR
XD
KA
CREATE
AMEND
UPDATE
FL
KEY.CHECK
PPID
LOAD.VFU
DSKUTL
SAU.INIT
DA
SYS-SUPERUTL
SYS-SUPERUTL
SYS-SUPERUTL
TAPEPD

11	FFC0 REQUIRES OVERFLOW DISK FFC1	SYS-SUPERUTL
12	ON LINE FFC0 FFC1	SYS-SUPERUTL
13	NOT ON LINE FFC0	SYS-SUPERUTL
14	NO FILES FOUND IN THE FAMILY FFC0	SYS-SUPERUTL
15	FFB0 IS A SYSTEM FILE	SYS-SUPERUTL
16	FFB0 REMOVED	ICMD COPY LD PL SYS-SUPERUTL TL LOAD.VFU DUMPADISK BUILDTRANS SYS-SUPERUTL
17	INDEXED PAIR FFB0 FFB1 REMOVED	SYS-SUPERUTL
18	INDEXED PAIR FFB0 FFB1 NOT REMOVED -	SYS-SUPERUTL
19	DATA FILE FFB0 NOT FOUND	SYS-SUPERUTL
20	DATA FILE FFB0 IN USE	SYS-SUPERUTL
21	FFC0 NOT REMOVED - NOT FOUND	SYS-SUPERUTL
22	FFC0 NOT REMOVED - IN USE	SYS-SUPERUTL
23	FFB0 NOT REMOVED - SYSTEM FILE	SYS-SUPERUTL
24	NO FILES FOR REMOVAL IN FAMILY FFC0	SYS-SUPERUTL
25	INVALID CHARACTER IN IDENTIFIER FFC0 - WARNING	ICMD DUMPADISK LD SYS-SUPERUTL
26	FFC0 CHANGED TO FFC1	SYS-SUPERUTL
27	FFC0 NOT CHANGED - NOT FOUND	SYS-SUPERUTL
28	FFC0 NOT CHANGED - IN USE	SYS-SUPERUTL
29	FILE IDENTIFIER TOO LONG FFC0	SYS-SUPERUTL
30	FFC0 NOT CHANGED - ILLEGAL REQUEST	SYS-SUPERUTL
31	FFC0 NOT CHANGED - FFC1 ALREADY ON DISK	SYS-SUPERUTL
32	KEY FILE FFC0 NOW POINTS TO DATA FILE FFC1	SYS-SUPERUTL
33	NO FILES FOR CHANGING IN THE FAMILY FFC0	SYS-SUPERUTL
34	FFC0 IS AN INVALID REQUEST	SYS-SUPERUTL
35	FFC0 - FFA1 SECTORS; AVAILABLE FFA2; TEMPORARY FFA3	SYS-SUPERUTL
36	FFC0 - FFA1 SECTORS; AV. FFA2; TEMP. FFA3	SYS-SUPERUTL
37	FFC0 IS AN UNACCEPTABLE RESPONSE FOR KX	SYS-SUPERUTL
38	TL ZIP FAIL - LOG FILES NOT CONSOLIDATED	SYS-SUPERUTL
39	LOGGING WILL NOT BE SWITCHED ON	SYS-SUPERUTL
40	UNABLE TO OPEN MINIMUM LOG FILES	SYS-SUPERUTL
41	UNABLE TO ZIP FFC0 AT WARMSTART	SYS-SUPERUTL
42	IR REQUIRED BEFORE FFC0 CAN BE DONE	SYS-SUPERUTL
43	FUNCTION ABORTED	BUILDTRANS TRANSLATOR
44	INVALID ESCAPE CODE IN FIX	TRANSLATOR
45	AU REVOIR UNTIL THE NEXT TIME	BUILDTRANS TRANSLATOR

46	TEXT LENGTH EXCEEDS UTILITY LIMIT	BUILDTRANS TRANSLATOR CONFIGURER
47	REQUEST FOR INVALID RECORD FROM FILE FFB0	TRANSLATOR
48	BREAK COMMAND RECEIVED	BUILDTRANS TRANSLATOR
49	FFC0 NOT A RECOGNIZED DUMP TAPE	SYCOPY CHECKADUMP LD TAPELR TAPEPD TAPED
50	FFC0 TAPE FFC1 <FFC2> DUMPED ON FFC3 FFC4 CONTAINS -	
51	FFB0 NOT LOADED - ALREADY ON DISK	DUMPADISK LD
52	ALTHOUGH WITH DIFFERENT ATTRIBUTES	DUMPADISK LD
53	FFB0 LOAD DISCREPANCY	DUMPADISK LD
54	FFB0 LOADED	DUMPADISK LD
55	FFB0 DUMPED	DUMPADISK LD
56	NO FILES TO DUMP	DUMPADISK LD
57	FILE FFB0 NOT DUMPED - HAS BEEN REMOVED	DUMPADISK LD
58	FILE FFB0 NOT DUMPED - IN OUTPUT USE	DUMPADISK LD
59	FILE FFB0 NOT DUMPED - HAS BEEN ALTERED	DUMPADISK LD
60	FFB0 DUMP DISCREPANCY	DUMPADISK LD
61	FFC0 FFC1 ON FFC2 FFC3 FOR FFC4	DUMPADISK LD
62	DUPLICATE - FFC0 ALREADY BEING DUMPED	DUMPADISK LD
63	NO FILES TO LOAD	DUMPADISK LD
64	FFC0 NOT DUMPED - DATA FILE NOT ONLINE	DUMPADISK LD
65	DUPLICATE - FFC0 ALREADY BEING LOADED	DUMPADISK LD
66	FFB0 - DATA FILE NOT FOUND ON TAPE FOR LOAD	LD
67	LOAD ABANDONED - ERROR IN DUMP	LD
68	TAPE SHOULD HAVE BEEN PURGED	CHECKADUMP LD
69	UTILITY LIMIT REACHED AT FILE FFC0	DUMPADISK LD
71	ILLEGAL FILE NAME FFC0	BUILDTRANS PL
72	NO LOG FILES FOUND FOR CONSOLIDATION	TL

73	NO ACTIVE LOG FILE FOUND	TL
74	TRANSFER COMPLETED	TL
75	NO READY-TO-TRANSFER FILES FOUND	TL
76	CONTINUING PROCESS	DUMPADISK
		TL
		PL
77	ILLEGAL USE OF RECOVER PARAMETER	TL
78	ACTIVE FILE NOT CONSOLIDATED	TL
79	ILLEGAL VALUE FFC0	BUILDTRANS
		PL
80	ILLEGAL OPTION FFC0	PL
81	NO TIME SPECIFICATION GIVEN AFTER FFC0	PL
82	NO FFC0 NUMBER GIVEN	PL
		CONFIGURER
83	FFC0 OPTION ALREADY SET	BUILDTRANS
		PL
		TL
84	ILLEGAL FFC0 SPECIFICATION	PL
85	FFB0 IS NOT A LOG FILE	PL
86	PARITY ON RECORD FFA0 IN LOG FILE FFC1	SYS-SUPERUTL
87	CANNOT LOCATE DESIRED LOG ENTRY	SYS-SUPERUTL
88	CANNOT READ RECORD FFA0 IN LOG FILE FFC1	SYS-SUPERUTL
89	RECALL ENTRY TOO LONG - TRUNCATED	SYS-SUPERUTL
90	UNABLE TO OPEN FILE FOR FFC0	SYS-SUPERUTL
91	CANNOT RECALL <FFC0> BEYOND THIS POINT	SYS-SUPERUTL
92	PSEUDO-DISK FFC0 ON DISK FFC1 NOT AVAILABLE	SYS-SUPERUTL
93	PSEUDO-DISK FFC0 ON DISK FFC1 CONTAINS -	SYS-SUPERUTL
94	FFC0 ON PSEUDO-DISK FFC1 ON DISK FFC2	SYS-SUPERUTL
	CONTAINS -	
95	FFC0 FFC1 - FFA2 SECTORS; AVAILABLE FFA3;	SYS-SUPERUTL
	TEMPORARY FFA4	
96	FFC0 FFC1 - FFA2 SECTORS; AV. FFA3; TEMP. FFA4	SYS-SUPERUTL
97	UTILITY LIMIT REACHED ON PSEUDO-DISK FFC0	SYS-SUPERUTL
98	LIMIT AT CHANGE OF FFC0 TO FFC1	SYS-SUPERUTL
99	DISK FFC0 IS A PSEUDO-DISK	SYS-SUPERUTL
		CHECK.DISK
		LR
		XD
		KA
		PPID
		DA
100	FFC0 ON PSEUDO-DISK FFC1 INCOMPLETE	SYS-SUPERUTL
101	FILE LIST FOR FFC0 MAY BE INCOMPLETE	DUMPADISK
		LD
		SYS-SUPERUTL
		FL
102	FILE LIST MAY BE INCOMPLETE	COPY
		DUMPADISK
		LD
		SYS-SUPERUTL
		COMPARE
		LR
		FL

103	COMMENCING LOG FILE CONSOLIDATION	SYS-SUPERUTL
104	COMMENCING LOG FILE REALLOCATION	SYS-SUPERUTL
105	LOGGING IS INITIATED ON FFC0 FFC1 FFC2	SYS-SUPERUTL
106	WORK FILE(S) ARE PRESENT	BUILDTRANS TRANSLATOR
107	RECOVER OR REMOVE REQUIRED	BUILDTRANS TRANSLATOR
108	INVALID REQUEST FFC0	BUILDTRANS PL TRANSLATOR
109	FFB0 IS NOT A RECOGNIZED SYSTEM FILE	BUILDTRANS SYS-SUPERUTL TRANSLATOR
110	FILE FFB0 HAS BEEN ALTERED	LOAD.VFU BUILDTRANS TL TRANSLATOR
111	WORK FILE(S) BEING REMOVED	BUILDTRANS TRANSLATOR
112	UNABLE TO RECOVER	BUILDTRANS TRANSLATOR
113	WORK FILE(S) ALREADY IN USE	BUILDTRANS TRANSLATOR
114	LAST OPERATION WAS FFC0	BUILDTRANS TRANSLATOR
115	FFB0 FILE ALREADY PRESENT	BUILDTRANS TRANSLATOR
116	SAVE OR REMOVE REQUIRED	BUILDTRANS TRANSLATOR
117	MESSAGE FILE REQUIRED	TRANSLATOR
118	FFC0 MESSAGE FILE LOADED	TRANSLATOR
119	UNKNOWN REQUEST FFC0	BUILDTRANS TRANSLATOR
120	NO FILE NAME IN PARAMETERS	BUILDTRANS PL TRANSLATOR
121	ENTER NEW LANGUAGE TYPE	TRANSLATOR
122	LANGUAGE TYPE HAS TOO MANY CHARACTERS	TRANSLATOR
123	END FFC0	SYS-SUPERUTL TAPEPD TRANSLATOR
124	FFA0 VERB ENTRIES UPDATED	LD TRANSLATOR
125	THIS ITEM CANNOT BE TRANSLATED	TRANSLATOR
126	MIS-MATCH FOUND IN NEW TEXT	TRANSLATOR
127	RECOVERY COMPLETED	BUILDTRANS TRANSLATOR
128	FFC0 WORK FILE LOADED	TRANSLATOR
129	FFA0 RECORDS SAVED IN FILE FFB1	TRANSLATOR
130	**** UTILITY LIMIT REACHED ****	SYCOPY BUILDTRANS PL TRANSLATOR

131	COMMAND IN PROCESS ...	BUILDTRANS TRANSLATOR
132	I/O ERROR DETECTED IN WRITE TO FILE FFB0	BUILDTRANS DUMPADISK PL SYS-SUPERUTL TL TRANSLATOR WL
133	I/O ERROR DETECTED IN READ FROM FILE FFB0	CONFIGURER BUILDTRANS DUMPADISK PL TL TRANSLATOR WL
134	ERROR AT RECORD NUMBER FFA0	CONFIGURER BUILDTRANS DUMPADISK PL TL TRANSLATOR WL
135	NO ENTRIES FOUND WITHIN PARAMETERS	CONFIGURER
136	UNRECOGNIZED DEVICE IN ERROR ENTRY	PL
137	INVALID DIRECTORY ON TAPE FFC0	CHECKADUMP
138	COMPARISON ERROR ON FFB0	SYCOPY
139	ON DISK FILE HEADERS	CHECKADUMP
140	FILE FFC0 NOT FOUND FOR CHECK	CHECKADUMP
141	FILE FFC0 NOT AVAILABLE FOR CHECK	CHECKADUMP
142	AROUND RECORD FFA0	SYCOPY
143	DIFFERING FILE SIZES	CHECKADUMP
144	DIFFERING FILE TYPES	CHECKADUMP
145	DIFFERING RECORD SIZES	SYCOPY
146	DIFFERING BLOCK SIZES	CHECKADUMP
147	NO DISCREPANCIES BETWEEN	CHECKADUMP
148	DUMP TAPE FFC0 AND DISK FFC1	CHECKADUMP
149	DISCREPANCIES FOUND BETWEEN	CHECKADUMP
150	COMPARISON TERMINATED	CHECKADUMP
151	RECORD SIZE TOO GREAT	COPY FS LIST
152	FFB0 SQUASHED FROM FFA1 TO FFA2 RECORDS	FS
153	KEYFILE FFC0 RECONSTRUCTED	FS
154	KEYFILE FFC0 SORT FAILURE – END OF JOB	FS
155	ZIP TO SORT FAILURE – END OF JOB	FS
159	JOB COMPLETED	CONFIGURER
163	ENTER TEXT TO BE ZIPPED AT WARMSTART	CONFIGURER
165	ENTER POWER OFF MESSAGE	CONFIGURER

166	SPO LOGGING REQUIRED? - <Y OR N>	CONFIGURER
167	** INVALID SELECTION - RETRY **	CONFIGURER
169	ILLEGAL FILE SIZE - FFC0	CONFIGURER
171	ENTER LOCAL SPO OPTION - <FFC0 OR FFC1>	CONFIGURER
173	<MINIMUM 32, MAXIMUM 16383>	CONFIGURER
174	ENTER FILE-ID OF REQUIRED MESSAGE FILE	CONFIGURER
175	CP. FFC0	CP
176	CP. NUMBER TOO LARGE	CP
177	CP. ILLEGAL NUMBER FFC0	CP
178	CP. MISSING NUMBER	CP
179	CP. ILLEGAL HEX. NUMBER	CP
180	CP. OVERFLOW	CP
181	CP. DIVISION BY ZERO	CP
182	CP. INVALID OPERATOR FFC0	CP
183	CP. MISSING OPERATOR	CP
184	FFC0 <DATAFILE> NOT FOUND	KEY.CHECK
185	FFC0 IS NOT A KEYFILE	KEY.CHECK
186	FFC0 IS A NULL-KEY KEYFILE	KEY.CHECK
187	READ ERROR ON KFPB OF FFC0	FS
		KEY.CHECK
188	READ ERROR ON KEYFILE FFC0	KEY.CHECK
189	UNABLE TO OPEN FFC0 <INDEXFILE>	KEY.CHECK
190	FFC0 <DATAFILE> IN USE	KEY.CHECK
191	FFC0 <DATAFILE> INCOMPATIBLE WITH KEYFILE	KEY.CHECK
192	DUMP ABANDONED - TAPE SHOULD BE PURGED	LD
193	READ ERROR ON LABEL OF FFC0	CHECK.DISK
		PPID
		DSKUTL
		DA
194	READ ERROR ON PPIT OF FFC0	COPY
		DUMPADISK
		LD
		LIST
		SYS-SUPERUTL
		COMPARE
		LR
		FL
		PPID
		DSKUTL
195	FFB0 IS NOT TYPE SOURCE OR DATA	ICMD
		CHECKADUMP
		COPY
		DUMPADISK
		FS
		LD
		LIST
		PL
		SYS-SUPERUTL
		TAPELR
		TAPEPD
		TL
		COMPARE
		LR

196 SPECIFICATION TOO LONG

KA
CREATE
AMEND
UPDATE
FL
PPID
KEY.CHECK
LOAD.VFU
DSKUTL
SAU.INIT
ICMD
CHECKADUMP
DUMPADISK
COPY
FS
LD
LIST
PL
SYS-SUPERUTL
TAPELR
TAPEPD
TL
COMPARE
LR

197 SORT-MERGE SPEC WARNING= >FFC0
198 SORT-MERGE SPEC ERRORS= >FFC0
199 SORT-MERGE INTRINSIC NOT INVOKED
200 MISSING SORT-MERGE SPECIFICATION
201 ILLEGAL SORT-MERGE SPECIFICATION
202 SORT-MERGE SPEC FILE NOT DATA OR SOURCE
203 SORT-MERGE SPEC FILE RECORD SIZE TOO BIG
204 FIX AND REPLY "OK" ELSE <NULL>
205 FFB0 UNAVAILABLE ON DISK
206 FILE FFC0 UNAVAILABLE
207 FILE FFC0 UNAVAILABLE ON SYSTEM DISK
208 SORT-MERGE SPEC CORRECTION APPLIED
209 SORT-MERGE INTRINSIC ZIP FAILURE
210 INTRINSIC CODE FILE NOT FOUND
211 INSUFFICIENT MEMORY TO LOAD INTRINSIC
212 NO DISK SPACE FOR INTRINSIC VM FILE
213 NO MIX SLOT AVAILABLE FOR INTRINSIC
214 INTRINSIC USERCOUNT ERROR
215 ERROR ON DISK CONTAINING INTRINSIC

KA
CREATE
AMEND
UPDATE
FL
PPID
KEY.CHECK
LOAD.VFU
DSKUTL
SAU.INIT
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE
SORT-MERGE

216	INTRINSIC CODE FILE CORRUPT	SORT-MERGE
217	INTRINSIC DS/DP'ED	SORT-MERGE
218	NO FILES FOUND FOR CHECK	CHECKADUMP
219	NO FILES IN FAMILY FFC0 FOUND FOR CHECK	CHECKADUMP
221	SECONDARY LOG FILE FFB0 CREATED	TL
222	ERROR NOTIFIED ON READING DISK FFC0	CHECK.DISK
223	CHECK.DISK ON FFC0 COMPLETED -	CHECK.DISK
224	NO ERRORS	CHECK.DISK
225	FFC0 CONSISTENT ERRORS NOTIFIED	CHECK.DISK
226	ONE CONSISTENT ERROR NOTIFIED	CHECK.DISK
227	PLEASE POWER OFF DISK FFC0	CHECK.DISK
228	DISK FFC0 EXCEEDS BAD SECTOR LIMIT	CHECK.DISK
229	DISK FFC0 EXPIRING - BACKUP RECOMMENDED	CHECK.DISK
230	DIFFERING GENERATION NUMBERS	CHECKADUMP
231	BAD ATTRIBUTES SPECIFIED	COPY
		LIST
232	FFB0 - DATA FILE NOT FOUND	COPY
		LIST
233	KEY FILE FFB0 OR DATA FILE IN USE	COPY
		LIST
234	TABLE SIZE EXCEEDED	LR
		KA
235	KEY FILE FFB0 OR DATA FILE NOT FOUND	COPY
		LIST
236	FFB0 NOT ACCEPTABLE - RECORD SIZE OF FFA1 EXCEEDS MAXIMUM FOR THIS RUN - RESUBMIT	COPY
		LIST
237	FFA0, FFA1 IN FFB2 NOT COPIED	COPY
238	FFA0, EOF IN FFB2 NOT COPIED	COPY
239	FFB0 EXHAUSTED DURING RANGE - FF01 - FF02	COPY
		LIST
240	FFB0 EXHAUSTED DURING RANGE - FF01 FFA2	COPY
		LIST
241	FFB0 EXHAUSTED DURING RANGE - FF01 EOF	COPY
		LIST
242	FFB0 EXHAUSTED DURING FFA1, FFA2	COPY
		LIST
243	FFB0 EXHAUSTED DURING FFA1, EOF	COPY
		LIST
244	SELECTIONS OF RECORDS FROM FFA0 IGNORED	COPY
		LIST
245	NO RECORDS FOR COPYING FROM FFB0	COPY
246	FFB0 TO FFB1 COPY FAILURE - NO RECORDS IN KEY FILE	COPY
247	FFB0 TO FFB1 COPY FAILURE - INPUT RECORD NO. FFA2, OUTPUT RECORD NO. FFA3 PERMANENT ERROR ON INPUT FILE	COPY
248	FFB0 TO FFB1 COPY FAILURE - INPUT RECORD NO. FFA2, OUTPUT RECORD NO. FFA3 PERMANENT ERROR ON OUTPUT FILE	COPY
249	FFB0 TO FFB1 COPY FAILURE - INPUT RECORD NO. FFA2, OUTPUT RECORD NO. FFA3 OUTPUT FILE TOO SMALL	COPY

250	FFB0 TO FFB1 COPY FAILURE – INPUT RECORD KEY FF02, OUTPUT RECORD NO. FFA3 PERMANENT ERROR ON INPUT FILE	COPY
251	FFB0 TO FFB1 COPY FAILURE – INPUT RECORD KEY FF02, OUTPUT RECORD NO. FFA3 PERMANENT ERROR ON OUTPUT FILE	COPY
252	FFB0 TO FFB1 COPY FAILURE – INPUT RECORD KEY FF02, OUTPUT RECORD NO. FFA3 OUTPUT FILE TOO SMALL	COPY
253	FFB0 TO FFB1 COPIED	ICMD COPY
254	FFB0 TO FFB1 BAD ATTRIBUTES	COPY
255	FFB0 NOT FOUND FOR EXTENDING	COPY
256	FFB0 TO FFB1 ILLEGAL REQUEST	COPY
257	FFB0 TO FFB1 ILLEGAL REQUEST SELECTION CRITERIA IGNORED	COPY
258	FFB0 TO FFB1 ILLEGAL REQUEST EXTENDING FLAG IGNORED	COPY
259	FAMILY FFB0 TO FFB1 COPIED	COPY
260	SECTOR FFC0 DENOTED BAD IN DIRECTORY	CHECK.DISK
261	SECTOR FFC0 NOT DENOTED BAD IN DIRECTORY	CHECK.DISK
262	DIRECTORY ERROR ON DISK FFC0	DUMPADISK
263	FFC0 BAD SECTOR<S> NOT DENOTED IN DIRECTORY	CHECK.DISK
264	SECTOR FFC0 BEYOND END OF DISK FFC1	CHECK.DISK XD
265	ILLEGAL RANGE FROM SECTOR FFC0	CHECK.DISK
266	CHECK.DISK ON FFC0 AREA FFC1 – FFC2 COMPLETED –	CHECK.DISK
267	CHECK.DISK ON FFC0 SECTOR FFC1 – NO ERROR	CHECK.DISK
268	SECTORS FOR XD NOT AVAILABLE	XD
269	CANNOT DELETE SECTORS FROM FFC0	XD
270	AVAILABLE TABLE FULL – ENTRY FFC0 FFC1 LOST	XD
271	FFC0 SECTORS FROM FFC1 DELETED	XD
272	NO PPIT ON SYSTEM	PPID DSKUTL
273	FF00 – PHYS. UNIT FFC1 FFC2	PPID
274	FF00 – PSEUDO ON FFC1	PPID
275	FF00 – PSEUDO UNRESTRICTED	PPID
276	FFC0 ALREADY DEFINED IN SYSVFU.LIB	LOAD.VFU
277	FFC0 NOT DEFINED IN SYSVFU.LIB	LOAD.VFU
278	READ ERROR ON SYSVFU.LIB	LOAD.VFU
279	WRITE ERROR ON SYSVFU.LIB	LOAD.VFU
280	FFC0 DEFINED IN SYSVFU.LIB	LOAD.VFU
281	FFC0 DELETED FROM SYSVFU.LIB	LOAD.VFU
282	PRINTER LPFFC0 FORMATTED WITH FFC1	LOAD.VFU
283	PRINTER LPFFC0 NOT FORMATTED WITH FFC1	LOAD.VFU
284	ILLEGAL PARAMETER LIST – BAD ATTRIBUTES	CREATE AMEND UPDATE
285	NO OUTPUT GENERATED BY KA FOR DISK FFC0	KA
286	FFB0 FILE TYPE IS NOT DATA	FS

	CREATE
	AMEND
	UPDATE
	KEY.CHECK
287	ILLEGAL PARAMETER LIST – TABS ERROR
	CREATE
	AMEND
	UPDATE
288	OUTPUT FILE FFB0 TOO SMALL
	PL
	CREATE
	AMEND
	UPDATE
289	PERMANENT ERROR ON OUTPUT FILE FFB0
	CREATE
	AMEND
	UPDATE
290	OK TO PO DRIVE FFC0
	DUMPADISK
291	INSUFFICIENT AREA TO DUMP TO ON DRIVE FFC0
	DUMPADISK
292	FILE FFC0 NOT LOADED – FFC1 WHEN DUMPING
	DUMPADISK
293	OVERFLOW DISK REQUIRED TO CONTINUE DUMP
	DUMPADISK
294	DISK FFC0 DOES NOT BELONG TO THIS LOAD
	DUMPADISK
295	OK TO PO DISK FFC0
	DUMPADISK
296	FFB0 – DATA FILE NOT FOUND ON DISK FOR LOAD
	DUMPADISK
297	DISK FFC0 REQUIRED TO CONTINUE FFC1
	DUMPADISK
298	DISK FFC0 IS NOT REQUIRED FOR THIS LOAD
	DUMPADISK
299	MAKE DISK FFC0 AVAILABLE AND ENTER “OK” OR MAKE ALTERNATIVE DISK AVAILABLE AND ENTER NAME
	DUMPADISK
300	FFC0 IS NOT A RECOGNIZED BACKUP DISK
	DUMPADISK
301	INVALID CHARACTER IN IDENTIFIER FFB0
	COPY
	LIST
	LR
302	INVALID CHARACTER IN IDENTIFIER FFB0 – WARNING
	COPY
	LIST
	PPID
303	NO FILES FOUND IN THE FAMILY FFB0
	COPY
	COMPARE
304	FFB0 FILE IDENTIFIER TOO LONG
	COPY
	COMPARE
305	FILE LIST FOR FFB0 MAY BE INCOMPLETE
	COPY
	COMPARE
	LR
306	ATTRIBUTE VALUE MISSING ... FF00
	MODIFY
307	KEYWORD IN ERROR ... FF00
	MODIFY
308	ATTRIBUTE-VAL INCONSISTENT ... FF00
	MODIFY
309	INCORRECT ATTRIBUTE ... FF00
	MODIFY
310	WARNING + + + DEVICE-MYUSE INCONSISTENT
	MODIFY
311	WARNING + + + FILE.SIZE TOO LARGE
	MODIFY
312	WARNING + + + TOO MANY BUFFERS
	MODIFY
313	WARNING + + + REC NOT INTEGRAL OF BUF
	MODIFY
314	CODE.FILE NAME IN ERROR ... FF00
	MODIFY
315	FILE NAME NOT FOUND ... FF00
	MODIFY
316	CURRENCY SYMBOL EXPECTED ... FF00
	MODIFY

317	NUMERIC ATTRIBUTE-VAL REQD ... FF00	MODIFY
318	FILE NOT SPECIFIED ... FF00	MODIFY
319	PPB NOT SPECIFIED ... FF00	MODIFY
320	NOT INDEXED FILE ... FF00	MODIFY
321	NOT COBOL PROGRAM FILE ... FF00	MODIFY
322	NOT MPL PROGRAM FILE ... FF00	MODIFY
323	NOT FPB ATTRIBUTE ... FF00	MODIFY
324	NOT PPB ATTRIBUTE ... FF00	MODIFY
325	ILLEGAL FILE NAME ... FF00	MODIFY
326	MUST HAVE 0 < KEY.LENGTH < 29 ... FF00	MODIFY
327	NUMBER TOO BIG ... FF00	MODIFY
328	CANNOT OPEN THAT CODE.FILE ... FF00	MODIFY
329	CODE.FILE - BAD FILETYPE ... FF00	MODIFY
330	CODE.FILE IN USE ... FF00	MODIFY
335	FFA0, FFA1 IN FFB2 NOT LISTED	LIST
336	FFA0, EOF IN FFB1 NOT LISTED	LIST
337	NO RECORDS FOR LISTING FROM FFB0	LIST
338	PERMANENT ERROR ON INPUT FILE FFB0 - NO. FFA1	LIST
339	PERMANENT ERROR ON INPUT FILE FFB0 - KEY FF01	LIST
340	ARCS WARNING FFA0 IN FFB1 AT RECORD FFA2	ARCS
341	ARCS WARNING FFA0 IN FFB1 AT SEQUENCE FFC2	ARCS
342	ARCS ERROR FFA0 IN FFB1 AT RECORD FFA2	ARCS
343	ARCS ERROR FFA0 IN FFB1 AT SEQUENCE FFC2	ARCS
344	END OF ARCS	ARCS
345	ZIP FAILURE DUE TO PROGRAM FILE NOT FOUND FOR FFC0	ARCS
		CO
346	ZIP FAILURE DUE TO INTERPRETER NOT FOUND FOR FFC0	ARCS
		CO
347	ZIP FAILURE DUE TO INSUFFICIENT MEMORY FOR FFC0	ARCS
		CO
348	ZIP FAILURE DUE TO NO USER DISK OR DIRECTORY FULL FOR FFC0	ARCS
		CO
349	ZIP FAILURE DUE TO FULL MIX FOR FFC0	ARCS
		CO
350	ZIP FAILURE DUE TO USER COUNT ERROR FFC0	ARCS
		CO
351	ZIP FAILURE DUE TO DUPLICATE PACK FFC0	ARCS
		CO
352	ZIP FAILURE DUE TO INVALID LOAD REQUEST FFC0	ARCS
		CO
353	ZIP FAILURE DUE TO MCS ALREADY PRESENT FFC0	ARCS
		CO
354	ZIP FAILURE DUE TO DISK ERROR FFC0	ARCS
		CO
355	ZIP FAILURE DUE TO CODE FILE ERROR FFC0	ARCS
		CO

356	ZIP FAILURE DUE TO ILLEGAL DATA COMM LOAD REQUEST FFC0	ARCS CO
357	ZIPPED PROGRAM DS'ED FFC0	ARCS
358	ZIPPED PROGRAM DP'ED FFC0	ARCS
359	ZIP FAILURE, REASON UNKNOWN FFC0	ARCS CO
360	CANNOT PAUSE	ARCS
361	MISSING ARCS SPECIFICATION	ARCS
362	PLEASE RE-ENTER NUMBER IN RANGE 1-65535	ARCS
363	ERROR FFA0 NO ASSIGN SYMBOL AFTER ASSIGN VARIABLE.	ARCS
364	ERROR FFA0 VARIABLE OUT OF SCOPE OF PROCEDURE.	ARCS
365	ERROR FFA0 NO NUMBER ENTERED ON ACCEPT OF NUMERIC VARIABLE	ARCS
366	ERROR FFA0 NON-NUMERIC CHARACTER ENTERED FOR NUMERIC ACCEPT.	ARCS
367	ERROR FFA0 NO IF FOR ENCOUNTERED ELSE.	ARCS
368	ERROR FFA0 PROCEDURE NAME IN CALL STATEMENT NOT VARIABLE OR IDENTIFIER.	ARCS
369	ERROR FFA0 FILE NAME OF PROCEDURE FROM FILE > 12 CHARACTERS.	ARCS
370	ERROR FFA0 CALL FOR UNDECLARED PROCEDURE ENCOUNTERED	ARCS
371	ERROR FFA0 FID AFTER MFID IN CALL FROM CLAUSE > 12 CHARACTERS.	ARCS
372	ERROR FFA0 OVERFLOW CAUSED BY ARITHMETIC EXPRESSION EVALUATION - CHECK RESULTS.	ARCS
373	ERROR FFA0 PARAMETER PASSED IN CALL STATEMENT NOT VARIABLE OR LITERAL.	ARCS
374	ERROR FFA0 INSUFFICIENT DATA SPACE FOR CALL PARAMETER	ARCS
375	ERROR FFA0 FORMAL PARAMETER IN PROCEDURE HEAD NOT AN IDENTIFIER	ARCS
376	ERROR FFA0 UNEQUAL NUMBER OF ACTUAL AND FORMAL PARAMETERS FOR PROCEDURE.	ARCS
377	ERROR FFA0 CALL PROCEDURE FROM FILE NAME NOT STRING OR VARIABLE	ARCS
378	ERROR FFA0 FILE NAME IN CALL PROCEDURE FROM CLAUSE TOO LONG.	ARCS
379	ERROR FFA0 MFID PART OF FILE NAME IN CALL PROCEDURE FROM CLAUSE > 7 CHARACTERS.	ARCS
380	ERROR FFA0 VARIABLE ALREADY DECLARED IN CURRENT SCOPE.	ARCS
381	ERROR FFA0 STRING LENGTH EXPRESSION NOT NUMERIC.	ARCS
382	ERROR FFA0 STRING LENGTH IN DECLARE STATEMENT = 0 OR > 255.	ARCS
383	ERROR FFA0 INSUFFICIENT DATA SPACE FOR DECLARED STRING	ARCS
384	ERROR FFA0 OPERATOR NOT FOLLOWED BY UNARY OPERATOR, LITERAL, VARIABLE OR (.	ARCS

385	ERROR FFA0 NO BEGIN FOR ENCOUNTERED END.	ARCS
386	ERROR FFA0 MISSING LEFT BRACKET AFTER FUNCTION NAME.	ARCS
387	ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME NOT STRING OR VARIABLE.	ARCS
388	ERROR FFA0 UNEXPECTED END OF ARCS INPUT FILE.	ARCS
389	ERROR FFA0 MISSING FID AFTER MFID IN FUNCTION CALL.	ARCS
390	ERROR FFA0 BLANK STRING AS FILE NAME/DISK NAME/TAPE NAME IN FUNCTION CALL.	ARCS
391	ERROR FFA0 CONTROL STACK UNDERFLOW OR OVERFLOW.	ARCS
392	ERROR FFA0 OPERAND STACK UNDERFLOW OR OVERFLOW.	ARCS
393	ERROR FFA0 OPERATOR STACK UNDERFLOW OR OVERFLOW.	ARCS
394	ERROR FFA0 FID AFTER MFID IN FUNCTION CALL TOO LONG	ARCS
395	ERROR FFA0 FILE MFID/TAPE MFID IN FUNCTION CALL TOO LONG.	ARCS
396	ERROR FFA0 NO DO FOR ENCOUNTERED UNTIL.	ARCS
397	ERROR FFA0 MISSING RIGHT BRACKET AFTER FUNCTION PARAMETER.	ARCS
398	ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME PARAMETER NOT STRING TYPE.	ARCS
399	ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME IN FUNCTION CALL TOO LONG.	ARCS
400	ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME IN FUNCTION CALL INVALID.	ARCS
401	ERROR FFA0 LIMIT ON NUMBER OF DECLARED VARIABLES EXCEEDED	ARCS
402	ERROR FFA0 INPUT FILE EITHER NOT PRESENT OR HAS WRONG FILE TYPE.	ARCS
403	ERROR FFA0 HEX NUMBER CONTAINS MORE THAN 4 HEX CHARACTERS.	ARCS
404	ERROR FFA0 NO PROCEDURE HEAD FOR ENCOUNTERED ENDPROC.	ARCS
405	ERROR FFA0 ENDPROC ENCOUNTERED AT LEVEL OTHER THAN THE OUTERMOST.	ARCS
406	ERROR FFA0 FIRST EXPRESSION OPERAND NOT VARIABLE, LITERAL OR FUNCTION.	ARCS
407	ERROR FFA0 UNTIL EXPRESSION NOT NUMERIC.	ARCS
408	ERROR FFA0 INVALID OPERATOR SYMBOL.	ARCS
409	ERROR FFA0 NON-STRING OPERANDS FOR STRING OPERATOR.	ARCS
410	ERROR FFA0 INSUFFICIENT DATA SPACE TO EVALUATE STRING EXPRESSION.	ARCS
411	ERROR FFA0 OPERATOR REQUIRES 2 PARAMETERS, ONLY 1 PRESENT.	ARCS
412	ERROR FFA0 UNARY OPERATOR NOT FOLLOWED BY VARIABLE, LITERAL OR LEFT BRACKET.	ARCS

413.	ERROR FFA0 LEFT BRACKET NOT FOLLOWED BY VARIABLE, LITERAL OR UNARY OPERATOR.	ARCS
414	ERROR FFA0 INVALID ENTRY AFTER OPERAND IN EXPRESSION.	ARCS
415	ERROR FFA0 TOO MANY OPERANDS FOR THE NUMBER OF OPERATORS.	ARCS
416	ERROR FFA0 TOO MANY OPERATORS FOR THE NUMBER OF OPERANDS.	ARCS
417	ERROR FFA0 NO LEFT BRACKET FOR ENCOUNTERED RIGHT BRACKET	ARCS
418	ERROR FFA0 RIGHT BRACKET NOT PRECEDED BY VARIABLE OR LITERAL.	ARCS
419	ERROR FFA0 INVALID OPERATOR ENCOUNTERED.	ARCS
420	ERROR FFA0 RIGHT BRACKET IN EXPR NOT FOLLOWED BY LEFT BRACKET, OPERATOR OR VALID TERMINATOR.	ARCS
421	ERROR FFA0 NO IF FOR ENCOUNTERED FI.	ARCS
422	ERROR FFA0 NAME OF INPUT FILE INVALID.	ARCS
423	ERROR ERROR FFA0 ARCS INPUT FILE NAME TOO LONG.	ARCS
424	ERROR FFA0 UNTIL CLAUSE ENCOUNTERED IN LOOP BODY.	ARCS
425	ERROR FFA0 WAIT EXPRESSION NOT NUMERIC.	ARCS
426	ERROR FFA0 NO DO FOR ENCOUNTERED WHILE.	ARCS
427	ERROR FFA0 ENCOUNTERED A STRING OF LENGTH > 255.	ARCS
428	ERROR FFA0 NO THEN CLAUSE AFTER IF EXPRESSION.	ARCS
429	ERROR FFA0 NO DO FOR ENCOUNTERED OD.	ARCS
430	ERROR FFA0 PROCEDURE NAME NOT AN IDENTIFIER.	ARCS
431	ERROR FFA0 PROCEDURE ALREADY DECLARED WITH THIS NAME.	ARCS
432	ERROR FFA0 END OF FILE ENCOUNTERED WHILE LOOKING FOR ENDPROC.	ARCS
433	ERROR FFA0 NUMBER OF PROCEDURES LIMIT EXCEEDED.	ARCS
434	ERROR FFA0 PROCEDURE NESTING NOT PERMITTED.	ARCS
435	ERROR FFA0 PROCEDURE DECLARATION NOT AT OUTERMOST LEVEL.	ARCS
436	ERROR FFA0 RETURN ENCOUNTERED OUTSIDE OF A PROCEDURE.	ARCS
437	ERROR FFA0 MISSING LEFT BRACKET AFTER RPGSWITCH.	ARCS
438	ERROR FFA0 INDICATOR-SETTING PARAMETER AFTER RPGSWITCH NOT STRING OR VARIABLE.	ARCS
439	ERROR FFA0 MISSING RIGHT BRACKET AT END OF RPGSWITCH STATEMENT.	ARCS
440	ERROR FFA0 FILE NAME IN RPGSWITCH STATEMENT NOT STRING OR VARIABLE.	ARCS
441	ERROR FFA0 FILE NAME IN RPGSWITCH STATEMENT > 12 CHARACTERS.	ARCS

442	ERROR FFA0 VARIABLE AS INDICATOR-SETTING PARAMETER NOT STRING TYPE.	ARCS
443	ERROR FFA0 VARIABLE AS FILE NAME PARAMETER NOT STRING TYPE.	ARCS
444	ERROR FFA0 NON-STANDARD VALUES USED IN SWITCH SETTING.	ARCS
445	ERROR FFA0 UNEXPECTED RESERVED WORD.	ARCS
446	ERROR FFA0 UNEXPECTED SPECIAL CHARACTER.	ARCS
447	ERROR FFA0 WHILE EXPRESSION NOT NUMERIC.	ARCS
448	ERROR FFA0 WHILE ENCOUNTERED IN LOOP BODY.	ARCS
449	ERROR FFA0 ZIP STRING > 255 CHARACTERS.	ARCS
450	ERROR FFA0 DECIMAL LITERAL > 65535.	ARCS
451	ERROR FFA0 NUMBER ENTERED FOR NUMERIC ACCEPT > 65535.	ARCS
452	COMPILER BEING ZIPPED.	CO
453	COMPILER NOT ZIPPED.	CO
454	CANNOT INITIATE THIS CO – MAXIMUM NUMBER OF CO'S ALREADY IN PROCESS.	CO
460	NO FILE FFC0 TO EXECUTE.	CO
461	ZIP FAILURE WHEN ZIPPING FFC0.	CO
462	RPG COMPILER ATTEMPTED TO ZIP RPGCOPY – NOT IMPLEMENTED.	CO
469	JOB TERMINATED.	
472	COMPILER PASS DS'ED FFC0.	CO
473	COMPILER PASS DP'ED FFC0.	CO
474	CO – MIX EMPTY .	CO
475	FEATURE NOT IMPLEMENTED – JOB ABORTED.	CO
476	CO ERROR FFA0 FIRST WORD IN INITIATING MESSAGE IS NOT A VALID DISK OR FILE NAME.	CO
477	CO ERROR FFA0 INVALID OBJECT PROGRAM NAME OR COMPILER NAME SPECIFIED.	CO
478	CO WARNING FFA0 COMPILER PACK ID. TRUNCATED TO 7 CHARACTERS.	CO
479	CO ERROR FFA0 NO COMPILER SPECIFIED.	CO
480	CO WARNING FFA0 PROGRAM PACK ID. TRUNCATED TO 7 CHARACTERS.	CO
481	CO WARNING FFA0 PROGRAM ID. TRUNCATED TO 12 CHARACTERS.	CO
482	CO ERROR FFA0 RESERVED WORD NOT FOUND WHEN EXPECTED.	CO
483	CO ERROR FFA0 FI/FILE NOT FOLLOWED BY PATCHFILE/SOURCEIN/SOURCEOUT/WORKFILE/ PRINTOUT.	CO
484	CO ERROR FFA0 RESERVED WORD USED OUT OF CONTEXT.	CO
485	CO WARNING FFA0 COMPILER OPTION ALREADY MODIFIED – PREVIOUS MOD. IGNORED.	CO
486	CO WARNING FFA0 PRINT OPTION SPECIFIED MORE THAN ONCE.	CO
487	CO WARNING FFA0 RECORD SIZE*BLOCKING FACTOR>65535. BLOCK SIZE SET EQUAL TO RECORD SIZE.	CO

488	CO WARNING FFA0 MESSAGE NOT USED BY COBOL - MESSAGE TEXT IGNORED.	CO
489	CO WARNING FFA0 MESSAGE TEXT ALREADY PROCESSED - THIS MESSAGE TEXT IGNORED.	CO
490	CO WARNING FFA0 MODIFICATION NOT VALID FOR FF01 - IGNORED.	CO
491	CO ERROR FFA0 COMPILER NOT ZIPPED - ERRORS IN INITIATING MESSAGE.	CO
492	CO ERROR FFA0 \$SEGMENT PARAMETER NOT A VALID FILE NAME.	CO
493	CO ERROR FFA0 CO.MASTER NO LONGER ON DISK - JOB TERMINATED.	CO
494	CO WARNING FFA0 \$SEGMENT PACK-ID TRUNCATED TO 7 CHARACTERS.	CO
495	CO WARNING FFA0 \$SEGMENT PROGRAM NAME TRUNCATED TO 12 CHARACTERS.	CO
496	CO ERROR FFA0 \$SEGSIZE PARAMETER NOT NUMERIC.	CO
497	CO WARNING FFA0 FF01 MODIFICATION NOT VALID FOR THIS COMPILER - IGNORED.	CO
498	CO WARNING FFA0 MESSAGE CONTAINS NO VALID DOLLAR CARDS.	CO
499	CO WARNING FFA0 \$CARD NOT VALID FOR THIS COMPILER.	CO
500	CO WARNING FFA0 INVALID \$CARD IN MESSAGE TEXT.	CO
501	CO WARNING FFA0 FF01 MODIFICATION(S) ALREADY DONE - THESE MODS. IGNORED	CO
502	CO WARNING FFA0 FF01 MODIFICATIONS INCOMPLETE.	CO
503	DISK FFC0 LOCKED	ICMD BUILDTRANS CHECKADUMP COPY DUMPADISK LD LIST PL SYS-SUPERUTL TL TRANSLATOR COMPARE CHECK.DISK LR XD KA CREATE AMEND UPDATE FL KEY.CHECK LOAD.VFU DSKUTL

		SAU.INIT
		DA
504	CO WARNING FFA0 NO FF01 MODIFICATIONS.	CO
505	CO WARNING FFA0 MFID/FID MODIFICATION ALREADY DONE FOR THIS FILE – MOD. IGNORED.	CO
506	CO ERROR FFA0 PACK NAME/FILE NAME IN NAME SPECIFICATION INVALID.	CO
507	CO WARNING FFA0 PACK NAME IN NAME SPECIFICATION TRUNCATED TO 7 CHARACTERS.	CO
508	CO WARNING FFA0 FILE NAME IN NAME SPECIFICATION TRUNCATED TO 12 CHARACTERS.	CO
509	CO ERROR FFA0 PACK NAME IN MFID CLAUSE INVALID.	CO
510	CO WARNING FFA0 PACK NAME IN MFID CLAUSE TRUNCATED TO 7 CHARACTERS.	CO
511	CO ERROR FFA0 FILE NAME IN FID CLAUSE INVALID.	CO
512	CO WARNING FFA0 FILE NAME IN FID CLAUSE TRUNCATED TO 12 CHARACTERS.	CO
513	CO WARNING FFA0 RECORD SIZE OF ZERO SPECIFIED – IGNORED – DEFAULT USED.	CO
514	CO WARNING FFA0 DEVICE MODIFICATION ALREADY DONE FOR THIS FILE – THIS MOD IGNORED.	CO
515	CO WARNING FFA0 SPECIFIED DEVICE TYPE INVALID – DEFAULT DEVICE TYPE USED.	CO
516	CO WARNING FFA0 FILE SIZE ALREADY SPECIFIED FOR THIS FILE – THIS MOD. IGNORED.	CO
517	CO ERROR FFA0 FILESIZE PARAMETER NOT NUMERIC.	CO
518	CO WARNING FFA0 BLOCK SIZE ALREADY MODIFIED – THIS MOD. IGNORED.	CO
519	CO ERROR FFA0 RECORDS. BLOCK/BLOCK PARAMETER NOT NUMERIC.	CO
520	CO ERROR FFA0 RECORD PARAMETER NOT NUMERIC.	CO
521	WARNING – DISK FFC0 NOT AVAILABLE	COPY
522	WARNING – FILE LIST FOR FFB0 MAY BE INCOMPLETE	COPY
523	PLEASE RUN SQ ON FFC0	LR XD
524	NO CONSISTENT ERRORS	CHECK.DISK
525	INCONSISTENT ERROR(S) NOTIFIED – POSSIBLE MEDIUM/DRIVE FAULT	CHECK.DISK
526	FIRST DIFFERENCE FOUND BETWEEN FILES –	COMPARE
531	CO ERROR FFA0 NULL INITIATING MESSAGE – NO FILE CO.STARTUP – JOB TERMINATED.	CO
532	CO ERROR FFA0 NESTED MACRO CALL FOUND.	CO
533	CO ERROR FFA0 INVALID FILE NAME IN MACRO CALL – FFC1.	CO
534	CO WARNING FFA0 PACK NAME IN MACRO CALL TRUNCATED TO 7 CHARACTERS.	CO

535	CO WARNING FFA0 FILE NAME IN MACRO CALL TRUNCATED TO 12 CHARACTERS.	CO
536	CO WARNING FFA0 RECORD SIZE FOR CARD DEVICE > 96. RECORD AND BLOCK SIZE SET TO 96.	CO
537	CO ERROR FFA0 FILE CO.STARTUP EMPTY – NO INITIATING MESSAGE – JOB TERMINATED.	CO
539	CO ERROR FFA0 CO.MASTER NOT FOUND – JOB TERMINATED	CO
542	CO ERROR FFA0 NO CO'S TO RESTART/CLEAR – JOB TERMINATED.	CO
544	CO ERROR FFA0 SYSTEM.STATUS COMMUNICATE FAILURE – CANNOT RUN CO ON THIS SYSTEM	CO
545	FFB0 RECORD. FF01 AT BYTE. FF02	COMPARE
546	CO WARNING FFA0 WORKFILE NOT FOLLOWED BY MFID – MOD. IGNORED.	CO
547	CO ERROR FFA0 SPECIFIED RESTART LINE-NUMBER FOR LISTING NOT NUMERIC.	CO
548	CO WARNING FFA0 SPECIFIED RESTART LINE-NUMBER > 65535. LINE-NUMBER SET TO ZERO.	CO
550	CO ERROR FFA0 NO PARAMETER SPECIFIED FOR CLEAR FUNCTION.	CO
551	CO WARNING FFA0 LINE-NUMBER PARAMETER VALID ONLY FOR COBOL – IGNORED.	CO
552	CO ERROR FFA0 ILLEGAL CHARACTER ENCOUNTERED IN INITIATING MESSAGE.	CO
553	CO ERROR FFA0 CANNOT CONTINUE COMPILATION – COMPILATION BLOCK INITIALIZED.	CO
554	CO WARNING FFA0 NON-INPUT DEVICE SPECIFIED FOR FF01 – IGNORED.	CO
555	WARNING – GENERATION NO. MISMATCH BETWEEN FFB0 AND DATAFILE FFB1	KEY.CHECK
556	CO ERROR FFA0 MACRO FILE NOT FOUND – FFC1.	CO
557	CO WARNING FFA0 NON-OUTPUT DEVICE SPECIFIED FOR SOURCEOUT – IGNORED.	CO
558	CO WARNING FFA0 BLOCK SIZE FOR SOURCEOUT IS NOT A MULTIPLE OF THE RECORD SIZE.	CO
559	CO WARNING FFA0 SPECIFIED BLOCK SIZE FOR TAPE DEVICE TOO LARGE – DEFAULT USED.	CO
560	CO WARNING FFA0 SPECIFIED RECORD SIZE FOR TAPE DEVICE TOO LARGE – DEFAULT USED.	CO
561	CO WARNING FFA0 80 COL CARD DEVICE RECORD SIZE > 80. RECORD AND BLOCK SIZE SET TO 80.	CO
562	CO WARNING FFA0 96 COL CARD DEVICE RECORD SIZE > 96. RECORD AND BLOCK SIZE SET TO 96.	CO
563	FIRST DISCREPANCY FOUND BETWEEN KEYFILE FFB0 AND DATAFILE FFB1	KEY.CHECK
564	CO WARNING FFA0 DEVICE SPECIFIED FOR PRINTOUT NOT PRINTER TYPE – IGNORED.	CO
565	CO WARNING FFA0 PRINTOUT NOT FOLLOWED BY DEVICE OR NO.BUFFERS – MOD. IGNORED.	CO
566	CO WARNING FFA0 BLOCK SIZE OR BLOCKING-FACTOR OF ZERO SPECIFIED – DEFAULT USED.	CO

567	CO WARNING FFA0 SOURCEOUT FILESIZE SPECIFIED AS ZERO - IGNORED - DEFAULT USED.	CO
568	CO WARNING FFA0 SPECIFIED FF01 FILESIZE > 65535 - DEFAULT USED.	CO
569	FFC0 DISCREPANCY FOUND WITH KEYFILE FFB1	KEY.CHECK
570	CO WARNING FFA0 SPECIFIED RECORD SIZE > 65535 - DEFAULT USED.	CO
571	CO WARNING FFA0 SPECIFIED BLOCK SIZE OR BLOCKING-FACTOR > 65535 - DEFAULT USED.	CO
572	CO ERROR FFA0 WHEN COMPILING MPL ON CUMBERNAULD SUPPORT SYSTEM, LEVEL NO. REQUIRED.	CO
573	CO WARNING FFA0 IF 80 COL CARD OUTPUT DEVICE USED, RECORD TRUNCATION WILL OCCUR.	CO
574	CO WARNING FFA0 RECORD SIZE ALREADY MODIFIED - THIS MOD. IGNORED.	CO
575	CO ERROR FFA0 MACRO FILE PRESENT BUT EMPTY - FFC1.	CO
576	FFC0 DISCREPANCY FOUND WITH DATAFILE FFB1.	KEY.CHECK
577	CO ERROR FFA0 NO.BUFFERS PARAMETER NOT NUMERIC.	CO
578	CO WARNING FFA0 VALUE SPECIFIED FOR NO.BUFFERS NOT IN RANGE 1 TO 16 - DEFAULT USED.	CO
579	CO WARNING FFA0 NO.BUFFERS ALREADY MODIFIED - THIS MOD. IGNORED.	CO
580	KEY.CHECK ON FFB0 COMPLETED	KEY.CHECK
581	KEY.CHECK COMMENCING ON DATAFILE FFB0	KEY.CHECK
583	VOLUME FFC0 OF FFC1 OUT OF SEQUENCE	ICMD
584	DUPLICATE FILE FFB0	ICMD
585	NO FREE LABEL ON FFC0	ICMD
586	NO SPACE FOUND ON FFC0	ICMD
587	RECORD SIZE OF FFB0 EXCEEDS FFA1	ICMD
588	NEXT SEQUENCE NUMBER OF FFC0 EXCEEDS FFA1	ICMD
589	BAD INDEX TRACK ON FFC0 SECTOR FFA1	ICMD
590	INPUT ERROR ON FFB0 SECTOR FFA1	ICMD
591	OUTPUT ERROR ON FFB0 SECTOR FFA1	ICMD
592	ENTER VOL-ID FOR FFC0 FFC1	CO
593	FFC0 PURGED	ICMD
594	FILESIZE REQUIRED FOR FFB0	ICMD
595	FFB0 FILE IDENTIFIER TOO SHORT	COMPARE
596	ILLEGAL KEY FOR FFB0 : FFA1-FFA2	COMPARE
597	ILLEGAL RECORDS FOR FFB0 : FFA1	COMPARE
598	CANNOT COMPARE PAST POSITION FFA0 IN FFB1	COMPARE
599	CANNOT READ RECORD FFA0 OF FFB1	COMPARE
600	FFB0 EXHAUSTED AT FFA1	COMPARE
601	CANNOT REALIGN FFB0 WITH FFB1 - FFA2 ERRORS	COMPARE
602	END OF FILE FFB0 BEFORE FFB1 - FFA2 ERRORS	COMPARE
603	FFB0 WITH FFB1 COMPARED - FFA2 ERRORS	COMPARE
604	INSUFFICIENT DISK SPACE TO CONSOLIDATE LOG FILES	SYS-SUPERUTL
		TL

605	ENTER A FOUR CHARACTER DUMP IDENTIFICATION PREFIX	DUMPADISK
606	SCREEN SIZE IS TOO SMALL TO RUN UTILITY	PL
607	ENTER FF00 TO FINISH OR FF01 TO CONTINUE	BUILDTRANS PL TRANSLATOR
608	KEYFILE FFB0 IS CONSISTENT WITH DATAFILE FFB1	KEY.CHECK
609	GENERATION NO. OF KEYFILE FFB0 MODIFIED TO FFC1	KEY.CHECK
610	UNABLE TO MOFIFY GENERATION NO. OF KEYFILE FFB0	KEY.CHECK
611	KEYFILE FFB0 IS NOT CONSISTENT WITH DATAFILE FFB1	KEY.CHECK
612	SYSTEM DISK IS LOCKED.	ICMD BUILDTRANS CHECKADUMP DUMPADISK LD SYS-SUPERUTL TL TRANSLATOR WL COMPARE CHECK.DISK LR KA CREATE AMEND UPDATE PPID FL LOAD.VFU DSKUTL DA
613	FFA0 RECORDS WRITTEN TO BACKUP FILE FFB1	PL
614	READ ERROR ENCOUNTERED ON FFC0. RANGE FFA1 - FFA2	DUMPADISK
615	OPEN DISK UNLABELLED NOT IMPLEMENTED	DUMPADISK DSKUTL RL
616	DISK LABEL OF FFC0 NOT IN CMS FORMAT	DUMPADISK DSKUTL RL
617	KEYFILE FFB0 PERTAINS TO DATAFILE FFB1	KEY.CHECK
618	WRITE ERROR ENCOUNTERED ON FFC0. RANGE FFA1 - FFA2	DUMPADISK
619	FFB0 NOT DUMPED DUE TO I/O ERROR	DUMPADISK
620	DOES THE SYSTEM HAVE A LOCAL SPO? - <Y OR N>	CONFIGURER
621	DOES THE SYSTEM HAVE A REMOTE SPO? - <Y OR N>	CONFIGURER
622	FFB0 NOT LOADED DUE TO I/O ERROR	DUMPADISK

623	ENTER FILE-ID AND INITIATING MESSAGE OF REMOTE SPO MCS	CONFIGURER
624	WARNING - DISK FFC0 LOCKED	COPY
625	WARNING - FILE LIST MAY BE INCOMPLETE	COPY
626	READY AND RESERVE REQUIRED DRIVE FOR DISK REFORMAT	DSKUTL
627	CANNOT READ SECTOR 0 OF FFC0	DSKUTL RL
628	FFC0 IS CMS DISK FFC1 - ENTER OK TO CONFIRM <RF> REQUIRED	DSKUTL
629	DSKUTL <RF> OF FFC0 NOT CONFIRMED	DSKUTL
630	FFC0 IS NOT A FIXED DISK	DSKUTL
631	FFC0 WILL BE REFORMATTED AS PHYSICAL UNIT FFC1	DSKUTL
632	TRACK 0 OF FFC0 BAD	DSKUTL
633	BAD MTR SECTOR FFC0 ON FFC1	DSKUTL
634	BOOTSTRAP FILE FFB0 NOT CMS STANDARD	DSKUTL SAU.INIT
635	READ ERROR ON OLD DIRECTORY OF FFC0 AT SECTOR FFC1	DSKUTL
636	OLD DIRECTORY OF FFC0 IS CORRUPT	DSKUTL
637	TOO MANY BAD ENTRIES IN THE OLD AVAILABLE TABLE OF FFC0	DSKUTL
638	CANNOT ACCOMMODATE DIRECTORY FOR FFC0	DSKUTL
639	AVAILABLE TABLE TOO SMALL FOR FFC0	DSKUTL
640	READ ERROR ON BOOTSTRAP FILE FFB0	DSKUTL SAU.INIT
641	DSKUTL <RF> ABORTED	DSKUTL
642	FFC0 BOOTSTRAP VERSION FFC1 USED ON FFC2 DISK FFC3	DSKUTL
643	FFC0 SECTOR(S) ARE NOT AVAILABLE ON FFC1 DISK FFC2	DSKUTL
644	PLEASE RUN CHECK.DISK ON FFC0 DISK FFC1	DSKUTL
645	DSKUTL <RF> OF FFC0 DISK <FFC1> AS FFC2 ON FFC3 COMPLETED	DSKUTL
646	DSKUTL <LIST> ABORTED	DSKUTL
647	ILLEGAL SECTOR ADDRESS FFC0 FOR FFC1	DSKUTL
648	FFC0 EXHAUSTED DURING SECTOR RANGE FFC1 - FFC2	DSKUTL
649	PRESENT INPUT DISK FIRST PLEASE	DSKUTL
650	PRESENT OUTPUT DISK NOW PLEASE	DSKUTL
651	FFC0 AND FFC1 HAVE DIFFERENT CAPACITY	DSKUTL
652	DSKUTL <COPY> ABORTED	DSKUTL
653	READ ERROR ENCOUNTERED ON FFC0 AT SECTOR FFC1	DSKUTL
654	WRITE ERROR ENCOUNTERED ON FFC0 AT SECTOR FFC1	DSKUTL
655	DSKUTL <COPY> FROM FFC0 TO FFC1 COMPLETED	DSKUTL
656	CANNOT OPEN FILE FFB0 - DIRECTORY FULL	TL CREATE UPDATE
657	SPECIFIED DISK-ID FFC0 FOR <COPY> IGNORED - INPUT FFC1 DISK FFC2 HAS PPIT	DSKUTL

658	ILLEGAL SECTOR RANGE FROM FFC0	DSKUTL
659	READING ALL FFA0 SECTORS ON FFC1.	SCR
660	READ ERROR ON FFC0, SECTOR FFD1.	SCR
661	ENTER DISK ID TO CONTINUE.	SCR
662	ENTER SPACE TO TERMINATE DCR	SCR
663	READING CYLINDERS ABORTED.	SCR
664	BUFFERED BY SECTOR.	SCR
665	MULTI-SECTOR BUFFERED.	SCR
666	READING ALL SECTORS ABORTED.	SCR
667	READING ALL SECTORS COMPLETED.	SCR
668	FFA0 BAD SECTORS READ ON FFC1.	SCR
669	SECOND DISK TO BE TESTED? <Y OR N>	SCR
670	ERROR ON FFC0 MULTI-SECTOR READ.	SCR
671	FFA0 SECTORS READ.	SCR
672	READ ERROR ON FFC0 AVAILABLE TABLE.	SCR
673	DCR TERMINATED BY USER.	SCR
674	NON-NUMERIC INPUT CHARACTER.	SCR
675	MULTI-SECTOR BUFFERING NOT AVAILABLE.	SCR
676	MULTI-SECTOR BUFFERING WANTED? <Y OR N>	SCR
677	FFC0 AVAIL.TABLE WILL NOT BE CHECKED.	SCR
678	MORE THAN 60 BAD AREAS IN AVAILABLE TABLE.	SCR
679	NOT LISTED AS BAD IN AVAIL.TABLE.	SCR
680	LISTED AS BAD IN AVAIL.TABLE.	SCR
681	CAPACITIES ON 2ND DISK NOT SAME AS ON 1ST.	SCR
682	NO ERROR ON MULTI-SECTOR READ.	SCR
683	ERROR READING FDNL ON FFC0.	SCR
684	OPEN.FILES TRUNCATED.	SCR
685	FFA0 OPENS TRIED ON FFC1.	SCR
686	FFC0 ON FFC1 DID NOT OPEN.	SCR
687	FFC0 ON FFC1 OPENED.	SCR
688	FFA0 SUCCESSFUL, FFA1 UNSUCCESSFUL.	SCR
689	OPENING UP TO 20 FILES ON FFC0.	SCR
690	READING FFA0 CYLINDERS ON FFC1.	SCR
691	READING CYLINDERS COMPLETED.	SCR
692	ENTER TEST NUMBERS OR ENTER HELP.	SCR
693	ENTER ON 1 LINE NUMBERS FOR TEST WANTED.	SCR
694	1 = READ.ALL	SCR
695	2 = OPEN.FILES	SCR
696	3 = READ.CYLINDERS	SCR
697	4 = WRITE.INDEX	SCR
698	5 = READ.RANDOM	SCR
699	ALL = ALL TESTS WANTED	SCR
700	SPACE = TERMINATE DCR	SCR
701	NO FILES ON FFC0.	SCR
702	BUT NO ERRORS WHEN SECTORS READ.	SCR
703	WRITING INDEX FILE ON FFC0.	SCR
704	WRITING INDEX FILE FINISHED.	SCR
705	READING FFA0 RANDOM SECTORS ON FFC1.	SCR
706	READING RANDOM SECTORS COMPLETED.	SCR
708	ENTER NUMBER OF RANDOM SECTORS TO BE READ FROM EACH DISK.	SCR
709	INPUT NUMBER > 16,777,215.	SCR
710	ENTER NUMBER OF RECORDS IN INDEX FILE.	SCR

711	DCR FINISHED.	SCR
712	OR SPACE TO CANCEL TEST.	SCR
713	REQUESTED IX FILE SIZE EXCEEDS DISK CAPACITY	SCR
714	FETCH.VALUE = FFD0.	SCR
		SCR.MPL
715	OPEN FAILED ON FFC0.	SCR
716	WRITE.INDEX ABORTED.	SCR
717	DATA SPACE NOT AVAILABLE ON FFC0.	SAU.INIT
		SCR
718	INDEX FILE TRUNCATED.	SCR
719	READING RANDOM SECTORS ABORTED.	SCR
720	DATA MISMATCH ON FFC0, RECORD FFD1.	SCR
721	READING INDEX COMPLETED.	SCR
722	READING INDEX ABORTED.	SCR
723	FFA0 DATA MISMATCHES ON FFC1.	SCR
724	WRITE ERROR ON FFC0, KEY = FF01	SCR
726	PARAMETER SPECIFICATION FOR <RF> INCOMPLETE	DSKUTL
727	NO UNIT SLOTS AVAILABLE IN SYSTEM PPIT	DSKUTL
728	DSKUTL <COPY> FROM FFC0 DISK FFC1 TO FFC2 DISK FFC 3 COMPLETED	DSKUTL
729	PRESENT DISK FOR LISTING PLEASE	DSKUTL
730	SPECIFIED DISK-ID FFC0 FOR <COPY> IGNORED - INPUT FFC1 DISK NOT IN CMS FORMAT	DSKUTL
731	NO PPIT ON DISK FFC0	RL
		PPID
		PPID
732	DISKS IN PPIT OF DISK FFC0 -	PPID
733	LABEL OF FFC0 DISK IS FFC1 - NOT FFC2	RL
734	FFC0 IS FIXED DISK FFC1 WITH PPIT	RL
735	FFC0 DISK FFC1 RELABELLED AS FFC2	RL
736	FFC0 IS CMS DISK FFC1 - CONFIRM <COPY>	DSKUTL
737	READ ERROR ON FFC0, KEY = FF01.	SCR
738	DSKUTL <COPY> TO FFC0 NOT CONFIRMED	DSKUTL
739	P1 COMPUTATION ERROR. I = FFA0	SCR.MPL
740	SYSTEM DISK NOT AVAILABLE.	SCR.MPL
741	SCR.MPL COPY FFC0 TERMINATED.	SCR.MPL
742	WRITE CONDITIONAL FAILS = FFA0, WRITE ERRORS = FFA1	SCR.MPL
743	READ CONDITIONAL FAILS = FFA0, READ ERRORS = FFA1.	SCR.MPL
744	INPUT/OUTPUT MISMATCH, I = FFA0	SCR.MPL
745	SCR.MPL COPY FFC0.	SCR.MPL
746	SCR TERMINATED BY USER.	SCR
747	INVALID INPUT CHARACTER.	SCR
748	1 = DISK CONFIDENCE ROUTINE (DCR)	SCR
749	2 = CPU/IO INTERACTION TEST (CPU.IO)	SCR
750	3 = MAGNETIC DEVICE EXERCISER (MT.IO)	SCR
751	4 = NON.MAGNETIC PERIPHERAL EXERCISER (OTHER.IO)	SCR
752	ENTER NUMBER OF ITERATIONS FOR CPU.IO - DEFAULT = 1.	SCR.MPL
753	TERMINATED. UNABLE TO WRITE 1ST RECORD.	SCR.MPL
754	LINE CONTROL = LINE(1).	SCR

755	ENTER NUMBER OF MT DEVICES – DEFAULT = 1.	SCR
756	ENTER NUMBER OF CS DEVICES – DEFAULT = 0.	SCR
757	ENTER NUMBER OF RECORDS IN FILE – DEFAULT = 256.	SCR
758	ENTER NUMBER OF MT.IO ITERATIONS – DEFAULT = 1.	SCR
759	SPECIFY MT.IO PARAMETERS – ENTER SPACE FOR DEFAULT VALUE.	SCR
760	ERROR: MORE THAN 5 DEVICES REQUESTED.	SCR
761	FILE FFA0 DID NOT OPEN FOR OUTPUT ON FFC1	SCR
762	FILE FFA0 DID NOT OPEN FOR INPUT ON FFC1	SCR
763	WRITE FAIL ON FFC0 FILE FFB1, RECORD FFA2	SCR
764	READ FAIL ON FFC0 FILE FFB1, RECORD FFA2	SCR
765	WRITE/READ MISMATCH ONE FFC0 FILE FFB1, RECORD FFA2	SCR
766	SCR MT.IO; ITERATION FFA0:	SCR
767	READY SCRATCH TAPES ON FFA0 MAGNETIC TAPE DRIVES, FFA1 CASSETTE DRIVES.	SCR
768	CAPACITY OF FFC0 DISK NOT RECOGNIZED – FFC1 SECTORS	DSKUTL
769	*** SQUASH VERSION FFC0 ***	SQ
770	NON-FILE DIRECTORY FULL – “PARTIAL” SQUASH REQUIRED	SQ
771	DISK INTEGRITY SUSPECT	SQ
772	USAGE MAP OF THE DISK WILL BE PRINTED	SQ
773	SQ ABORTED – I/O ERROR AT DISK ADDRESS FFD0	SQ
774	SQ INVALID – NO INITIATING MESSAGE	SQ
775	SQ INVALID – INVALID PACK-ID	SQ
776	SQ INVALID – INVALID SYNTAX	SQ
777	FAST OPTION – INTEGER MUST NOT BE GREATER THAN 65535	SQ
778	SQ INVALID – INVALID ADDRESS	SQ
779	OVERLAP – NO WAY TO SEPARATE THEM, SAVE AND/OR PURGE AND AT LEAST RERUN SQ VERIFY	SQ
780	OVERLAP – THERE IS NO WAY TO SEPARATE THEM, SAVE ONE OR BOTH, PURGE AND AT LEAST RERUN SQ VERIFY	SQ
781	SQ ABORTED – REQUESTED AREA ALREADY EXISTS	SQ
782	SQ ABORTED – INVALID DISK ALLOC.UNIT = 0	SQ
783	IT MAY BE CORRUPTED	SQ
784	SQ INVALID – SPECIFIED DISK NOT AVAILABLE	SQ
785	SQ NEEDS A PRINTER-DO YOU WANT TO CONTINUE WITH NO PRINTOUT <Y OR N>	SQ
786	SQ ABORTED – NOTHING TO SQUASH IN THAT AREA	SQ
787	IRRECOVERABLE ERROR ON DISK – SAVE AND/OR REINITIALIZE	SQ
788	SQ ABORTED – NO WAY TO GET REQUIRED AREA	SQ
789	DIRECTORY DESTROYED – REINITIALIZE THE DISK	SQ
790	ADDRESS MISMATCH – SAVE AND REINITIALIZE THE DISK	SQ

791	TOO MANY DATA AREAS ON DISK - "PARTIAL"	SQ
	SQ REQUIRED	
792	MEMORY INCONSISTENCY OR SOME OTHER	SQ
	IRRECOVERABLE PROBLEMS - RERUN SQ	
793	TOO MANY FILES OPEN OR/AND BAD AREAS - NO	SQ
	WAY TO SQUASH THIS DISK	
794	CANNOT SQUASH TOO FRAGMENTED DISK -	SQ
	PROGRAM LIMITATION	
795	SQ ABORTED - SPECIFIED PACK-ID IS NOT	SQ
	PHYSICAL UNIT	
796	SQ ABORTED - SPECIFIED PACK-ID NOT FOUND IN	SQ
	PPIT	
797	I/O ERROR ON FFC0	SQ
798	THIS AREA SKIPPED - EXECUTION CONTINUES	SQ
799	BETWEEN DISK ADDRESSES FFD0 TO FFD1	SQ
800	READ ERROR WHEN MOVING FILE AREA	SQ
801	WRITE ERROR IN NEW LOCATION WHEN MOVING	SQ
	FILE AREA	
802	LAST SQ-EXECUTION WAS ABNORMALLY	SQ
	TERMINATED WHEN MOVING FILE - FFC0	
803	INTEGRITY OF FILE SUSPECT - EXECUTION	SQ
	CONTINUES	
804	AREAS STILL MISSING - RERUN SQ VERIFY	SQ
807	BAD SECTORS AND FFC0	SQ
808	FFC0 AND BAD SECTORS	SQ
809	LARGEST AVAILABLE SPACE IS FFA0 SECTORS	SQ
810	TOTAL AVAILABLE SPACE IS FFA0 SECTORS IN	SQ
	FFA1 AREA(S)	
811	*** SQ COMPLETED ***	SQ
812	*** VALIDATION PHASE BEGINS ***	SQ
813	*** SQUASH PHASE BEGINS ***	SQ
814	*** SYCOPY VERSION FFC0 ***	SYCOPY
815	*** INITIATING MESSAGE FOR SYCOPY ***	SYCOPY
816	INVALID CASE - SYCOPY ABORTED	SYCOPY
817	FFA0 TAPE(S) COPIED	SYCOPY
818	FFA0 TAPE(S) COMPARED	SYCOPY
819	FFA0 TAPE(S) COPIED AND COMPARED	SYCOPY
820	FFA0 TAPE(S) MERGED	SYCOPY
821	NO ERROR	SYCOPY
822	FFA0 TAPE(S) DELETED	SYCOPY
823	SYCOPY ABORTED	SYCOPY
824	INITIATING MESSAGE INCOMPLETE	SYCOPY
825	DISCREPANCY IN DUMP LEVELS	SYCOPY
826	END OF FILE	SYCOPY
827	NOT END OF FILE	SYCOPY
828	FFC0 AND FFC1	SQ
829	FFB0 I/O ERROR	PB
830	INVALID FILENAME	PB
831	INVALID PACK-ID	PB
832	FIELD TOO LARGE	PB
833	NUMERIC FIELD NOT NUMERIC	PB
834	FFB0 EOF ENCOUNTERED BEFORE FFA1 PAGE(S)	PB
	SKIPPED	

835	FFB0 NOT SUCCESSFULLY OPENED	PB
836	FFB0 FILE TYPE NOT PB	PB
837	FFB0 HEADER RECORD NOT ASCII	PB
838	FFB0 INVALID BLOCK SIZE	PB
839	FFB0 INVALID RECORD SIZE	PB
840	FFB0 HEADER RECORD PB NUMBER GREATER THAN 65535	PB
841	COPIES PARAMETER EQUALS ZERO	PB
842	DUPLICATE OPTION	PB
843	PAGE PARAMETER EQUALS ZERO	PB
844	RECORD PARAMETER EQUALS ZERO	PB
845	TOO MANY START-LENGTH PAIRS SPECIFIED	PB
846	ERROR FFA0 PROCEDURE CALL ENCOUNTERED WITHIN A PROCEDURE IN A SECONDARY INPUT FILE	ARCS
847	PB NUMBER GREATER THAN 65535	PB
848	INVALID EQUALS PARAMETER	PB
849	INVALID AUTO PARAMETER	PB
850	AUTO OPTION REQUIRES CLOCK ON SYSTEM	PB
851	INVALID ON PARAMETER	PB
852	INVALID PB RECORD FORMAT	PB
853	PARAMETER MUST BE NUMERIC	PB
854	SPECIAL FORMS REQUIRED FOR FFB0	PB
855	CREATED AS FFC0-FFC1 BY FFC2	PB
856	AX INVALID - REQUIRES Y OR N	PB
857	AX INVALID - REQUIRES OK	PB
858	INVALID OPTION	PB
859	INITIATING MESSAGE TOO LONG	PB
860	REMOVE FORMS	PB
861	INVALID PB PARAMETER	PB
862	INVALID LIST PARAMETER	PB
863	NOT A BASIC ECMA TAPE	ECMA.LD
864	UNEXPECTED END OF TAPE	ECMA.LD
865	NOT A COMPACT ECMA TAPE	ECMA.LD
866	INCORRECT TAPE NAME	ECMA.LD
867	NOT FIRST PART OF FILE	ECMA.LD
868	DATA BLOCK TOO LARGE - FFB0	ECMA.LD
869	SCR FINISHED.	SCR
870	SCR OTHER.IO: ITERATION FFA0, WRITE PRINTER	SCR
871	LINE CONTROL = PAGE.	SCR
872	LINE CONTROL = LINE(1) BEFORE.	SCR
873	LINE CONTROL = LINE(1) AFTER.	SCR
874	ENTER NUMBER OF OTHER.IO ITERATIONS - DEFAULT = 1.	SCR
875	PRINTER TO BE TESTED? <Y OR N>	SCR
876	CARD READER/PUNCH TO BE TESTED? <Y OR N>	SCR
877	PRINTER TEST FINISHED.	SCR
878	PRINTER TEST ABORTED.	SCR
879	ENTER SPACE FOR DEFAULT VALUES OR 0 TO TERMINATE SCR	SCR
880	OTHER.IO FINISHED.	SCR
881	CPU.IO FINISHED.	SCR
882	MT.IO FINISHED.	SCR

883	SECOND DISK NOT WANTED.	SCR
884	REPLY NOT "Y" OR "N".	SCR
885	MULTI-SECTOR BUFFERING NOT WANTED.	SCR
886	FF00/SORTINTRINS [178] SORT-MERGE OUTPUT FILE NOT CREATED	SORTINTRINS
887	FF00/SORTINTRINS [188] INITIATING MESSAGE INVALID	SORTINTRINS
888	FF00/SORTINTRINS [189] SORT-MERGE VER FFC1 INITIATED FROM FF02/FFC3	SORTINTRINS
889	FF00/SORTINTRINS [182] NO INITIATING MESSAGE	SORTINTRINS
890	FF00/SORTINTRINS [193] INPUT RECORD SIZES UNEQUAL - BAD FILE FFC1	SORTINTRINS
891	FF00/SORTINTRINS [194] IN/OUT RECORD SIZES UNEQUAL - OUTPUT SIZE CHANGED	SORTINTRINS
892	FF00/SORTINTRINS [170] DUPLICATE RECORD FFA1	SORTINTRINS
893	FF00/SORTINTRINS [184] FILE ERROR FFA1 NEAR RECORD FFA2 ON ON FFC3	SORTINTRINS
894	FF00/SORTINTRINS [180] SORT-MERGE SOFTWARE ERROR	SORTINTRINS
895	FF00/SORTINTRINS [174] FFA1 RECORDS CONTAINING INVALID INDEX KEYS	SORTINTRINS
896	FF00/SORTINTRINS [444] KEYBUILD NOT FOUND	SORTINTRINS
897	FF00/SORTINTRINS [446] ALL INPUT RECORDS DELETED	SORTINTRINS
898	FF00/SORTINTRINS [173] FFA1 DUPLICATE RECORDS	SORTINTRINS
899	FF00/SORTINTRINS [187] DUPLICATE RECORDS - KEYFILE NOT BUILT	SORTINTRINS
900	FF00/SORTINTRINS [196] KEY OVER-RUNS RECORD END	SORTINTRINS
901	FF00/SORTINTRINS [185] UNORDERED MERGE INPUT FILE FFC1 NEAR RECORD FFA2	SORTINTRINS
902	FF00/SORTINTRINS [176] FFA1 RECORDS MERGED	SORTINTRINS
903	FF00/SORTINTRINS [177] FFA1 FILES MERGED	SORTINTRINS
904	FF00/SORTINTRINS [183] FFA1 RECORDS SORTED	SORTINTRINS
905	FF00/SORTINTRINS [181] FFA1 RECORDS REFERENCED BY KEYFILE/TAGFILE	SORTINTRINS
906	FF00/SORTINTRINS [175] FFA1 DELETED RECORDS	SORTINTRINS
907	FF00/SORTINTRINS [198] FFA1 PARITY BLOCKS	SORTINTRINS
908	FF00/SORTINTRINS [445] MERGER NOT FOUND	SORTINTRINS
909	FFB0 HEADER RECORD DEVICE TYPE INVALID	SORTINTRINS
910	FF00 VERSION FF01	ARCS BUILDTRANS
911	FF00/SORTINTRINS [171] ILLEGAL INDEX KEY IN RECORD FFA1	SORTINTRINS
912	FF00/SORTINTRINS [172] RECORDS LOST/GAINED BY SORT-MERGE	SORTINTRINS
913	FF00/SORTINTRINS [179] SORT-MERGE ABNORMAL EOJ	SORTINTRINS
914	FF00/SORTINTRINS [186] TOO MANY RECORDS FOR SORT-MERGE	SORTINTRINS
915	FF00/SORTINTRINS [195] BAD RECORD/BLOCK SIZE FOR OUTPUT DEVICE	SORTINTRINS
916	FF00/SORTINTRINS [197] CANNOT SPLIT INDEX FILE	SORTINTRINS

917	FF00/SORTINTRINS [454] LOC: FFD1, FETCH.VALUE FFD2	SORTINTRINS
918	FF00/SORTINTRINS [455] FILE ERROR FFA1 ON SORT- MERGE FILE FFC2	SORTINTRINS
919	FFB0 CREATED	SAU.INIT
920	FFB0 NOT CREATED	SAU.INIT
921	FFB0 IS VERSION FFC1	SAU.INIT BUILDTRANS TRANSLATOR
922	SAU.INIT DEFINE PARAMETERS INCOMPLETE	SAU.INIT
923	PARAMETERS IN FILE FFB0 – FFC1 DISK FFC2 AS FFC3 FOR FFC4 FILES,	SAU.INIT
924	SERIAL NO. FFC0, OWNER FFC1, DATE FFC2	SAU.INIT
925	READ ERROR ON FFB0	SAU.INIT
926	WRITE ERROR ON FFB0	SAU.INIT
927	FFB0 NOT CMS STANDARD	SAU.INIT
928	DO YOU WANT TO CHANGE IT? – <Y OR N>	CONFIGURER
929	SPO LOGGING IS SWITCHED ON	CONFIGURER
930	SPO LOGGING IS SWITCHED OFF	CONFIGURER
931	IS LOGGING OF SYSTEM MESSAGES TO BE CYCLIC? – <Y OR N>	CONFIGURER
932	CURRENT BACKUP DISK FOR LOG FILES IS FFC0	CONFIGURER
933	CURRENT TEXT TO BE ZIPPED AT WARMSTART IS FFC0	CONFIGURER
934	CURRENT POWER OFF MESSAGE IS FFC0	CONFIGURER
935	LOCAL SPO IS FFC0	CONFIGURER
936	THE SYSTEM HAS A REMOTE SPO	CONFIGURER
937	THE SYSTEM DOES NOT HAVE A REMOTE SPO	CONFIGURER
938	CURRENT ID OF MESSAGE FILE IS FFC0	CONFIGURER
939	CURRENT REMOTE SPO MCS IS FFC0	CONFIGURER
940	READ ERROR ON KEYFILE FFB0 – KEYFILE/ DATAFILE PAIR NOT REMOVED	SYS-SUPERUTL
941	WORK FILES REMOVED	BUILDTRANS TRANSLATOR
942	LIST OF FFC0 SECTION EXHAUSTED AFTER FFA1 ENTRIES	TRANSLATOR
943	FFB0 NOT AVAILABLE – DISK LOCKED	BUILDTRANS CHECKADUMP COPY DUMPADISK FS LD LIST SYS-SUPERUTL TAPELR TAPEPD TL TRANSLATOR WL COMPARE CHECK.DISK LR KA

		CREATE
		AMEND
		UPDATE
944	WORK FILES NOT AVAILABLE – DISK LOCKED	KEY.CHECK
		BUILDTRANS
		TRANSLATOR
945	DO YOU WANT TO CHANGE ANY TRANSLATION TABLES? – <Y OR N>	CONFIGURER
946	TABLE NUMBER FFA0	CONFIGURER
947	CURRENT NUMBER OF TRANSLATION TABLES IS FFA0	CONFIGURER
948	CURRENT TRANSLATOR NAME IS FFC0	CONFIGURER
949	CURRENT DEVICE KIND AND NUMBER IS FFC0 FFC1	CONFIGURER
950	ENTER NUMBER OF TRANSLATION TABLES	CONFIGURER
951	ENTER DEVICE KIND AND NUMBER	CONFIGURER
952	ENTER NAME OF TRANSLATOR	CONFIGURER
953	ILLEGAL NUMBER OF TRANSLATION TABLES – FFC0	CONFIGURER
954	PARAMETER SPECIFICATION FOR <IN> INCOMPLETE	DSKUTL
955	FFC0 IS CMS DISK FFC1 – ENTER OK TO CONFIRM <IN> REQUIRED	DSKUTL
956	DSKUTL <IN> OF FFC0 NOT CONFIRMED	DSKUTL
957	DSKUTL <IN> ABORTED	DSKUTL
958	FFC0 WILL BE INITIALIZED AS PHYSICAL UNIT FFC1	DSKUTL
959	DSKUTL <IN> OF FFC0 DISK <FFC1> AS FFC2 ON FFC3 COMPLETED	DSKUTL
960	INITIALIZATION ERROR ON FFC0 AT SECTOR FFC1	DSKUTL
961	INIT.TRACK COMMUNICATE NOT SUPPORTED	DSKUTL
962	TEST PATTERN<S> NOT REQUIRED FOR FFC0 DISK	DSKUTL
963	READY AND RESERVE REQUIRED DRIVE FOR DISK INITIALIZE	DSKUTL
964	KEYFILE FFB0 OR DATA FILE NOT AVAILABLE – DISK LOCKED	COPY
		LIST
965	<MAXIMUM 255>	CONFIGURER
966	CURRENT PRINTER BACKUP FILE SIZE IS FFA0 RECORD(S)	CONFIGURER
967	ENTER PRINTER BACKUP FILE SIZE IN RECORDS	CONFIGURER
968	<MINIMUM 1, MAXIMUM 1048560>	CONFIGURER
969	PRINTER FILES ARE SENT TO BACKUP	CONFIGURER
970	PRINTER FILES ARE NOT SENT TO BACKUP	CONFIGURER
971	ARE PRINTER FILES TO GO TO BACKUP? – <Y OR N>	CONFIGURER
972	CURRENT PRINTER BACKUP DESIGNATE DISK IS FFC0	CONFIGURER
973	ENTER ID OF PRINTER BACKUP DESIGNATE DISK	CONFIGURER
975	WARNING – FFB0 FILETYPE IS FFC1, FFB2 FILETYPE IS FFC3	COMPARE
976	WARNING – FFB0 RECORD SIZE IS FFC1, FFB2 RECORD SIZE IS FFC3	COMPARE

977	WARNING – COMPARISON KEY LENGTHS FOR FFB0 AND FFB1 ARE DIFFERENT	COMPARE
978	– ONLY FFC0 BYTES WILL BE COMPARED	COMPARE
980	LOGGING OF SYSTEM MESSAGES IS CYCLIC	CONFIGURER
981	LOGGING OF SYSTEM MESSAGES IS NON-CYCLIC	CONFIGURER
982	ENTER SIZE OF PRIMARY LOG FILES IN SECTORS	CONFIGURER
983	ENTER SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IN SECTORS	CONFIGURER
985	<MINIMUM 32, MAXIMUM 262140>	CONFIGURER
986	CURRENT SIZE OF PRIMARY LOG FILES IS FFA0 SECTORS	CONFIGURER
987	CURRENT SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IS FFA0 SECTORS	CONFIGURER
988	ENTER SIZE OF SECONDARY MAINTENANCE ENTRY FILE(S) IN SECTORS	CONFIGURER
989	CURRENT SIZE OF SECONDARY MAINTENANCE ENTRY FILE(S) IS FFA0 SECTORS	CONFIGURER
990	ENTER ID OF BACKUP DISK FOR LOG FILES	CONFIGURER
991	MCS/DATA COMM SUBSYSTEM NOT AVAILABLE	PB
992	INVALID STATION NAME	PB
993	PRINTER DEVICE SPECIFIED IS NOT AVAILABLE	PB
994	CANNOT CREATE A FILE "DATA" ON ICMD FFC0 – RESERVED NAME	ICMD
995	FILESIZE OF DISK FILE FFB0 TOO SMALL	ICMD
996	ZIP FAILURE DUE TO SUITABLE MIX FOR FFC0	ARCS CO
997	ZIP FAILURE DUE TO DUAL ALPHABET/REVERSE ESCAPEMENT NOT SUPPORTED FFC0	ARCS CO
998	ZIP FAILURE DUE TO INSUFFICIENT REAL STORE FOR FFC0	ARCS CO
999	ZIP FAILURE DUE TO DISK LOCKED FOR FFC0	ARCS CO
1000	CO ERROR FFA0 INVALID ENTRY ENCLOSED IN QUOTES IN FFC1 STATEMENT	CO
1001	CO ERROR FFA0 FFC1 PROGRAM NAME INVALID	CO
1002	CO WARNING FFA0 PACK NAME IN FFC1 STATEMENT TRUNCATED TO 7 CHARACTERS	CO
1003	CO WARNING FFA0 FILE NAME IN FFC1 STATEMENT TRUNCATED TO 12 CHARACTERS	CO
1004	CO ERROR FFA0 NO COMPILE WITH PROGRAM NAME FFC1 EXISTS	CO
1005	CO ERROR FFA0 NO COMPILE WITH PROGRAM NAME FFC1 AND COMPILE TAG FFC2 EXISTS	CO
1006	FFA0 COMPILES EXIST WITH PROGRAM NAME FFC1, THEIR DETAILS ARE AS FOLLOWS :	CO
1007	ENTER COMPILE TAG OF REQUIRED COMPILE, OR NULL ACCEPT TO TERMINATE FFC0 ATTEMPT	CO
1008	COMPILE TAG NOT APPROPRIATE. PLEASE TRY AGAIN.	CO
1009	COMPILE WITH PROGRAM NAME FFC0 RESTARTED	CO

1010	COMPILE WITH PROGRAM NAME FFC0 AND COMPILE TAG FFC1 RESTARTED	CO
1011	COMPILE WITH PROGRAM NAME FFC0 CLEARED	CO
1012	COMPILE WITH PROGRAM NAME FFC0 AND COMPILE TAG FFC1 CLEARED	CO
1013	CO ERROR FFA0 INVALID ENTRY IN FFC1 STATEMENT	CO
1014	EXISTING FILE CO.MASTER IS INCOMPATIBLE WITH THIS VERSION OF CO	CO
1015	TO REMOVE CO.MASTER AND CONTINUE ENTER "Y" TO ACCEPT. ANY OTHER ENTRY WILL CAUSE CO TO TERMINATE	CO
1016	DSKUTL <FFC0> - ENTER NO. OF FILES <NUMBER LESS THAN 2805>	DSKUTL
1017	DSKUTL <FFC0> - ENTER SERIAL NO. <6 DIGITS>	DSKUTL
1018	DSKUTL <FFC0> - ENTER DISK NAME <UP TO 7 LEGAL CHARACTERS>	DSKUTL
1019	DSKUTL <FFC0> - ENTER OWNERS NAME <UP TO 14 CHARACTERS. DELIMIT WITH ">	DSKUTL
1020	DSKUTL <FFC0> - IS THIS DISK FOR MTR USE? - <Y OR N>	DSKUTL
1021	DSKUTL <FFC0> - ENTER REQUIRED BOOTSTRAP FILE-NAME OR NULL IF FFB1 IS TO BE USED	DSKUTL
1022	DSKUTL <IN> - ENTER NO. OF TEST PATTERNS TO BE SPECIFIED <NUMBER 0-4>	DSKUTL
1023	DSKUTL <IN> - ENTER FFC0 TEST PATTERN <20 CHARACTER MAX. DELIMIT WITH FFC1 IF HEX, " IF ASCII>	DSKUTL
1024	DSKUTL <RF> - IS DIRECTORY ON THIS DISK TO BE IGNORED? - <Y OR N>	DSKUTL
1025	FFC0 DEVICE TYPE NOT SUPPORTED	DSKUTL
1026	BOOTSTRAP CODE NOT COPIED TO OUTPUT FFC0 DISK FFC1	DSKUTL
1027	CANNOT RUN MTR ROUTINES ON OUTPUT FFC0 DISK FFC1	DSKUTL
1028	FFB0 NOT COPIED TO OUTPUT FFC1 DISK FFC2	DSKUTL
1029	AREAS POSSIBLY MISSING FROM OUTPUT FFC0 DISK FFC1	DSKUTL
1030	OR "UNIT.FDA" IF DISK FOR FIXED ASSEMBLAGE	DSKUTL
1031	COMPILE WITH PROGRAM NAME FFC0 AND COMPILE TAG FFCQ SHOULD BE RESTARTED OR CLEARED	BUILDTRANS
1037	WORKFILE REQUIRED - DO A FFC0 OR FFC1	BUILDTRANS
1038	NO TABLENAME SPECIFIED	BUILDTRANS
1039	ILLEGAL TABLENAME FFC0	BUILDTRANS
1040	DUPLICATE TABLENAME FFC0	BUILDTRANS
1041	TABLE FFC0 NOT FOUND	BUILDTRANS
1042	FFA0 ENTRIES LISTED	BUILDTRANS
1043	FFC0 IS NOT A VALID HEX STRING	BUILDTRANS
1044	ENTER TRANSLATE MODE	BUILDTRANS
1045	ENTER ANALYSIS MODE	BUILDTRANS
1046	ENTER HEX SHIFT CODES	BUILDTRANS

1047	ENTER MACHINE RESTRICTIONS (FFC0)	BUILDTRANS
1048	ENTER MAXIMUM SIZE OF TRANSLATION VALUES	BUILDTRANS
1049	FFC0 IS A SYSTEM TRANSLATION TABLE	BUILDTRANS
1050	FFA0 DIRECTORY ENTRIES, FFA1 TABLE(S) SAVED IN FILE FFB2	BUILDTRANS
1051	ILLEGAL DISK NAME FFC0	BUILDTRANS
1052	WARNING – TABLE FFC0 HAS MULTIPLE DIRECTORY ENTRIES	BUILDTRANS
1053	INVALID DIRECTORY ENTRY FFD0	BUILDTRANS
1054	NUMBER OF TABLES (FFA0) EXCEEDS UTILITY LIMIT (FFA1)	BUILDTRANS
1055	NUMBER OF DIRECTORY ENTRIES (FFA0) EXCEEDS UTILITY LIMIT (FFA1)	BUILDTRANS
1056	SIZE OF TABLE ENTRY (FFA0 BYTES) EXCEEDS UTILITY LIMIT (FFA1 BYTES)	BUILDTRANS
1057	TRANSLATION VALUES ARE FFA0 BYTE(S)	BUILDTRANS
1058	FFA0 ENTRY LISTED	BUILDTRANS
1059	FFA0 DIRECTORY ENTRY, FFA1 TABLE(S) SAVED IN FILE FFB2	BUILDTRANS
1060	NO ITEM NUMBER GIVEN	BUILDTRANS
1061	NO TRANSLATION VALUE(S) GIVEN	BUILDTRANS
1062	WARNING – FILE FFB0 MAY NOT BE PURGED	PL
1064	WARNING – ENTRY OPTION NOT ALLOWED FOR FILE FFB0	PL
1065	DEVICE FFC0 HAS BEEN SELECTED ALREADY	PL
1066	UNRECOGNIZED DEVICE IDENTIFICATION FFC0	PL
1067	NO DEVICE(S) FOLLOWING DEVICE SPECIFICATION	PL
1068	DEVICE FFC0 NOT FOUND IN HARDWARE CONFIGURATION AREA	PL
1069	INVALID IMPLEMENTATION LEVEL OF FFC0	SYS-SUPERUTL TL WL DSKUTL
1070	FFC0 OF FFC1 SURFACE CHECK PASSES COMPLETED ON FFC2	DSKUTL
1071	NO TEXT WILL BE ZIPPED AT WARMSTART	CONFIGURER
1072	THERE IS NO POWER OFF MESSAGE	CONFIGURER
1073	FF00 – PSEUDO UNRESTRICTED/SYSTEM	PPID
1074	FILE NOT DATA OR INDEX TYPE – FF00	MODIFY
1075	FILETYPE NOT 0-15 OR £80 – FF00	MODIFY
1076	DECIMAL NUMBER REQUIRED – FF00	MODIFY
1077	INVALID HARDWARE CONFIGURATION AREA IN FILE FFB0	TL
1078	CARD PERIPHERALS NOT IMPLEMENTED	SCR
1079	DUPLICATE SCR SUBTEST	SCR
1080	ERROR: TOTAL MAGNETIC DEVICES = 0	SCR
1081	ERROR: FILESIZE = 0	SCR
1082	WARNING: SECOND DISK.ID = FIRST DISK.ID	SCR
1083	DEFAULT VALUES ARE: TESTS 1,2,4	SCR
1084	DCR – ALL SUBTESTS, 1 DISK	SCR
1085	INDEX FILESIZE = 100 RECORDS	SCR
1086	READING 1000 RANDOM SECTORS	SCR
1087	CPU.IO – 1 ITERATION	SCR
1088	OTHER.IO – 1 ITERATION	SCR

1089	EXPECTED 'BEFORE' AFTER 'DETAIL'	RB
		BF
1090	EXPECTED COMMA OR 'BEFORE' AFTER FILEID	RB
		BF
1091	EXPECTED COMMA, 'DETAIL' OR 'BEFORE' AFTER FILEID	RB
		BF
1092	EXPECTED INTEGER AS SECOND HALF OF BACKUP NUMBER RANGE	RB
		BF
1093	EXPECTED INTEGER AS SECOND HALF OF DELETE RANGE	RB
		BF
1094	BACKUP NUMBER RANGE UPPER BOUND LESS THAN LOWER BOUND	RB
		BF
1095	DELETE RANGE UPPER BOUND LESS THAN LOWER BOUND	RB
		BF
1096	FFC0 SYNTAX ERROR	RB
		BF
1097	TOKEN SCANNED WAS: FFC0	RB
		BF
1098	'FFC0 HELP' FOR INFO	RB
		BF
1099	TOO MANY DISKS SPECIFIED	RB
		BF
1100	INVALID PACK OR BACKUP FILE NAME	RB
		BF
1101	TOO MANY FILES SPECIFIED	RB
		BF
1102	EXPECTED INTEGER FOR MONTH IN DATE	RB
		BF
1103	EXPECTED INTEGER FOR DAY IN DATE	RB
		BF
1104	INTEGER FOR MONTH OUT OF RANGE	RB
		BF
1105	INTEGER FOR DAY OUT OF RANGE	RB
		BF
1106	INTEGER FOR MONTH MUST BE GREATER THAN ZERO	RB
		BF
1107	MALFORMED DATE FIELD	RB
		BF
1108	EXPECTED 2 DIGIT INTEGER FOR YEAR	RB
		BF
1109	GARBAGE FOUND AT END OF GOOD COMMAND	RB
		BF
1110	FFB0 NOT REMOVED - OPEN DENIED	RB
		BF
1111	FFB0 NOT REMOVED - CREATED AFTER SPECIFIED DATE	RB
		BF

1112	FFB0 NOT REMOVED – PURGE FAILED	RB
		BF
1113	FFB0 NOT REMOVED – NOT FOUND	RB
		BF
1114	FFB0 NOT REMOVED – IN USE	RB
		BF
1115	NO BACKUP FILES FOUND IN RANGE FFC0-FFC1	RB
		BF
1116	NO BACKUP FILES FOUND IN RANGE FFC0-FFC1 ON FFC2	RB
		BF
1117	COULD NOT FIND BACKUP FILE FFB0	RB
		BF
1118	NO BACKUP FILES FOUND IN FAMILY FFB0	RB
		BF
1119	NO BACKUP FILES FOUND ON DISK FFC0	RB
		BF
1120	FILEID, EQUAL SIGN OR INTEGER EXPECTED TO IDENTIFY FILE(S)	RB
		BF
1121	AP JOURNAL PRINTER NEEDS OPERATOR ATTENTION	PB
1122	SCR ABORTED	SCR
1123	SCR SUBTEST NAME EXPECTED	SCR
1124	0 IS INVALID PARAMETER	SCR
1125	TRUNCATED	SCR
1126	INVALID KEYWORD	SCR
1127	INVALID PARAMETER	SCR
1128	KEYWORD EXPECTED: FFC0	SCR
1129	LAST KEYWORD: FFC0	SCR
1130	NO. OF SUBTEST ITERATIONS EXPECTED	SCR
1131	DISK ID EXPECTED	SCR
1132	SCR PARAMETER PART EXCEEDS 40 CHARACTERS	SCR
1133	SCR INITIATING MESSAGE ERROR	SCR
1134	DCR SUBTEST NAME EXPECTED	SCR
1135	DUPLICATE DCR SUBTEST	SCR
1136	INDEX FILE SIZE EXPECTED	SCR
1137	NUMBER OF RANDOM RECORDS EXPECTED	SCR
1138	'M' OR 'S' EXPECTED	SCR
1139	ENTER NUMBER OF DCR ITERATIONS – DEFAULT – 1	SCR
1140	ERROR FFA0 MISSING LEFT BRACKET AFTER ELAPSED TIME	ARCS
1141	ERROR FFA0 PARAMETER TO ELAPSED.TIME NOT STRING OR VARIABLE	ARCS
1142	ERROR FFA0 TIME PARAMETER TO ELAPSED.TIME NOT OF LENGTH 8	ARCS
1143	ERROR FFA0 MISSING COMMA BETWEEN PARAMETERS IN ELAPSED.TIME STATEMENT	ARCS
1144	ERROR FFA0 DATE PARAMETER TO ELAPSED.TIME NOT OF LENGTH 9	ARCS
1145	ERROR FFA0 INVALID DATE PASSED AS PARAMETER TO ELAPSED.TIME – FFC1	ARCS

1146	ERROR FFA0 INVALID COMBINATION OF DATE AND TIME PARAMETERS IN ELAPSED.TIME STATEMENT	ARCS
1147	ERROR FFA0 CALCULATED ELAPSED TIME EXCEEDS THE MAXIMUM - 'NULL' VALUE RETURNED	ARCS
1148	ERROR FFA0 THE TIME HAS BEEN REQUESTED BUT THERE IS NO CLOCK ON THIS MACHINE	ARCS
1149	ERROR FFA0 INVALID TIME PASSED AS PARAMETER TO ELAPSED.TIME - FFC1	ARCS
1150	ERROR FFA0 MISSING RIGHT BRACKET AT END OF ELAPSED.TIME STATEMENT	ARCS
1151	DCR ABORTED	SCR
1152	ZIP FAILURE DUE TO FAULTY FILE EQUATE FOR FFC0	ARCS
1153	ZIP FAILURE DUE TO INVALID NON-LOAD REQUEST FFC0	ARCS
1154	ZIP FAILURE DUE TO DEFINED BUT UNIMPLEMENTED SCL FFC0	ARCS
1155	ZIP FAILURE DUE TO VALID BUT UNSUCCESSFUL NON-LOAD REQUEST FFC0	ARCS
1156	DO YOU WANT TO CHANGE ANY LOGGING OPTIONS? - <Y OR N>	CONFIGURER
1157	DO YOU WANT TO CHANGE ANY PRINTER BACKUP OPTIONS? - <Y OR N>	CONFIGURER
1158	DO YOU WANT TO CHANGE ANY SPO OPTIONS? - <Y OR N>	CONFIGURER
1159	FFC0 IS FFC1 FFC2	SECURITY
1160	OLD PASSWORD INVALID	PW
1161	PASSWORD CHANGED	PW
1162	PASSWORD NOT CHANGED	PW
1163	USERCODE FFC0 ALREADY EXISTS	MAKEUSER
1164	COPYING USERCODE FILE... PATIENCE	MAKEUSER
1165	USERCODE FFC0 ADDED	MAKEUSER
1166	USERCODE FFC0 NOT ADDED	MAKEUSER
1167	USERCODE FFC0 NOT FOUND	MAKEUSER
1168	USERCODE FFC0 DELETED	MAKEUSER
1169	USERCODE FFC0 IS PRIVILEGED	MAKEUSER
1170	USERCODE (FFC0) IS NOT PRIVILEGED	MAKEUSER
1171	PLEASE, ENTER YOUR NEXT COMMAND OR 'END'	MAKEUSER
1172	FFC0 SYNTAX IS:	SECURITY PW MAKEUSER MAKEGUARDFL MAKEUSER
1173	USERCODE FILE CORRUPTED - PROGRAM ABORTED	MAKEUSER
1175	SYSUSERCODE CONTAINS:	MAKEUSER
1176	SYSUSERCODE LISTED	MAKEUSER
1177	NO USERCODE FOUND IN SYSUSERCODE FILE	MAKEUSER
1178	FFC0 MUST RUN UNDER A USERCODE	SECURITY PW MAKEUSER MAKEGUARDFL MAKEUSER
1180	USERCODE FFC0 NOT DELETED	MAKEUSER

1181	ERROR FFA0 PRE-DECLARED VARIABLE – FFC1 – HAS BEEN RE-DECLARED	ARCS
1182	FFB0 DOES NOT CONTAIN A COLDSTART BOOTSTRAP	DSKUTL
1183	IRRECOVERABLE DATA COMM. ERROR ON FFC0 RUNNING FILEUTL – PROGRAM TERMINATING	FILEUTL
1184	FILE FFB0 NOT DUMPED – IS INCOMPLETE	LD
1185	FILE FFB0 NOT LOADED – NOT ON TAPE – WAS IN USE AT DUMP TIME	LD
1186	FILE FFB0 NOT LOADED – NOT ON TAPE – HAD BEEN REMOVED AT DUMP TIME	LD
1187	FILE FFB0 NOT LOADED – NOT ON TAPE – HAD BEEN ALTERED AT DUMP TIME	LD
1188	FILE FFB0 NOT LOADED – NOT ON TAPE – DISK WAS LOCKED AT DUMP TIME	LD
1189	FILE FFB0 NOT LOADED – NOT ON TAPE – WAS INCOMPLETE AT DUMP TIME	LD
1190	FILE FFB0 NOT LOADED – NOT ON TAPE – REASON UNKNOWN	LD
1191	ZIP FAILURE WHEN ZIPPING CHECKADUMP FROM LD	LD
1192	ENTER “Y” TO RETRY ZIP, OR “N” TO TERMINATE	LD
1193	COMMUNICATE FAILURE	LD
1194	FILE FFB0 NOT COMPARED – NOT ON TAPE – WAS IN USE AT DUMP TIME	CHECKADUMP
1195	FILE FFB0 NOT COMPARED – NOT ON TAPE – HAD BEEN REMOVED AT DUMP TIME	CHECKADUMP
1196	FILE FFB0 NOT COMPARED – NOT ON TAPE – HAD BEEN ALTERED AT DUMP TIME	CHECKADUMP
1197	FILE FFB0 NOT COMPARED – NOT ON TAPE – DISK WAS LOCKED AT DUMP TIME	CHECKADUMP
1198	FILE FFB0 NOT COMPARED – NOT ON TAPE – WAS INCOMPLETE AT DUMP TIME	CHECKADUMP
1199	FILE FFB0 NOT COMPARED – NOT ON TAPE – REASON UNKNOWN	CHECKADUMP
1200	COLDSTART REQUEST IGNORED – VALID ONLY ON B90 SYSTEM	DSKUTL
1201	FFC0 IS NOT A MINI DISK – CANNOT BE USED FOR COLDSTART	DSKUTL
1202	FUNCTION VALID ONLY ON B90 AND B900 SYSTEMS	DSKUTL
1203	COLDSTART OPTION IS VALID ONLY ON B90 SYSTEM	DSKUTL
1204	DISK FFC0 WAS NOT INITIALIZED ON FFC1 SYSTEM	DSKUTL
1205	DSKUTL <FFC0> – FFC1 BOOTSTRAP VERSION FFC2 COPIED TO DISK FFC3	DSKUTL
1206	DSKUTL <FFC0> – IS THIS DISK FOR COLDSTART USE? – <Y OR N>	DSKUTL
1207	DSKUTL <FFC0> ABORTED	DSKUTL
1208	DISK FFC0 CANNOT BE USED FOR MTR PURPOSES	DSKUTL
1209	FFC0 CAN ONLY BE RUN AS A DATA-COMM PROGRAM	FILEUTL
1210	SPECIFIED OUTPUT DISK FFC0 IS NOT AVAILABLE	INSTALL

1211	SPECIFIED OUTPUT DISK FFC0 IS LOCKED	INSTALL
1212	SPECIFIED OUTPUT DISK FFC0 IS DUPLICATE	INSTALL
1213	SYSTEM DISK FFC0 HAS DIRECTORY ERRORS	INSTALL
1214	SYSTEM DISK FFC0 IS NOT AVAILABLE	INSTALL
1215	FFB0 NOT COPIED – NOT FOUND	INSTALL
1216	FFB0 NOT COPIED – IN USE	INSTALL
1217	FFB0 COPIED	INSTALL
1218	NOW POWER OFF SYSTEM DISK FFC0 AND RE- WARMSTART FROM DISK FFC1	INSTALL
1219	PLEASE USE “COPY” TO COPY THESE FILES FROM SOME OTHER DISK TO DISK FFC0	INSTALL
1220	BEFORE POWERING OFF DISK FFC0 AND WARMSTARTING FROM DISK FFC1	INSTALL
1221	THE SYSTEM IS NOT A B90	INSTALL
1222	FFC0 NOT EXECUTED FROM THE SYSTEM DISK	INSTALL
1223	THE SYSTEM DISK FFC0 IS NOT A MINI	INSTALL
1224	SPECIFIED OUTPUT DISK FFC0 IS A MINI	INSTALL
1225	THE FOLLOWING FFC0 FILE(S) WERE NOT COPIED –	INSTALL
1226	FFB0 NOT COPIED – READ ERROR ON DISK FFC1	INSTALL
1227	FFB0 NOT COPIED – WRITE ERROR ON DISK FFC1	INSTALL
1228	FFC0 TERMINATED	INSTALL
1229	FFB0 NOT COPIED – FILE IS A KEYFILE	INSTALL
1230	FFB0 NOT COPIED – CANNOT ALLOCATE ON DISK FFC1	INSTALL
1231	DIRECTORY OF SPECIFIED DISK FFC0 IS FULL – DISK SHOULD BE REFORMATTED FOR A LARGER NUMBER OF FILES	INSTALL
1232	THE NUMBER OF MATCHED KEYFILE ENTRIES IS NOT EQUAL TO	KEY.CHECK
1233	THE NUMBER OF DATAFILE RECORDS WITH A KEY IN KEYFILE	KEY.CHECK
1234	DATAKOM ERROR – PB TERMINATING	PB
1235	ERROR CONDITION REPORTED BY FFC0	PB
1236	PRINTER DEVICE FFC0 IS NOT AVAILABLE	PB
1237	FFB0 EOF ENCOUNTERED BEFORE FFA1 RECORD(S) SKIPPED	PB
1238	DSKUTL <FFC0> CONTINUING – READ ERROR IGNORED	DSKUTL

Utility Message Usage – SYSLANGUAGE Version 03.04.05

Utilities v Message Numbers

Utility Name	Message Number	Message Text
AMEND	1	INVALID DICTIONARY ENTRY FFA0
	2	ILLEGAL PARAMETER LIST FFC0
	3	INVALID CHARACTER IN IDENTIFIER FFC0
	4	FFB0 NOT FOUND
	5	FFB0 NOT AVAILABLE – IN USE
	6	NO SPECIFICATION GIVEN
	7	DISK FFC0 NOT AVAILABLE

ARCS

195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
284 ILLEGAL PARAMETER LIST - BAD ATTRIBUTES
286 FFB0 FILE TYPE IS NOT DATA
287 ILLEGAL PARAMETER LIST - TABS ERROR
288 OUTPUT FILE FFB0 TOO SMALL
289 PERMANENT ERROR ON OUTPUT FILE FFB0
503 DISK FFC0 LOCKED
612 SYSTEM DISK IS LOCKED
943 FFB0 NOT AVAILABLE - DISK LOCKED
340 ARCS WARNING FFA0 IN FFB1 AT RECORD FFA2
341 ARCS WARNING FFA0 IN FFB1 AT SEQUENCE FFC2
342 ARCS ERROR FFA0 IN FFB1 AT RECORD FFA2
343 ARCS ERROR FFA0 IN FFB1 AT SEQUENCE FFC2
344 END OF ARCS
345 ZIP FAILURE DUE TO PROGRAM FILE NOT FOUND
FOR FFC0
346 ZIP FAILURE DUE TO INTERPRETER NOT FOUND
FOR FFC0
347 ZIP FAILURE DUE TO INSUFFICIENT MEMORY FOR
FFC0
348 ZIP FAILURE DUE TO NO USER DISK OR
DIRECTORY FULL FOR FFC0
349 ZIP FAILURE DUE TO FULL MIX FOR FFC0
350 ZIP FAILURE DUE TO USER COUNT ERROR FFC0
351 ZIP FAILURE DUE TO DUPLICATE PACK FFC0
352 ZIP FAILURE DUE TO INVALID LOAD REQUEST
FFC0
353 ZIP FAILURE DUE TO MCS ALREADY PRESENT
FFC0
354 ZIP FAILURE DUE TO DISK ERROR FFC0
355 ZIP FAILURE DUE TO CODE FILE ERROR FFC0
356 ZIP FAILURE DUE TO ILLEGAL DATA COMM LOAD
REQUEST FFC0
357 ZIPPED PROGRAM DS'ED FFC0
358 ZIPPED PROGRAM DP'ED FFC0
359 ZIP FAILURE, REASON UNKNOWN FFC0
360 CANNOT PAUSE
361 MISSING ARCS SPECIFICATION
362 PLEASE RE-ENTER NUMBER IN RANGE 1-65535
363 ERROR FFA0 NO ASSIGN SYMBOL AFTER ASSIGN
VARIABLE.
364 ERROR FFA0 VARIABLE OUT OF SCOPE OF
PROCEDURE.
365 ERROR FFA0 NO NUMBER ENTERED ON ACCEPT
OF NUMERIC VARIABLE.
366 ERROR FFA0 NON-NUMERIC CHARACTER ENTERED
FOR NUMERIC ACCEPT.
367 ERROR FFA0 NO IF FOR ENCOUNTERED ELSE.

368 ERROR FFA0 PROCEDURE NAME IN CALL
STATEMENT NOT VARIABLE OR IDENTIFIER.
369 ERROR FFA0 FILE NAME OF PROCEDURE FROM
FILE > 12 CHARACTERS.
370 ERROR FFA0 CALL FOR UNDECLARED PROCEDURE
ENCOUNTERED.
371 ERROR FFA0 FID AFTER MFID IN CALL FROM
CLAUSE > 12 CHARACTERS.
372 ERROR FFA0 OVERFLOW CAUSED BY ARITHMETIC
EXPRESSION EVALUATION – CHECK RESULTS.
373 ERROR FFA0 PARAMETER PASSED IN CALL
STATEMENT NOT VARIABLE OR LITERAL.
374 ERROR FFA0 INSUFFICIENT DATA SPACE FOR
CALL PARAMETER.
375 ERROR FFA0 FORMAL PARAMETER IN PROCEDURE
HEAD NOT AN IDENTIFIER.
376 ERROR FFA0 UNEQUAL NUMBER OF ACTUAL AND
FORMAL PARAMETERS FOR PROCEDURE.
377 ERROR FFA0 CALL PROCEDURE FROM FILE NAME
NOT STRING OR VARIABLE.
378 ERROR FFA0 FILE NAME IN CALL PROCEDURE
FROM CLAUSE TOO LONG.
379 ERROR FFA0 MFID PART OF FILE NAME IN CALL
PROCEDURE FROM CLAUSE > 7 CHARACTERS.
380 ERROR FFA0 VARIABLE ALREADY DECLARED IN
CURRENT SCOPE.
381 ERROR FFA0 STRING LENGTH EXPRESSION NOT
NUMERIC.
382 ERROR FFA0 STRING LENGTH IN DECLARE
STATEMENT = 0 OR > 255.
383 ERROR FFA0 INSUFFICIENT DATA SPACE FOR
DECLARED STRING.
384 ERROR FFA0 OPERATOR NOT FOLLOWED BY
UNARY OPERATOR, LITERAL, VARIABLE OR (.
385 ERROR FFA0 NO BEGIN FOR ENCOUNTERED END.
386 ERROR FFA0 MISSING LEFT BRACKET AFTER
FUNCTION NAME.
387 ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME
NOT STRING OR VARIABLE.
388 ERROR FFA0 UNEXPECTED END OF ARCS INPUT
FILE.
389 ERROR FFA0 MISSING FID AFTER MFID IN
FUNCTION CALL.
390 ERROR FFA0 BLANK STRING AS FILE NAME/DISK
NAME/TAPE NAME IN FUNCTION CALL.
391 ERROR FFA0 CONTROL STACK UNDERFLOW OR
OVERFLOW.
392 ERROR FFA0 OPERAND STACK UNDERFLOW OR
OVERFLOW.

393 ERROR FFA0 OPERATOR STACK UNDERFLOW OR
OVERFLOW.

394 ERROR FFA0 FID AFTER MFID IN FUNCTION CALL
TOO LONG.

395 ERROR FFA0 FILE MFID/TAPE MFID IN FUNCTION
CALL TOO LONG.

396 ERROR FFA0 NO DO FOR ENCOUNTERED UNTIL.

397 ERROR FFA0 MISSING RIGHT BRACKET AFTER
FUNCTION PARAMETER.

398 ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME
PARAMETER NOT STRING TYPE.

399 ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME
IN FUNCTION CALL TOO LONG.

400 ERROR FFA0 FILE NAME/DISK NAME/TAPE NAME
IN FUNCTION CALL INVALID.

401 ERROR FFA0 LIMIT ON NUMBER OF DECLARED
VARIABLES EXCEEDED.

402 ERROR FFA0 INPUT FILE EITHER NOT PRESENT OR
HAS WRONG FILE TYPE.

403 ERROR FFA0 HEX NUMBER CONTAINS MORE THAN
4 HEX CHARACTERS.

404 ERROR FFA0 NO PROCEDURE HEAD FOR
ENCOUNTERED ENDPROC.

405 ERROR FFA0 ENDPROC ENCOUNTERED AT LEVEL
OTHER THAN THE OUTERMOST.

406 ERROR FFA0 FIRST EXPRESSION OPERAND NOT
VARIABLE, LITERAL OR FUNCTION.

407 ERROR FFA0 UNTIL EXPRESSION NOT NUMERIC.

408 ERROR FFA0 INVALID OPERATOR SYMBOL.

409 ERROR FFA0 NON-STRING OPERANDS FOR STRING
OPERATOR.

410 ERROR FFA0 INSUFFICIENT DATA SPACE TO
EVALUATE STRING EXPRESSION.

411 ERROR FFA0 OPERATOR REQUIRES 2
PARAMETERS, ONLY 1 PRESENT.

412 ERROR FFA0 UNARY OPERATOR NOT FOLLOWED
BY VARIABLE, LITERAL OR LEFT BRACKET.

413 ERROR FFA0 LEFT BRACKET NOT FOLLOWED BY
VARIABLE, LITERAL OR UNARY OPERATOR.

414 ERROR FFA0 INVALID ENTRY AFTER OPERAND IN
EXPRESSION.

415 ERROR FFA0 TOO MANY OPERANDS FOR THE
NUMBER OF OPERATORS.

416 ERROR FFA0 TOO MANY OPERATORS FOR THE
NUMBER OF OPERANDS.

417 ERROR FFA0 NO LEFT BRACKET FOR
ENCOUNTERED RIGHT BRACKET.

418 ERROR FFA0 RIGHT BRACKET NOT PRECEDED BY
VARIABLE OR LITERAL.

451 ERROR FFA0 NUMBER ENTERED FOR NUMERIC
ACCEPT > 65535.
419 ERROR FFA0 INVALID OPERATOR ENCOUNTERED.
420 ERROR FFA0 RIGHT BRACKET IN EXPR NOT
FOLLOWED BY LEFT BRACKET, OPERATOR OR
VALID TERMINATOR.
421 ERROR FFA0 NO IF FOR ENCOUNTERED FI.
422 ERROR FFA0 NAME OF ARCS INPUT FILE INVALID.
423 ERROR FFA0 ARCS INPUT FILE NAME TOO LONG.
424 ERROR FFA0 UNTIL CLAUSE ENCOUNTERED IN
LOOP BODY.
425 ERROR FFA0 WAIT EXPRESSION NOT NUMERIC.
426 ERROR FFA0 NO DO FOR ENCOUNTERED WHILE.
427 ERROR FFA0 ENCOUNTERED A STRING OF LENGTH
> 255.
428 ERROR FFA0 NO THEN CLAUSE AFTER IF
EXPRESSION.
429 ERROR FFA0 NO DO FOR ENCOUNTERED OD.
430 ERROR FFA0 PROCEDURE NAME NOT AN
IDENTIFIER.
431 ERROR FFA0 PROCEDURE ALREADY DECLARED
WITH THIS NAME.
432 ERROR FFA0 END OF FILE ENCOUNTERED WHILE
LOOKING FOR ENDPROC.
433 ERROR FFA0 NUMBER OF PROCEDURES LIMIT
EXCEEDED.
434 ERROR FFA0 PROCEDURE NESTING NOT
PERMITTED.
435 ERROR FFA0 PROCEDURE DECLARATION NOT AT
OUTERMOST LEVEL.
436 ERROR FFA0 RETURN ENCOUNTERED OUTSIDE OF
A PROCEDURE.
437 ERROR FFA0 MISSING LEFT BRACKET AFTER
RPGSWITCH.
438 ERROR FFA0 INDICATOR-SETTING PARAMETER
AFTER RPGSWITCH NOT STRING OR VARIABLE.
439 ERROR FFA0 MISSING RIGHT BRACKET AT END OF
RPGSWITCH STATEMENT.
440 ERROR FFA0 FILE NAME IN RPGSWITCH
STATEMENT NOT STRING OR VARIABLE.
441 ERROR FFA0 FILE NAME IN RPGSWITCH
STATEMENT > 12 CHARACTERS.
442 ERROR FFA0 VARIABLE AS INDICATOR-SETTING
PARAMETER NOT STRING TYPE.
443 ERROR FFA0 VARIABLE AS FILE NAME
PARAMETER NOT STRING TYPE.
444 ERROR FFA0 NON-STANDARD VALUES USED IN
SWITCH SETTING.
445 ERROR FFA0 UNEXPECTED RESERVED WORD.
446 ERROR FFA0 UNEXPECTED SPECIAL CHARACTER.

447 ERROR FFA0 WHILE EXPRESSION NOT NUMERIC.
448 ERROR FFA0 WHILE ENCOUNTERED IN LOOP
BODY.
449 ERROR FFA0 ZIP STRING > 255 CHARACTERS.
450 ERROR FFA0 DECIMAL LITERAL > 65535.
846 ERROR FFA0 PROCEDURE CALL ENCOUNTERED
WITHIN A PROCEDURE IN A SECONDARY INPUT
FILE
996 ZIP FAILURE DUE TO SUITABLE MIX REQUIRED
FOR FFC0
997 ZIP FAILURE DUE TO DUAL ALPHABET/REVERSE
ESCAPEMENT NOT SUPPORTED FFC0
998 ZIP FAILURE DUE TO INSUFFICIENT REAL STORE
FOR FFC0
999 ZIP FAILURE DUE TO DISK LOCKED FOR FFC0
1140 ERROR FFA0 MISSING LEFT BRACKET AFTER
ELAPSED TIME
1141 ERROR FFA0 PARAMETER TO ELAPSED.TIME NOT
STRING OR VARIABLE
1142 ERROR FFA0 TIME PARAMETER TO ELAPSED.TIME
NOT OF LENGTH 8
1143 ERROR FFA0 MISSING COMMA BETWEEN
PARAMETERS IN ELAPSED.TIME STATEMENT
1144 ERROR FFA0 DATE PARAMETER TO ELAPSED.TIME
NOT OF LENGTH 9
1145 ERROR FFA0 INVALID DATE PASSED AS
PARAMETER TO ELAPSED.TIME – FFC1
1146 ERROR FFA0 INVALID COMBINATION OF DATE
AND TIME PARAMETERS IN ELAPSED.TIME
STATEMENT
1147 ERROR FFA0 CALCULATED ELAPSED TIME
EXCEEDS THE MAXIMUM – “NULL” VALUE
RETURNED
1148 ERROR FFA0 THE TIME HAS BEEN REQUESTED
BUT THERE IS NO CLOCK ON THIS MACHINE
1149 ERROR FFA0 INVALID TIME PASSED AS
PARAMETER TO ELAPSED.TIME – FFC1
1150 ERROR FFA0 MISSING RIGHT BRACKET AT END OF
ELAPSED.TIME STATEMENT
1152 ZIP FAILURE DUE TO FAULTY FILE EQUATE FOR
FFC0
1153 ZIP FAILURE DUE TO INVALID NON-LOAD
REQUEST FFC0
1154 ZIP FAILURE DUE TO DEFINED BUT
UNIMPLEMENTED SCL FFC0
1155 ZIP FAILURE DUE TO VALID BUT UNSUCCESSFUL
NON-LOAD REQUEST FFC0
1181 ERROR FFA0 PRE-DECLARED VARIABLE – FFC1 –
HAS BEEN RE-DECLARED
910 FF00 VERSION FF01
1089 EXPECTED ‘BEFORE’ AFTER ‘DETAIL’

BF

1120 FILEID, EQUALSIGN OR INTEGER EXPECTED TO IDENTIFY FILE(S)
1090 EXPECTED COMMA OR 'BEFORE' AFTER FILEID
1091 EXPECTED COMMA, 'DETAIL' OR 'BEFORE' AFTER FILEID
1092 EXPECTED INTEGER AS SECOND HALF OF BACKUP NUMBER RANGE
1093 EXPECTED INTEGER AS SECOND HALF OF DELETE RANGE
1094 BACKUP NUMBER RANGE UPPER BOUND LESS THAN LOWER BOUND
1095 DELETE RANGE UPPER BOUND LESS THAN LOWER BOUND
1096 FFC0 SYNTAX ERROR
1097 TOKEN SCANNED WAS: FFC0
1098 'FFC0 HELP' FOR INFO
1099 TOO MANY DISKS SPECIFIED
1100 INVALID PACK OR BACKUP FILE NAME
1101 TOO MANY FILES SPECIFIED
1102 EXPECTED INTEGER FOR MONTH IN DATE
1103 EXPECTED INTEGER FOR DAY IN DATE
1104 INTEGER FOR MONTH OUT OF RANGE
1105 INTEGER FOR DAY OUT OF RANGE
1106 INTEGER FOR MONTH MUST BE GREATER THAN ZERO
1107 MALFORMED DATE FIELD
1108 EXPECTED 2 DIGIT INTEGER FOR YEAR
1109 GARBAGE FOUND AT END OF GOOD COMMAND
1110 FFB0 NOT REMOVED - OPEN DENIED
1111 FFB0 NOT REMOVED - CREATED AFTER SPECIFIED DATE
1112 FFB0 NOT REMOVED - PURGE FAILED
1113 FFB0 NOT REMOVED - NOT FOUND
1114 FFB0 NOT REMOVED - IN USE
1115 NO BACKUP FILES FOUND IN RANGE FFC0-FFC1
1116 NO BACKUP FILES FOUND IN RANGE FFC0-FFC1 ON FFC2
1117 COULD NOT FIND BACKUP FILE FFB0
1118 NO BACKUP FILES FOUND IN FAMILY FFB0
1119 NO BACKUP FILES FOUND ON DISK FFC0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
17 INDEXED PAIR FFB0 FFB1 REMOVED
43 FUNCTION ABORTED
45 AU REVOIR UNTIL THE NEXT TIME
46 TEXT LENGTH EXCEEDS UTILITY LIMIT
48 BREAK COMMAND RECEIVED

BUILDTRANS

71 ILLEGAL FILE NAME FFC0
79 ILLEGAL VALUE FFC0
83 FFC0 OPTION ALREADY SET
106 WORK FILE(S) ARE PRESENT
107 RECOVER OR REMOVE REQUIRED
108 INVALID REQUEST FFC0
109 FFB0 IS NOT A RECOGNIZED SYSTEM FILE
110 FILE FFB0 HAS BEEN ALTERED
111 WORK FILE(S) BEING REMOVED
112 UNABLE TO RECOVER
113 WORK FILE(S) ALREADY IN USE
114 LAST OPERATION WAS FFC0
115 FFB0 FILE ALREADY PRESENT
116 SAVE OR REMOVE REQUIRED
119 UNKNOWN REQUEST FFC0
120 NO FILE NAME IN PARAMETERS
127 RECOVERY COMPLETED
130 **** UTILITY LIMIT REACHED ****
131 COMMAND IN PROCESS ...
132 I/O ERROR DETECTED IN WRITE TO FILE FFB0
133 I/O ERROR DETECTED IN READ FROM FILE FFB0
134 ERROR AT RECORD NUMBER FFA0
503 DISK FFC0 LOCKED
607 ENTER FF00 TO FINISH OR FF01 TO CONTINUE
612 SYSTEM DISK IS LOCKED
910 FF00 VERSION FF01
943 FFB0 NOT AVAILABLE – DISK LOCKED
921 FFB0 IS VERSION FFC1
941 WORK FILES REMOVED
944 WORK FILES NOT AVAILABLE – DISK LOCKED
1037 WORKFILE REQUIRED – DO A FFC0 OR FFC1
1038 NO TABLENAME SPECIFIED
1039 ILLEGAL TABLENAME FFC0
1040 DUPLICATE TABLE NAME FFC0
1041 TABLE FFC0 NOT FOUND
1042 FFA0 ENTRIES LISTED
1043 FFC0 IS NOT A VALID HEX STRING
1044 ENTER TRANSLATE MODE
1045 ENTER ANALYSIS MODE
1046 ENTER HEX SHIFT CODES
1047 ENTER MACHINE RESTRICTIONS (FFC0)
1048 ENTER MAXIMUM SIZE OF TRANSLATION VALUES
1049 FFC0 IS A SYSTEM TRANSLATION TABLE
1050 FFA0 DIRECTORY ENTRIES, FFA1 TABLE(S) SAVED
IN FILE FFB2
1051 ILLEGAL DISK NAME FFC0
1052 WARNING – TABLE FFC0 HAS MULTIPLE
DIRECTORY ENTRIES
1053 INVALID DIRECTORY ENTRY FFD0
1054 NUMBER OF TABLES (FFA0) EXCEEDS UTILITY
LIMIT (FFA1)
1055 NUMBER OF DIRECTORY ENTRIES (FFA0) EXCEEDS
UTILITY LIMIT (FFA1)

1056 SIZE OF TABLE ENTRY (FFA0 BYTES) EXCEEDS
UTILITY LIMIT (FFA1 BYTES)
1057 TRANSLATION VALUES ARE FFA0 BYTE(S)
1058 FFA0 ENTRY LISTED
1059 FFA0 DIRECTORY ENTRY, FFA1 TABLE(S) SAVED IN
FILE FFB2
1060 NO ITEM NUMBER GIVEN
1061 NO TRANSLATION VALUE(S) GIVEN

CHECK.DISK

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
7 DISK FFC0 NOT AVAILABLE
99 DISK FFC0 IS A PSEUDO-DISK
193 READ ERROR ON LABEL OF FFC0
222 ERROR NOTIFIED ON READING DISK FFC0
223 CHECK.DISK ON FFC0 COMPLETED -
224 NO ERRORS
225 FFC0 CONSISTENT ERRORS NOTIFIED
226 ONE CONSISTENT ERROR NOTIFIED
227 PLEASE POWER OFF DISK FFC0
228 DISK FFC0 EXCEEDS BAD SECTOR LIMIT
229 DISK FFC0 EXPIRING - BACKUP RECOMMENDED
260 SECTOR FFC0 DENOTED BAD IN DIRECTORY
261 SECTOR FFC0 NOT DENOTED BAD IN DIRECTORY
263 FFC0 BAD SECTOR<S> NOT DENOTED IN
DIRECTORY
264 SECTOR FFC0 BEYOND END OF DISK FFC1
265 ILLEGAL RANGE FROM SECTOR FFC0
266 CHECK.DISK ON FFC0 AREA FFC1 - FFC2
COMPLETED -
267 CHECK.DISK ON FFC0 SECTOR FFC1 - NO ERROR
503 DISK FFC0 LOCKED
524 NO CONSISTENT ERRORS
525 INCONSISTENT ERROR(S) NOTIFIED - POSSIBLE
MEDIUM/DRIVE FAULT
612 SYSTEM DISK IS LOCKED
943 FFB0 NOT AVAILABLE - DISK LOCKED

CHECKADUMP

2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
49 FFC0 NOT A RECOGNIZED DUMP TAPE
68 TAPE SHOULD HAVE BEEN PURGED
137 INVALID DIRECTORY ON TAPE FFC0
138 COMPARISON ERROR ON FFB0
139 ON DISK FILE HEADERS
140 FILE FFC0 NOT FOUND FOR CHECK
141 FILE FFC0 NOT AVAILABLE FOR CHECK
142 AROUND RECORD FFA0

143 DIFFERING FILE SIZES
144 DIFFERING FILE TYPES
145 DIFFERING RECORD SIZES
146 DIFFERING BLOCK SIZES
147 NO DISCREPANCIES BETWEEN
148 DUMP TAPE FFC0 AND DISK FFC1
149 DISCREPANCIES FOUND BETWEEN
150 COMPARISON TERMINATED
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
218 NO FILES FOUND FOR CHECK
219 NO FILES IN FAMILY FFC0 FOUND FOR CHECK
230 DIFFERING GENERATION NUMBERS
503 DISK FFC0 LOCKED
612 SYSTEM DISK IS LOCKED
943 FFB0 NOT AVAILABLE - DISK LOCKED
1194 FILE FFB0 NOT COMPARED - NOT ON TAPE -
WAS IN USE AT DUMP TIME
1195 FILE FFB0 NOT COMPARED - NOT ON TAPE -
HAD BEEN REMOVED AT DUMP TIME
1196 FILE FFB0 NOT COMPARED - NOT ON TAPE -
HAD BEEN ALTERED AT DUMP TIME
1197 FILE FFB0 NOT COMPARED - NOT ON TAPE -
DISK WAS LOCKED AT DUMP TIME
1198 FILE FFB0 NOT COMPARED - NOT ON TAPE -
WAS INCOMPLETE AT DUMP TIME
1199 FILE FFB0 NOT COMPARED - NOT ON TAPE -
REASON UNKNOWN

CO 345 ZIP FAILURE DUE TO PROGRAM FILE NOT FOUND
FOR FFC0
346 ZIP FAILURE DUE TO INTERPRETER NOT FOUND
FOR FFC0
347 ZIP FAILURE DUE TO INSUFFICIENT MEMORY FOR
FFC0
348 ZIP FAILURE DUE TO NO USER DISK OR
DIRECTORY FULL FOR FFC0
349 ZIP FAILURE DUE TO FULL MIX FOR FFC0
350 ZIP FAILURE DUE TO USER COUNT ERROR FFC0
351 ZIP FAILURE DUE TO DUPLICATE PACK FFC0
352 ZIP FAILURE DUE TO INVALID LOAD REQUEST
FFC0
353 ZIP FAILURE DUE TO MCS ALREADY PRESENT
FFC0
354 ZIP FAILURE DUE TO DISK ERROR FFC0
355 ZIP FAILURE DUE TO CODE FILE ERROR FFC0
356 ZIP FAILURE DUE TO ILLEGAL DATA COMM LOAD
REQUEST FFC0
359 ZIP FAILURE, REASON UNKNOWN FFC0
452 COMPILER BEING ZIPPED.
453 COMPILER NOT ZIPPED.
454 CANNOT INITIATE THIS CO - MAXIMUM NUMBER
OF CO'S ALREADY IN PROCESS.

460 NO FILE FFC0 TO EXECUTE.
461 ZIP FAILURE WHEN ZIPPING FFC0.
462 RPG COMPILER ATTEMPTED TO ZIP RPGCOPY –
NOT IMPLEMENTED.
472 COMPILER PASS DS'ED FFC0.
473 COMPILER PASS DP'ED FFC0.
474 CO – MIX EMPTY
475 FEATURE NOT IMPLEMENTED – JOB ABORTED.
476 CO ERROR FFA0 FIRST WORD IN INITIATING
MESSAGE IS NOT A VALID DISK OR FILE NAME.
477 CO ERROR FFA0 INVALID OBJECT PROGRAM
NAME OR COMPILER NAME SPECIFIED.
478 CO WARNING FFA0 COMPILER PACK ID.
TRUNCATED TO 7 CHARACTERS.
479 CO ERROR FFA0 NO COMPILER SPECIFIED.
480 CO WARNING FFA0 PROGRAM PACK ID.
TRUNCATED TO 7 CHARACTERS.
481 CO WARNING FFA0 PROGRAM ID. TRUNCATED TO
12 CHARACTERS.
482 CO ERROR FFA0 RESERVED WORD NOT FOUND
WHEN EXPECTED.
483 CO ERROR FFA0 FI/FILE NOT FOLLOWED BY
PATCHFILE/SOURCEIN/SOURCEOUT/WORKFILE/
PRINTOUT.
484 CO ERROR FFA0 RESERVED WORD USED OUT OF
CONTEXT
485 CO WARNING FFA0 COMPILER OPTION ALREADY
MODIFIED – PREVIOUS MOD. IGNORED.
486 CO WARNING FFA0 PRINT OPTION SPECIFIED
MORE THAN ONCE.
487 CO WARNING FFA0 RECORD SIZE*BLOCKING
FACTOR>65535. BLOCK SIZE SET EQUAL TO
RECORD SIZE.
488 CO WARNING FFA0 MESSAGE NOT USED BY
COBOL – MESSAGE TEXT IGNORED.
489 CO WARNING FFA0 MESSAGE TEXT ALREADY
PROCESSED – THIS MESSAGE TEXT IGNORED.
490 CO WARNING FFA0 MODIFICATION NOT VALID
FOR FF01 – IGNORED.
491 CO ERROR FFA0 COMPILER NOT ZIPPED –
ERRORS IN INITIATING MESSAGE.
492 CO ERROR FFA0 \$SEGMENT PARAMETER NOT A
VALID FILE NAME.
493 CO ERROR FFA0 CO.MASTER NO LONGER ON DISK
– JOB TERMINATED.
494 CO WARNING FFA0 \$SEGMENT PACK-ID
TRUNCATED TO 7 CHARACTERS.
495 CO WARNING FFA0 \$SEGMENT PROGRAM NAME
TRUNCATED TO 12 CHARACTERS.
496 CO ERROR FFA0 \$SEGSIZE PARAMETER NOT
NUMERIC.
497 CO WARNING FFA0 FF01 MODIFICATION NOT
VALID FOR THIS COMPILER – IGNORED.

498 CO WARNING FFA0 MESSAGE CONTAINS NO VALID
DOLLAR CARDS.
499 CO WARNING FFA0 \$CARD NOT VALID FOR THIS
COMPILER.
500 CO WARNING FFA0 INVALID \$CARD IN MESSAGE
TEXT.
501 CO WARNING FFA0 FF01 MODIFICATION(S)
ALREADY DONE – THESE MODS. IGNORED
502 CO WARNING FFA0 FF01 MODIFICATIONS
INCOMPLETE.
504 CO WARNING FFA0 NO FF01 MODIFICATIONS.
505 CO WARNING FFA0 MFID/FID MODIFICATION
ALREADY DONE FOR THIS FILE – MOD. IGNORED.
506 CO ERROR FFA0 PACK NAME/FILE NAME IN NAME
SPECIFICATION INVALID.
507 CO WARNING FFA0 PACK NAME IN NAME
SPECIFICATION TRUNCATED TO 7 CHARACTERS.
508 CO WARNING FFA0 FILE NAME IN NAME
SPECIFICATION TRUNCATED TO 12 CHARACTERS.
509 CO ERROR FFA0 PACK NAME IN MFID CLAUSE
INVALID.
510 CO WARNING FFA0 PACK NAME IN MFID CLAUSE
TRUNCATED TO 7 CHARACTERS.
511 CO ERROR FFA0 FILE NAME IN FID CLAUSE
INVALID.
512 CO WARNING FFA0 FILE NAME IN FID CLAUSE
TRUNCATED TO 12 CHARACTERS.
513 CO WARNING FFA0 RECORD SIZE OF ZERO
SPECIFIED – IGNORED – DEFAULT USED.
514 CO WARNING FFA0 DEVICE MODIFICATION
ALREADY DONE FOR THIS FILE – THIS MOD
IGNORED.
515 CO WARNING FFA0 SPECIFIED DEVICE TYPE
INVALID – DEFAULT DEVICE TYPE USED.
516 CO WARNING FFA0 FILE SIZE ALREADY SPECIFIED
FOR THIS FILE – THIS MOD. IGNORED.
517 CO ERROR FFA0 FILESIZE PARAMETER NOT
NUMERIC.
518 CO WARNING FFA0 BLOCK SIZE ALREADY
MODIFIED – THIS MOD. IGNORED.
519 CO ERROR FFA0 RECORDS.BLOCK/BLOCK
PARAMETER NOT NUMERIC.
520 CO ERROR FFA0 RECORD PARAMETER NOT
NUMERIC.
531 CO ERROR FFA0 NULL INITIATING MESSAGE – NO
FILE CO.STARTUP – JOB TERMINATED.
532 CO ERROR FFA0 NESTED MACRO CALL FOUND.
533 CO ERROR FFA0 INVALID FILE NAME IN MACRO
CALL – FFC1.
534 CO WARNING FFA0 PACK NAME IN MACRO CALL
TRUNCATED TO 7 CHARACTERS.
535 CO WARNING FFA0 FILE NAME IN MACRO CALL
TRUNCATED TO 12 CHARACTERS.

536 CO WARNING FFA0 RECORD SIZE FOR CARD
DEVICE > 96. RECORD AND BLOCK SIZE SET TO
96.

537 CO ERROR FFA0 FILE CO.STARTUP EMPTY – NO
INITIATING MESSAGE – JOB TERMINATED.

539 CO ERROR FFA0 CO.MASTER NOT FOUND – JOB
TERMINATED.

542 CO ERROR FFA0 NO CO'S TO RESTART/CLEAR –
JOB TERMINATED.

544 CO ERROR FFA0 SYSTEM.STATUS COMMUNICATE
FAILURE – CANNOT RUN CO ON THIS SYSTEM

546 CO WARNING FFA0 WORKFILE NOT FOLLOWED BY
MFID – MOD. IGNORED.

547 CO ERROR FFA0 SPECIFIED RESTART LINE-
NUMBER FOR LISTING NOT NUMERIC.

548 CO WARNING FFA0 SPECIFIED RESTART LINE-
NUMBER > 65535. LINE-NUMBER SET TO ZERO.

550 CO ERROR FFA0 NO PARAMETER SPECIFIED FOR
CLEAR FUNCTION.

551 CO WARNING FFA0 LINE-NUMBER PARAMETER
VALID ONLY FOR COBOL – IGNORED.

552 CO ERROR FFA0 ILLEGAL CHARACTER
ENCOUNTERED IN INITIATING MESSAGE.

553 CO ERROR FFA0 CANNOT CONTINUE
COMPILATION – COMPILATION BLOCK
INITIALIZED.

554 CO WARNING FFA0 NON-INPUT DEVICE SPECIFIED
FOR FF01 – IGNORED.

556 CO ERROR FFA0 MACRO FILE NOT FOUND – FFC1.

557 CO WARNING FFA0 NON-OUTPUT DEVICE
SPECIFIED FOR SOURCEOUT – IGNORED.

558 CO WARNING FFA0 BLOCK SIZE FOR SOURCEOUT
IS NOT A MULTIPLE OF THE RECORD SIZE.

559 CO WARNING FFA0 SPECIFIED BLOCK SIZE FOR
TAPE DEVICE TOO LARGE – DEFAULT USED.

560 CO WARNING FFA0 SPECIFIED RECORD SIZE FOR
TAPE DEVICE TOO LARGE – DEFAULT USED.

561 CO WARNING FFA0 80 COL CARD DEVICE RECORD
SIZE > 80. RECORD AND BLOCK SIZE SET TO 80.

562 CO WARNING FFA0 96 COL CARD DEVICE RECORD
SIZE > 96. RECORD AND BLOCK SIZE SET TO 96.

564 CO WARNING FFA0 DEVICE SPECIFIED FOR
PRINTOUT NOT PRINTER TYPE – IGNORED.

565 CO WARNING FFA0 PRINTOUT NOT FOLLOWED BY
DEVICE OR NO.BUFFERS – MOD. IGNORED.

566 CO WARNING FFA0 BLOCK SIZE OR BLOCKING-
FACTOR OF ZERO SPECIFIED – DEFAULT USED.

567 CO WARNING FFA0 SOURCEOUT FILESIZE
SPECIFIED AS ZERO – IGNORED – DEFAULT
USED.

568 CO WARNING FFA0 SPECIFIED FF01 FILESIZE >
65535 – DEFAULT USED.

570 CO WARNING FFA0 SPECIFIED RECORD SIZE >
65535 – DEFAULT USED.

571 CO WARNING FFA0 SPECIFIED BLOCK SIZE OR
BLOCKING-FACTOR > 65535 – DEFAULT USED.

572 CO ERROR FFA0 WHEN COMPILING MPL ON
CUMBERNAULD SUPPORT SYSTEM, LEVEL NO.
REQUIRED.

573 CO WARNING FFA0 IF 80 COL CARD OUTPUT
DEVICE USED, RECORD TRUNCATION WILL
OCCUR.

574 CO WARNING FFA0 RECORD SIZE ALREADY
MODIFIED – THIS MOD. IGNORED.

575 CO ERROR FFA0 MACRO FILE PRESENT BUT
EMPTY – FFC1.

577 CO ERROR FFA0 NO.BUFFERS PARAMETER NOT
NUMERIC.

578 CO WARNING FFA0 VALUE SPECIFIED FOR
NO.BUFFERS NOT IN RANGE 1 TO 16 – DEFAULT
USED.

579 CO WARNING FFA0 NO.BUFFERS ALREADY
MODIFIED – THIS MOD. IGNORED.

996 ZIP FAILURE DUE TO SUITABLE MIX REQUIRED
FOR FFC0

997 ZIP FAILURE DUE TO DUAL ALPHABET/REVERSE
ESCAPEMENT NOT SUPPORTED FFC0

998 ZIP FAILURE DUE TO INSUFFICIENT REAL STORE
FOR FFC0

999 ZIP FAILURE DUE TO DISK LOCKED FOR FFC0

1000 CO ERROR FFA0 INVALID ENTRY ENCLOSED IN
QUOTES IN FFC1 STATEMENT

1001 CO ERROR FFA0 FFC1 PROGRAM NAME INVALID

1002 CO WARNING FFA0 PACK NAME IN FFC1
STATEMENT TRUNCATED TO 7 CHARACTERS

1003 CO WARNING FFA0 FILE NAME IN FFC1
STATEMENT TRUNCATED TO 12 CHARACTERS

1004 CO ERROR FFA0 NO COMPILE WITH PROGRAM
NAME FFC1 EXISTS

1005 CO ERROR FFA0 NO COMPILE WITH PROGRAM
NAME FFC1 AND COMPILE TAG FFC2 EXISTS

1006 FFA0 COMPILES EXIST WITH PROGRAM NAME
FFC1, THEIR DETAILS ARE AS FOLLOWS:

1007 ENTER COMPILE TAG OF REQUIRED COMPILE, OR
NULL ACCEPT TO TERMINATE FFC0 ATTEMPT

1008 COMPILE TAG NOT APPROPRIATE. PLEASE TRY
AGAIN.

1009 COMPILE WITH PROGRAM NAME FFC0 RESTARTED

1010 COMPILE WITH PROGRAM NAME FFC0 AND
COMPILE TAG FFC1 RESTARTED

1011 COMPILE WITH PROGRAM NAME FFC0 CLEARED

1012 COMPILE WITH PROGRAM NAME FFC0 AND
COMPILE TAG FFC1 CLEARED

1013 CO ERROR FFA0 INVALID ENTRY IN FFC1
STATEMENT

1014 EXISTING FILE CO.MASTER IS INCOMPATIBLE
WITH THIS VERSION OF CO
1015 TO REMOVE CO.MASTER AND CONTINUE ENTER
'Y' TO ACCEPT. ANY OTHER ENTRY WILL CAUSE
CO TO TERMINATE
1031 COMPILE WITH PROGRAM NAME FFC0 AND
COMPILE TAG FFC1 SHOULD BE RESTARTED OR
CLEARED

COMPARE

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
102 FILE LIST MAY BE INCOMPLETE
194 READ ERROR ON PPIT OF FFC0
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
303 NO FILES FOUND IN THE FAMILY FFB0
304 FFB0 FILE IDENTIFIER TOO LONG
305 FILE LIST FOR FFB0 MAY BE INCOMPLETE
503 DISK FFC0 LOCKED
526 FIRST DIFFERENCE FOUND BETWEEN FILES -
545 FFB0 RECORD. FF01 AT BYTE. FF02
595 FFB0 FILE IDENTIFIER TOO SHORT
596 ILLEGAL KEY FOR FFB0 : FFA1-FFA2
597 ILLEGAL RECORDS FOR FFB0 : FFA1
598 CANNOT COMPARE PAST POSITION FFA0 IN FFB1
599 CANNOT READ RECORD FFA0 OF FFB1
600 FFB0 EXHAUSTED AT FFA1
601 CANNOT REALIGN FFB0 WITH FFB1 - FFA2
ERRORS
602 END OF FILE FFB0 BEFORE FFB1 - FFA2 ERRORS
603 FFB0 WITH FFB1 COMPARED - FFA2 ERRORS
612 SYSTEM DISK IS LOCKED
943 FFB0 NOT AVAILABLE - DISK LOCKED
975 WARNING - FFB0 FILETYPE IS FFC1, FFB2
FILETYPE IS FFC3
976 WARNING - FFB0 RECORD SIZE IS FFC1, FFB2
RECORD SIZE IS FFC3
977 WARNING - COMPARISON KEY LENGTHS FOR
FFB0 AND FFB1 ARE DIFFERENT
978 - ONLY FFC0 BYTES WILL BE COMPARED

CONFIGURER

3 INVALID CHARACTER IN IDENTIFIER FFC0
46 TEXT LENGTH EXCEEDS UTILITY LIMIT
82 NO FFC0 NUMBER GIVEN
132 I/O ERROR DETECTED IN WRITE TO FILE FFB0
133 I/O ERROR DETECTED IN READ FROM FILE FFB0
134 ERROR AT RECORD NUMBER FFA0
159 JOB COMPLETED
163 ENTER TEXT TO BE ZIPPED AT WARMSTART

165 ENTER POWER OFF MESSAGE
166 SPO LOGGING REQUIRED? - <Y OR N>
167 ** INVALID SELECTION - RETRY **
169 ILLEGAL FILE SIZE - FFC0
171 ENTER LOCAL SPO OPTION - <FFC0 OR FFC1>
173 <MINIMUM 32, MAXIMUM 16383>
174 ENTER FILE-ID OF REQUIRED MESSAGE FILE
620 DOES THE SYSTEM HAVE A LOCAL SPO? - <Y
OR N>
621 DOES THE SYSTEM HAVE A REMOTE SPO? - <Y
OR N>
623 ENTER FILE-ID AND INITIATING MESSAGE OF
REMOTE SPO MCS
928 DO YOU WANT TO CHANGE IT? - <Y OR N>
929 SPO LOGGING IS SWITCHED ON
930 SPO LOGGING IS SWITCHED OFF
931 IS LOGGING OF SYSTEM MESSAGES TO BE CYCLIC?
- <Y OR N>
932 CURRENT BACKUP DISK FOR LOG FILES IS FFC0
933 CURRENT TEXT TO BE ZIPPED AT WARMSTART IS
FFC0
934 CURRENT POWER OFF MESSAGE IS FFC0
935 LOCAL SPO IS FFC0
936 THE SYSTEM HAS A REMOTE SPO
937 THE SYSTEM DOES NOT HAVE A REMOTE SPO
938 CURRENT ID OF MESSAGE FILE IS FFC0
939 CURRENT REMOTE SPO MCS IS FFC0
945 DO YOU WANT TO CHANGE ANY TRANSLATION
TABLES? - <Y OR N>
946 TABLE NUMBER FFA0
947 CURRENT NUMBER OF TRANSLATION TABLES IS
FFA0
948 CURRENT TRANSLATOR NAME IS FFC0
949 CURRENT DEVICE KIND AND NUMBER IS FFC0FFC1
950 ENTER NUMBER OF TRANSLATION TABLES
951 ENTER DEVICE KIND AND NUMBER
952 ENTER NAME OF TRANSLATOR
953 ILLEGAL NUMBER OF TRANSLATION TABLES -
FFC0
965 <MAXIMUM 255>
966 CURRENT PRINTER BACKUP FILE SIZE IS FFA0
RECORD(S)
967 ENTER PRINTER BACKUP FILE SIZE IN RECORDS
968 <MINIMUM 1, MAXIMUM 1048560>
969 PRINTER FILES ARE SENT TO BACKUP
970 PRINTER FILES ARE NOT SENT TO BACKUP
971 ARE PRINTER FILES TO GO TO BACKUP? - <Y
OR N>
972 CURRENT PRINTER BACKUP DESIGNATE DISK IS
FFC0
973 ENTER ID OF PRINTER BACKUP DESIGNATE DISK
980 LOGGING OF SYSTEM MESSAGES IS CYCLIC
981 LOGGING OF SYSTEM MESSAGES IS NON-CYCLIC

982 ENTER SIZE OF PRIMARY LOG FILES IN SECTORS
983 ENTER SIZE OF SECONDARY SYSTEM MESSAGE
FILE(S) IN SECTORS
985 <MINIMUM 32, MAXIMUM 262140>
986 CURRENT SIZE OF PRIMARY LOG FILES IS FFA0
SECTORS
987 CURRENT SIZE OF SECONDARY SYSTEM MESSAGE
FILE(S) IS FFA0 SECTORS
988 ENTER SIZE OF SECONDARY MAINTENANCE
ENTRY FILE(S) IN SECTORS
989 CURRENT SIZE OF SECONDARY MAINTENANCE
ENTRY FILE(S) IS FFA0 SECTORS
990 ENTER ID OF BACKUP DISK FOR LOG FILES
1071 NO TEXT WILL BE ZIPPED AT WARMSTART
1072 THERE IS NO POWER OFF MESSAGE
1156 DO YOU WANT TO CHANGE ANY LOGGING
OPTIONS? - <Y OR N>
1157 DO YOU WANT TO CHANGE ANY PRINTER
BACKUP OPTIONS - <Y OR N>
1158 DO YOU WANT TO CHANGE ANY SPO OPTIONS? -
<Y OR N>

COPY

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
16 FFB0 REMOVED
102 FILE LIST MAY BE INCOMPLETE
151 RECORD SIZE TOO GREAT
194 READ ERROR ON PPIT OF FFC0
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
231 BAD ATTRIBUTES SPECIFIED
232 FFB0 - DATA FILE NOT FOUND
233 KEY FILE FFB0 OR DATA FILE IN USE
235 KEY FILE FFB0 OR DATA FILE NOT FOUND
236 FFB0 NOT ACCEPTABLE - RECORD SIZE OF FFA1
EXCEEDS MAXIMUM FOR THIS RUN - RESUBMIT
237 FFA0, FFA1 IN FFB2 NOT COPIED
238 FFA0, EOF IN FFB2 NOT COPIED
239 FFB0 EXHAUSTED DURING RANGE - FF01 - FF02
240 FFB0 EXHAUSTED DURING RANGE - FF01 FFA2
241 FFB0 EXHAUSTED DURING RANGE - FF01 EOF
242 FFB0 EXHAUSTED DURING FFA1, FFA2
243 FFB0 EXHAUSTED DURING FFA1, EOF
244 SELECTIONS OF RECORDS FROM FFA0 IGNORED
245 NO RECORDS FOR COPYING FROM FFB0
246 FFB0 TO FFB1 COPY FAILURE - NO RECORDS IN
KEY FILE

247 FFB0 TO FFB1 COPY FAILURE – INPUT RECORD
 NO. FFA2, OUTPUT RECORD NO. FFA3 PERMANENT
 ERROR ON INPUT FILE
 248 FFB0 TO FFB1 COPY FAILURE – INPUT RECORD
 NO. FFA2, OUTPUT RECORD NO. FFA3 PERMANENT
 ERROR ON OUTPUT FILE
 249 FFB0 TO FFB1 COPY FAILURE – INPUT RECORD
 NO. FFA2, OUTPUT RECORD NO. FFA3 OUTPUT
 FILE TOO SMALL
 250 FFB0 TO FFB1 COPY FAILURE – INPUT RECORD
 KEY FF02, OUTPUT RECORD NO. FFA3 PERMANENT
 ERROR ON INPUT FILE
 251 FFB0 TO FFB1 COPY FAILURE – INPUT RECORD
 KEY FF02, OUTPUT RECORD NO. FFA3 PERMANENT
 ERROR ON OUTPUT FILE
 252 FFB0 TO FFB1 COPY FAILURE – INPUT RECORD
 KEY FF02, OUTPUT RECORD NO. FFA3 OUTPUT
 FILE TOO SMALL
 253 FFB0 TO FFB1 COPIED
 254 FFB0 TO FFB1 BAD ATTRIBUTES
 255 FFB0 NOT FOUND FOR EXTENDING
 256 FFB0 TO FFB1 ILLEGAL REQUEST
 257 FFB0 TO FFB1 ILLEGAL REQUEST SELECTION
 CRITERIA IGNORED
 258 FFB0 TO FFB1 ILLEGAL REQUEST EXTENDING
 FLAG IGNORED
 259 FAMILY FFB0 TO FFB1 COPIED
 301 INVALID CHARACTER IN IDENTIFIER FFB0
 302 INVALID CHARACTER IN IDENTIFIER FFB0 –
 WARNING
 303 NO FILES FOUND IN THE FAMILY FFB0
 304 FFB0 FILE IDENTIFIER TOO LONG
 305 FILE LIST FOR FFB0 MAY BE INCOMPLETE
 503 DISK FFC0 LOCKED
 521 WARNING – DISK FFC0 NOT AVAILABLE
 522 WARNING – FILE LIST FOR FFB0 MAY BE
 INCOMPLETE
 624 WARNING – DISK FFC0 LOCKED
 625 WARNING – FILE LIST MAY BE INCOMPLETE
 943 FFB0 NOT AVAILABLE – DISK LOCKED
 964 KEYFILE FFB0 OR DATAFILE NOT AVAILABLE –
 DISK LOCKED
 CP
 175 CP. FFC0
 176 CP. NUMBER TOO LARGE
 177 CP. ILLEGAL NUMBER FFC0
 178 CP. MISSING NUMBER
 179 CP. ILLEGAL HEX. NUMBER
 180 CP. OVERFLOW
 181 CP. DIVISION BY ZERO
 182 CP. INVALID OPERATOR FFC0
 183 CP. MISSING OPERATOR
 CREATE
 1 INVALID DICTIONARY ENTRY FFA0

2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE – IN USE
 6 NO SPECIFICATION GIVEN
 7 DISK FFC0 NOT AVAILABLE
 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 284 ILLEGAL PARAMETER LIST – BAD ATTRIBUTES
 286 FFB0 FILE TYPE IS NOT DATA
 287 ILLEGAL PARAMETER LIST – TABS ERROR
 288 OUTPUT FILE FFB0 TOO SMALL
 289 PERMANENT ERROR ON OUTPUT FILE FFB0
 503 DISK FFC0 LOCKED
 612 SYSTEM DISK IS LOCKED
 656 CANNOT OPEN FILE FFB0 – DIRECTORY FULL
 943 FFB0 NOT AVAILABLE – DISK LOCKED

DA

4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE – IN USE
 7 DISK FFC0 NOT AVAILABLE
 99 DISK FFC0 IS A PSEUDO-DISK
 193 READ ERROR ON LABEL OF FFC0
 503 DISK FFC0 LOCKED
 612 SYSTEM DISK IS LOCKED

DSKUTL

1 INVALID DICTIONARY ENTRY FFA0
 2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE – IN USE
 6 NO SPECIFICATION GIVEN
 7 DISK FFC0 NOT AVAILABLE
 193 READ ERROR ON LABEL OF FFC0
 194 READ ERROR ON PPIT OF FFC0
 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 272 NO PPIT ON SYSTEM
 503 DISK FFC0 LOCKED
 612 SYSTEM DISK IS LOCKED
 615 OPEN DISK UNLABELLED NOT IMPLEMENTED
 616 DISK LABEL OF FFC0 NOT IN CMS FORMAT
 626 READY AND RESERVE REQUIRED DRIVE FOR DISK
 REFORMAT
 627 CANNOT READ SECTOR 0 OF FFC0
 628 FFC0 IS CMS DISK FFC1 – ENTER OK TO CONFIRM
 <RF> REQUIRED
 629 DSKUTL <RF> OF FFC0 NOT CONFIRMED
 630 FFC0 IS NOT A FIXED DISK
 631 FFC0 WILL BE REFORMATTED AS PHYSICAL UNIT
 FFC1
 632 TRACK 0 OF FFC0 BAD
 633 BAD MTR SECTOR FFC0 ON FFC1
 634 BOOTSTRAP FILE FFB0 NOT CMS STANDARD

635 READ ERROR ON OLD DIRECTORY OF FFC0 AT
SECTOR FFC1

636 OLD DIRECTORY OF FFC0 IS CORRUPT

637 TOO MANY BAD ENTRIES IN THE OLD AVAILABLE
TABLE FFC0

638 CANNOT ACCOMMODATE DIRECTORY FOR FFC0
639 AVAILABLE TABLE TOO SMALL FOR FFC0

640 READ ERROR ON BOOTSTRAP FILE FFB0

641 DSKUTL <RF> ABORTED

642 FFC0 BOOTSTRAP VERSION FFC1 USED ON FFC2
DISK FFC3

643 FFC0 SECTOR(S) ARE NOT AVAILABLE ON FFC1
DISK FFC2

644 PLEASE RUN CHECK.DISK ON FFC0 DISK FFC1

645 DSKUTL <RF> OF FFC0 DISK <FFC1> AS FFC2 ON
FFC3 COMPLETED

646 DSKUTL <LIST> ABORTED

647 ILLEGAL SECTOR ADDRESS FFC0 FOR FFC1

648 FFC0 EXHAUSTED DURING SECTOR RANGE FFC1 -
FFC2

649 PRESENT INPUT DISK FIRST PLEASE

650 PRESENT OUTPUT DISK NOW PLEASE

651 FFC0 AND FFC1 HAVE DIFFERENT CAPACITY

652 DSKUTL <COPY> ABORTED

653 READ ERROR ENCOUNTERED ON FFC0 AT SECTOR
FFC1

654 WRITE ERROR ENCOUNTERED ON FFC0 AT SECTOR
FFC1

655 DSKUTL <COPY> FROM FFC0 TO FFC1
COMPLETED

657 SPECIFIED DISK-ID FFC0 FOR <COPY> IGNORED
- INPUT FFC1 DISK FFC2 HAS PPIT

658 ILLEGAL SECTOR RANGE FROM FFC0

726 PARAMETER SPECIFICATION FOR <RF>
INCOMPLETE

727 NO UNIT SLOTS AVAILABLE IN SYSTEM PPIT

728 DSKUTL <COPY> FROM FFC0 DISK FFC1 TO FFC2
DISK FFC3 COMPLETED

729 PRESENT DISK FOR LISTING PLEASE

730 SPECIFIED DISK-ID FFC0 FOR <COPY> IGNORED
- INPUT FFC1 DISK NOT IN CMS FORMAT

736 FFC0 IS CMS DISK FFC1 - CONFIRM <COPY>

738 DSKUTL <COPY> TO FFC0 NOT CONFIRMED

768 CAPACITY OF FFC0 DISK NOT RECOGNIZED -
FFC1 SECTORS

954 PARAMETER SPECIFICATION FOR <IN>
INCOMPLETE

955 FFC0 IS CMS DISK FFC1 - ENTER OK TO CONFIRM
<IN> REQUIRED

956 DSKUTL <IN> OF FFC0 NOT CONFIRMED

957 DSKUTL <IN> ABORTED

958 FFC0 WILL BE INITIALIZED AS PHYSICAL UNIT
FFC1

959 DSKUTL <IN> OF FFC0 DISK <FFC1> AS FFC2 ON
FFC3 COMPLETED

960 INITIALIZATION ERROR ON FFC0 AT SECTOR FFC1

961 INIT.TRACK COMMUNICATE NOT SUPPORTED

962 TEST PATTERN<S> NOT REQUIRED FOR FFC0
DISK

963 READY AND RESERVE REQUIRED DRIVE FOR DISK
INITIALIZE

1016 DSKUTL <FFC0> - ENTER NO. OF FILES
<NUMBER LESS THAN 2805>

1017 DSKUTL <FFC0> - ENTER SERIAL NO. <6
DIGITS>

1018 DSKUTL <FFC0> - ENTER DISK NAME <UP TO 7
LEGAL CHARACTERS>

1019 DSKUTL <FFC0> - ENTER OWNERS NAME <UP
TO 14 CHARACTERS. DELIMIT WITH " >

1020 DSKUTL <FFC0> - IS THIS DISK FOR MTR USE?
- <Y OR N>

1021 DSKUTL <FFC0> - ENTER REQUIRED BOOTSTRAP
FILE-NAME OR NULL IF FFB1 IS TO BE USED

1022 DSKUTL <IN> - ENTER NO. OF TEST PATTERNS
TO BE SPECIFIED <NUMBER 0-4>

1023 DSKUTL <IN> - ENTER FFC0 TEST PATTERN <20
CHARACTER MAX. DELIMIT WITH FFC1 IF HEX, "
IF ASCII>

1024 DSKUTL <RF> - IS DIRECTORY ON THIS DISK TO
BE IGNORED? - <Y OR N>

1025 FFC0 DEVICE TYPE NOT SUPPORTED

1026 BOOTSTRAP CODE NOT COPIED TO OUTPUT FFC0
DISK FFC1

1027 CANNOT RUN MTR ROUTINES ON OUTPUT FFC0
DISK FFC1

1028 FFB0 NOT COPIED TO OUTPUT FFC1 DISK FFC2

1029 AREAS POSSIBLY MISSING FROM OUTPUT FFC0
DISK FFC1

1030 OR "UNIT.FDA" IF DISK FOR FIXED ASSEMBLAGE

1070 FFC0 OF FFC1 SURFACE CHECK PASSES
COMPLETED ON FFC2

1182 FFB0 DOES NOT CONTAIN A COLDSTART
BOOTSTRAP

1200 COLDSTART REQUEST IGNORED - VALID ONLY
ON B 90 SYSTEM

1208 DISK FFC0 CANNOT BE USED FOR MTR PURPOSES

1201 FFC0 IS NOT A MINI DISK - CANNOT BE USED
FOR COLDSTART

1202 FUNCTION VALID ONLY ON B 90 AND B 900
SYSTEMS

1203 COLDSTART OPTION IS VALID ONLY ON B 90
SYSTEM

1204 DISK FFC0 WAS NOT INITIALIZED ON FFC1 SYSTEM

1205 DSKUTL <FFC0> - FFC1 BOOTSTRAP VERSION
FFC2 COPIED TO DISK FFC3

1206 DSKUTL <FFC0> - IS THIS DISK FOR COLDSTART
USE? - <Y OR N>
1207 DSKUTL <FFC0> ABORTED
1238 DSKUTL <FFC0> CONTINUING - READ ERROR
IGNORED

DUMPADISK

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
16 FFB0 REMOVED
25 INVALID CHARACTER IN IDENTIFIER FFC0 -
WARNING
51 FFB0 NOT LOADED - ALREADY ON DISK
52 ALTHOUGH WITH DIFFERENT ATTRIBUTES
53 FFB0 LOAD DISCREPANCY
54 FFB0 LOADED
55 FFB0 DUMPED
56 NO FILES TO DUMP
57 FILE FFB0 NOT DUMPED - HAS BEEN REMOVED
58 FILE FFB0 NOT DUMPED - IN OUTPUT USE
59 FILE FFB0 NOT DUMPED - HAS BEEN ALTERED
60 FFB0 DUMP DISCREPANCY
61 FFC0 FFC1 ON FFC2 FFC3 FOR FFC4
62 DUPLICATE - FFC0 ALREADY BEING DUMPED
63 NO FILES TO LOAD
64 FFC0 NOT DUMPED - DATA FILE NOT ONLINE
65 DUPLICATE - FFC0 ALREADY BEING LOADED
69 UTILITY LIMIT REACHED AT FILE FFC0
76 CONTINUING PROCESS
101 FILE LIST FOR FFC0 MAY BE INCOMPLETE
102 FILE LIST MAY BE INCOMPLETE
132 I/O ERROR DETECTED IN WRITE TO FILE FFB0
133 I/O ERROR DETECTED IN READ FROM FILE FFB0
134 ERROR AT RECORD NUMBER FFA0
194 READ ERROR ON PPIT OF FFC0
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
262 DIRECTORY ERROR ON DISK FFC0
290 OK TO PO DRIVE FFC0
291 INSUFFICIENT AREA TO DUMP TO ON DRIVE FFC0
292 FILE FFC0 NOT LOADED - FFC1 WHEN DUMPING
293 OVERFLOW DISK REQUIRED TO CONTINUE DUMP
294 DISK FFC0 DOES NOT BELONG TO THIS LOAD
295 OK TO PO DISK FFC0
296 FFB0 - DATA FILE NOT FOUND ON DISK FOR
LOAD
297 DISK FFC0 REQUIRED TO CONTINUE FFC1
298 DISK FFC0 IS NOT REQUIRED FOR THIS LOAD

299 MAKE DISK FFC0 AVAILABLE AND ENTER "OK"
 OR MAKE ALTERNATIVE DISK AVAILABLE AND
 ENTER NAME
 300 FFC0 IS NOT A RECOGNIZED BACKUP DISK
 503 DISK FFC0 LOCKED
 605 ENTER A FOUR CHARACTER DUMP
 IDENTIFICATION PREFIX
 612 SYSTEM DISK IS LOCKED
 614 READ ERROR ENCOUNTERED ON FFC0. RANGE
 FFA1 - FFA2
 615 OPEN DISK UNLABELLED NOT IMPLEMENTED
 616 DISK LABEL OF FFC0 NOT IN CMS FORMAT
 618 WRITE ERROR ENCOUNTERED ON FFC0. RANGE
 FFA1 - FFA2
 619 FFB0 NOT DUMPED DUE TO I/O ERROR
 622 FFB0 NOT LOADED DUE TO I/O ERROR
 943 FFB0 NOT AVAILABLE - DISK LOCKED

ECMA.LD
 863 NOT A BASIC ECMA TAPE
 864 UNEXPECTED END OF TAPE
 865 NOT A COMPACT ECMA TAPE
 866 INCORRECT TAPE NAME
 867 NOT FIRST PART OF FILE
 868 DATA BLOCK TOO LARGE - FFB0

FILEUTL
 1183 IRRECOVERABLE DATA COMM. ERROR ON FFC0
 RUNNING FILEUTL - PROGRAM TERMINATING
 1209 FFC0 CAN ONLY BE RUN AS A DATA COMM
 PROGRAM

FL
 1 INVALID DICTIONARY ENTRY FFA0
 2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE - IN USE
 6 NO SPECIFICATION GIVEN
 7 DISK FFC0 NOT AVAILABLE
 101 FILE LIST FOR FFC0 MAY BE INCOMPLETE
 102 FILE LIST MAY BE INCOMPLETE
 194 READ ERROR ON PPIT OF FFC0
 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 503 DISK FFC0 LOCKED
 612 SYSTEM DISK IS LOCKED

FS
 2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE - IN USE
 6 NO SPECIFICATION GIVEN
 7 DISK FFC0 NOT AVAILABLE
 151 RECORD SIZE TOO GREAT
 152 FFB0 SQUASHED FROM FFA1 TO FFA2 RECORDS
 153 KEYFILE FFC0 RECONSTRUCTED
 154 KEYFILE FFC0 SORT FAILURE - END OF JOB

155 ZIP TO SORT FAILURE – END OF JOB
 187 READ ERROR ON KFPB OF FFC0
 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 286 FFB0 FILE TYPE IS NOT DATA
 943 FFB0 NOT AVAILABLE – DISK LOCKED

ICMD

1 INVALID DICTIONARY ENTRY FFA0
 2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE – IN USE
 6 NO SPECIFICATION GIVEN
 7 DISK FFC0 NOT AVAILABLE
 16 FFB0 REMOVED
 25 INVALID CHARACTER IN IDENTIFIER FFC0 –
 WARNING
 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 253 FFB0 TO FFB1 COPIED
 503 DISK FFC0 LOCKED
 583 VOLUME FFC0 OF FFC1 OUT OF SEQUENCE
 584 DUPLICATE FILE FFB0
 585 NO FREE LABEL ON FFC0
 586 NO SPACE FOUND ON FFC0
 587 RECORD SIZE OF FFB0 EXCEEDS FFA1
 588 NEXT SEQUENCE NUMBER OF FFC0 EXCEEDS FFA1
 589 BAD INDEX TRACK ON FFC0 SECTOR FFA1
 590 INPUT ERROR ON FFB0 SECTOR FFA1
 591 OUTPUT ERROR ON FFB0 SECTOR FFA1
 592 ENTER VOL-ID FOR FFC0 FFC1
 593 FFC0 PURGED
 594 FILESIZE REQUIRED FOR FFB0
 612 SYSTEM DISK IS LOCKED
 994 CANNOT CREATE A FILE ‘DATA’ ON ICMD FFC0 –
 RESERVED NAME
 995 FILESIZE OF DISK FILE FFB0 TOO SMALL

INSTALL

1210 SPECIFIED OUTPUT DISK FFC0 IS NOT AVAILABLE
 1211 SPECIFIED OUTPUT DISK FFC0 IS LOCKED
 1212 SPECIFIED OUTPUT DISK FFC0 IS DUPLICATE
 1213 SYSTEM DISK FFC0 HAS DIRECTORY ERRORS
 1214 SYSTEM DISK FFC0 IS NOT AVAILABLE
 1215 FFB0 NOT COPIED – NOT FOUND
 1216 FFB0 NOT COPIED – IN USE
 1217 FFB0 COPIED
 1218 NOW POWER OFF SYSTEM DISK FFC0 AND RE-
 WARMSTART FROM DISK FFC1
 1219 PLEASE USE “COPY” TO COPY THESE FILES FROM
 SOME OTHER DISK TO DISK FFC0
 1220 BEFORE POWERING OFF DISK FFC0 AND
 WARMSTARTING FROM DISK FFC1
 1221 THE SYSTEM IS NOT A B 90
 1222 FFC0 NOT EXECUTED FROM THE SYSTEM DISK

1223 THE SYSTEM DISK FFC0 IS NOT A MINI
1224 SPECIFIED OUTPUT DISK FFC0 IS A MINI
1225 THE FOLLOWING FFC0 FILE(S) WERE NOT COPIED
-
1226 FFB0 NOT COPIED - READ ERROR ON DISK FFC1
1227 FFB0 NOT COPIED - WRITE ERROR ON DISK FFC1
1228 FFC0 TERMINATED
1229 FFB0 NOT COPIED - FILE IS A KEYFILE
1230 FFB0 NOT COPIED - CANNOT ALLOCATE ON DISK
FFC1
1231 DIRECTORY OF SPECIFIED DISK FFC0 IS FULL -
DISK SHOULD BE REFORMATTED FOR A LARGER
NUMBER OF FILES

KA

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
99 DISK FFC0 IS A PSEUDO-DISK
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
234 TABLE SIZE EXCEEDED
285 NO OUTPUT GENERATED BY KA FOR DISK FFC0
503 DISK FFC0 LOCKED
612 SYSTEM DISK IS LOCKED
943 FFB0 NOT AVAILABLE - DISK LOCKED

KEY.CHECK

2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
184 FFC0 <DATAFILE> NOT FOUND
185 FFC0 IS NOT A KEYFILE
186 FFC0 IS A NULL-KEY KEYFILE
187 READ ERROR ON KFPB OF FFC0
188 READ ERROR ON KEYFILE FFC0
189 UNABLE TO OPEN FFC0 <INDEXFILE>
190 FFC0 <DATAFILE> IN USE
191 FFC0 <DATAFILE> INCOMPATIBLE WITH KEYFILE
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
286 FFB0 FILE TYPE IS NOT DATA
503 DISK FFC0 LOCKED
555 WARNING - GENERATION NO. MISMATCH
BETWEEN FFB0 AND DATAFILE FFB1
563 FIRST DISCREPANCY FOUND BETWEEN KEYFILE
FFB0 AND DATAFILE FFB1
569 FFC0 DISCREPANCIE(S) FOUND WITH KEYFILE FFB1

576 FFC0 DISCREPANCIE(S) FOUND WITH DATAFILE
FFB1
580 KEY.CHECK ON FFB0 COMPLETED
581 KEY.CHECK COMMENCING ON DATAFILE FFB0
608 KEYFILE FFB0 IS CONSISTENT WITH DATAFILE
FFB1
609 GENERATION NO. OF KEYFILE FFB0 MODIFIED TO
FFC1
610 UNABLE TO MODIFY GENERATION NO. OF
KEYFILE FFB0
611 KEYFILE FFB0 IS NOT CONSISTENT WITH
DATAFILE FFB1
617 KEYFILE FFB0 PERTAINS TO DATAFILE FFB1
943 FFB0 NOT AVAILABLE - DISK LOCKED
1232 THE NUMBER OF MATCHED KEYFILE ENTRIES IS
NOT EQUAL TO
1233 THE NUMBER OF DATAFILE RECORDS WITH A
KEY IN KEYFILE

LD

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
16 FFB0 REMOVED
25 INVALID CHARACTER IN IDENTIFIER FFC0 -
WARNING
49 FFC0 NOT A RECOGNIZED DUMP TAPE
51 FFB0 NOT LOADED - ALREADY ON DISK
52 ALTHOUGH WITH DIFFERENT ATTRIBUTES
53 FFB0 LOAD DISCREPANCY
54 FFB0 LOADED
55 FFB0 DUMPED
56 NO FILES TO DUMP
57 FILE FFB0 NOT DUMPED - HAS BEEN REMOVED
58 FILE FFB0 NOT DUMPED - IN OUTPUT USE
59 FILE FFB0 NOT DUMPED - HAS BEEN ALTERED
60 FFB0 DUMP DISCREPANCY
61 FFC0 FFC1 ON FFC2 FFC3 FOR FFC4
62 DUPLICATE - FFC0 ALREADY BEING DUMPED
63 NO FILES TO LOAD
64 FFC0 NOT DUMPED - DATA FILE NOT ONLINE
65 DUPLICATE - FFC0 ALREADY BEING LOADED
66 FFB0 - DATA FILE NOT FOUND ON TAPE FOR
LOAD
67 LOAD ABANDONED - ERROR IN DUMP
68 TAPE SHOULD HAVE BEEN PURGED
69 UTILITY LIMIT REACHED AT FILE FFC0
101 FILE LIST FOR FFC0 MAY BE INCOMPLETE
102 FILE LIST MAY BE INCOMPLETE
123 END FFC0
192 DUMP ABANDONED - TAPE SHOULD BE PURGED
194 READ ERROR ON PPIT OF FFC0

195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 503 DISK FFC0 LOCKED
 612 SYSTEM DISK IS LOCKED
 943 FFB0 NOT AVAILABLE - DISK LOCKED
 1184 FILE FFB0 NOT DUMPED - IS INCOMPLETE
 1185 FILE FFB0 NOT LOADED - NOT ON TAPE - WAS
 IN USE AT DUMP TIME
 1186 FILE FFB0 NOT LOADED - NOT ON TAPE - HAD
 BEEN REMOVED AT DUMP TIME
 1187 FILE FFB0 NOT LOADED - NOT ON TAPE - HAD
 BEEN ALTERED AT DUMP TIME
 1188 FILE FFB0 NOT LOADED - NOT ON TAPE - DISK
 WAS LOCKED AT DUMP TIME
 1189 FILE FFB0 NOT LOADED - NOT ON TAPE - WAS
 INCOMPLETE AT DUMP TIME
 1190 FILE FFB0 NOT LOADED - NOT ON TAPE -
 REASON UNKNOWN
 1191 ZIP FAILURE WHEN ZIPPING CHECKADUMP FROM
 LD
 1192 ENTER "Y" TO RETRY ZIP, OR "N" TO
 TERMINATE
 1193 COMMUNICATE FAILURE
 1 INVALID DICTIONARY ENTRY FFA0
 2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 4 FFB0 NOT FOUND
 5 FFB0 NOT AVAILABLE - IN USE
 6 NO SPECIFICATION GIVEN
 7 DISK FFC0 NOT AVAILABLE
 151 RECORD SIZE TOO GREAT
 194 READ ERROR ON PPIT OF FFC0
 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 231 BAD ATTRIBUTES SPECIFIED
 232 FFB0 - DATA FILE NOT FOUND
 233 KEY FILE FFB0 OR DATA FILE IN USE
 235 KEY FILE FFB0 OR DATA FILE NOT FOUND
 236 FFB0 NOT ACCEPTABLE - RECORD SIZE OF FFA1
 EXCEEDS MAXIMUM FOR THIS RUN - RESUBMIT
 239 FFB0 EXHAUSTED DURING RANGE - FF01 - FF02
 240 FFB0 EXHAUSTED DURING RANGE - FF01 FFA2
 241 FFB0 EXHAUSTED DURING RANGE - FF01 EOF
 242 FFB0 EXHAUSTED DURING FFA1, FFA2
 243 FFB0 EXHAUSTED DURING FFA1, EOF
 244 SELECTIONS OF RECORDS FROM FFA0 IGNORED
 301 INVALID CHARACTER IN IDENTIFIER FFB0
 302 INVALID CHARACTER IN IDENTIFIER FFB0 -
 WARNING
 335 FFA0, FFA1 IN FFB2 NOT LISTED
 336 FFA0, EOF IN FFB1 NOT LISTED
 337 NO RECORDS FOR LISTING FROM FFB0

LIST

	338	PERMANENT ERROR ON INPUT FILE FFB0 - NO. FFA1
	339	PERMANENT ERROR ON INPUT FILE FFB0 - KEY FF01
	503	DISK FFC0 LOCKED
	943	FFB0 NOT AVAILABLE - DISK LOCKED
	964	KEYFILE FFB0 OR DATAFILE NOT AVAILABLE - DISK LOCKED
LOAD.VFU	1	INVALID DICTIONARY ENTRY FFA0
	2	ILLEGAL PARAMETER LIST FFC0
	3	INVALID CHARACTER IN IDENTIFIER FFC0
	4	FFB0 NOT FOUND
	5	FFB0 NOT AVAILABLE - IN USE
	6	NO SPECIFICATION GIVEN
	7	DISK FFC0 NOT AVAILABLE
	16	FFB0 REMOVED
	109	FFB0 IS NOT A RECOGNIZED SYSTEM FILE
	195	FFB0 IS NOT TYPE SOURCE OR DATA
	196	SPECIFICATION TOO LONG
	276	FFC0 ALREADY DEFINED IN SYSVFU.LIB
	277	FFC0 NOT DEFINED IN SYSVFU.LIB
	278	READ ERROR ON SYSVFU.LIB
	279	WRITE ERROR ON SYSVFU.LIB
	280	FFC0 DEFINED IN SYSVFU.LIB
	281	FFC0 DELETED FROM SYSVFU.LIB
	282	PRINTER LPFFC0 FORMATTED WITH FFC1
	283	PRINTER LPFFC0 NOT FORMATTED WITH FFC1
	503	DISK FFC0 LOCKED
	612	SYSTEM DISK IS LOCKED
LR	1	INVALID DICTIONARY ENTRY FFA0
	2	ILLEGAL PARAMETER LIST FFC0
	3	INVALID CHARACTER IN IDENTIFIER FFC0
	4	FFB0 NOT FOUND
	5	FFB0 NOT AVAILABLE - IN USE
	6	NO SPECIFICATION GIVEN
	7	DISK FFC0 NOT AVAILABLE
	99	DISK FFC0 IS A PSEUDO-DISK
	102	FILE LIST MAY BE INCOMPLETE
	194	READ ERROR ON PPIT OF FFC0
	195	FFB0 IS NOT TYPE SOURCE OR DATA
	196	SPECIFICATION TOO LONG
	234	TABLE SIZE EXCEEDED
	301	INVALID CHARACTER IN IDENTIFIER FFB0
	305	FILE LIST FOR FFB0 MAY BE INCOMPLETE
	503	DISK FFC0 LOCKED
	523	PLEASE RUN SQ ON FFC0
	612	SYSTEM DISK IS LOCKED
	943	FFB0 NOT AVAILABLE - DISK LOCKED
MAKEGUARDFL	1172	FFC0 SYNTAX IS :
	1178	FFC0 MUST RUN UNDER A USERCODE
MAKEUSER	1163	USERCODE FFC0 ALREADY EXISTS

MODIFY

1164 COPYING USERCODE FILE... PATIENCE
1165 USERCODE FFC0 ADDED
1166 USERCODE FFC0 NOT ADDED
1167 USERCODE FFC0 NOT FOUND
1168 USERCODE FFC0 DELETED
1169 USERCODE FFC0 IS PRIVILEGED
1170 USERCODE (FFC0) IS NOT PRIVILEGED
1171 PLEASE, ENTER YOUR NEXT COMMAND OR 'END'
1172 FFC0 SYNTAX IS :
1173 USERCODE FILE CORRUPTED - PROGRAM
ABORTED
1175 SYSUSERCODE CONTAINS :
1176 SYSUSERCODE LISTED
1177 NO USERCODE FOUND IN SYSUSERCODE FILE
1178 FFC0 MUST RUN UNDER A USERCODE
1180 USERCODE FFC0 NOT DELETED
306 ATTRIBUTE VALUE MISSING ... FF00
307 KEYWORD IN ERROR ... FF00
308 ATTRIBUTE-VAL INCONSISTENT ... FF00
309 INCORRECT ATTRIBUTE ... FF00
310 WARNING + + + DEVICE-MYUSE INCONSISTENT
311 WARNING + + + FILE.SIZE TOO LARGE
312 WARNING + + + TOO MANY BUFFERS
313 WARNING + + + REC NOT INTEGRAL OF BUF
314 CODE.FILE NAME IN ERROR ... FF00
315 FILE NAME NOT FOUND ... FF00
316 CURRENCY SYMBOL EXPECTED ... FF00
317 NUMERIC ATTRIBUTE-VAL REQD ... FF00
318 FILE NOT SPECIFIED ... FF00
319 PPB NOT SPECIFIED ... FF00
320 NOT INDEXED FILE ... FF00
321 NOT COBOL PROGRAM FILE ... FF00
322 NOT MPL PROGRAM FILE ... FF00
323 NOT FPB ATTRIBUTE ... FF00
324 NOT PPB ATTRIBUTE ... FF00
325 ILLEGAL FILE NAME ... FF00
326 MUST HAVE 0 < KEY.LENGTH < 29 ... FF00
327 NUMBER TOO BIG ... FF00
328 CANNOT OPEN THAT CODE.FILE ... FF00
329 CODE.FILE - BAD FILETYPE ... FF00
330 CODE.FILE IN USE ... FF00
1074 FILE NOT DATA OR INDEX TYPE - - - FF00
1075 FILETYPE NOT 0-15 OR \$80 - - - FF00
1076 DECIMAL NUMBER REQUIRED - - - FF00
PB 5 FFB0 NOT AVAILABLE - IN USE
829 FFB0 I/O ERROR
830 INVALID FILENAME
831 INVALID PACK-ID
832 FIELD TOO LARGE
833 NUMERIC FIELD NOT NUMERIC
834 FFB0 EOF ENCOUNTERED BEFORE FFA1 PAGE(S)
SKIPPED

835 FFB0 NOT SUCCESSFULLY OPENED
836 FFB0 FILE TYPE NOT PB
837 FFB0 HEADER RECORD NOT ASCII
838 FFB0 INVALID BLOCK SIZE
839 FFB0 INVALID RECORD SIZE
840 FFB0 HEADER RECORD PB NUMBER GREATER
THAN 65535
841 COPIES PARAMETER EQUALS ZERO
842 DUPLICATE OPTION
843 PAGE PARAMETER EQUALS ZERO
844 RECORD PARAMETER EQUALS ZERO
845 TOO MANY START-LENGTH PAIRS SPECIFIED
847 PB NUMBER GREATER THAN 65535
848 INVALID EQUALS PARAMETER
849 INVALID AUTO PARAMETER
850 AUTO OPTION REQUIRES CLOCK ON SYSTEM
851 INVALID ON PARAMETER
852 INVALID PB RECORD FORMAT
853 PARAMETER MUST BE NUMERIC
854 SPECIAL FORMS REQUIRED FOR FFB0
855 CREATED AS FFC0-FFC1 BY FFC2
856 AX INVALID - REQUIRES Y OR N
857 AX INVALID - REQUIRES OK
858 INVALID OPTION
859 INITIATING MESSAGE TOO LONG
860 REMOVE FORMS
861 INVALID PB PARAMETER
862 INVALID LIST PARAMETER
1121 AP JOURNAL PRINTER NEEDS OPERATOR
ATTENTION
991 MCS/DATA COMM SUBSYSTEM NOT AVAILABLE
992 INVALID STATION NAME
993 PRINTER DEVICE SPECIFIED IS NOT AVAILABLE
1234 DATA COMM ERROR - PB TERMINATING
1235 ERROR CONDITION REPORTED BY FFC0
1236 PRINTER DEVICE FFC0 IS NOT AVAILABLE
1237 FFB0 EOF ENCOUNTERED BEFORE FFA1 RECORD(S)
SKIPPED

PL

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
16 FFB0 REMOVED
71 ILLEGAL FILE NAME FFC0
76 CONTINUING PROCESS
79 ILLEGAL VALUE FFC0
80 ILLEGAL OPTION FFC0
81 NO TIME SPECIFICATION GIVEN AFTER FFC0
82 NO FFC0 NUMBER GIVEN

83 FFC0 OPTION ALREADY SET
84 ILLEGAL FFC0 SPECIFICATION
85 FFB0 IS NOT A LOG FILE
108 INVALID REQUEST FFC0
120 NO FILE NAME IN PARAMETERS
130 **** UTILITY LIMIT REACHED ****
132 I/O ERROR DETECTED IN WRITE TO FILE FFB0
133 I/O ERROR DETECTED IN READ FROM FILE FFB0
134 ERROR AT RECORD NUMBER FFA0
135 NO ENTRIES FOUND WITHIN PARAMETERS
136 UNRECOGNIZED DEVICE IN ERROR ENTRY
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
288 OUTPUT FILE FFB0 TOO SMALL
503 DISK FFC0 LOCKED
606 SCREEN SIZE IS TOO SMALL TO RUN UTILITY
607 ENTER FF00 TO FINISH OR FF01 TO CONTINUE
613 FFA0 RECORDS WRITTEN TO BACKUP FILE FFB1
1062 WARNING - FILE FFB0 MAY NOT BE PURGED
1064 WARNING - ENTRY OPTION NOT ALLOWED FOR
FILE FFB0
1065 DEVICE FFC0 HAS BEEN SELECTED ALREADY
1066 UNRECOGNIZED DEVICE IDENTIFICATION FFC0
1067 NO DEVICE(S) FOLLOWING DEVICE SPECIFICATION
1068 DEVICE FFC0 NOT FOUND IN HARDWARE
CONFIGURATION AREA

PPID

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
7 DISK FFC0 NOT AVAILABLE
99 DISK FFC0 IS A PSEUDO-DISK
193 READ ERROR ON LABEL OF FFC0
194 READ ERROR ON PPIT OF FFC0
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
272 NO PPIT ON SYSTEM
273 FF00 - PHYS. UNIT FFC1 FFC2
274 FF00 - PSEUDO ON FFC1
275 FF00 - PSEUDO UNRESTRICTED
302 INVALID CHARACTER IN IDENTIFIER FFB0 -
WARNING
612 SYSTEM DISK IS LOCKED
731 NO PPIT ON DISK FFC0
732 DISKS IN PPIT OF DISK FFC0 -
1073 FF00 - PSEUDO UNRESTRICTED/SYSTEM

PW

1160 OLD PASSWORD INVALID
1161 PASSWORD CHANGED
1162 PASSWORD NOT CHANGED
1172 FFC0 SYNTAX IS :
1178 FFC0 MUST RUN UNDER A USERCODE

RB

1089 EXPECTED 'BEFORE' AFTER 'DETAIL'

1090 EXPECTED COMMA OR 'BEFORE' AFTER FILEID
 1091 EXPECTED COMMA, 'DETAIL' OR 'BEFORE' AFTER
 FILEID
 1092 EXPECTED INTEGER AS SECOND HALF OF BACKUP
 NUMBER RANGE
 1093 EXPECTED INTEGER AS SECOND HALF OF DELETE
 RANGE
 1094 BACKUP NUMBER RANGE UPPER BOUND LESS
 THAN LOWER BOUND
 1095 DELETE RANGE UPPER BOUND LESS THAN LOWER
 BOUND
 1096 FFC0 SYNTAX ERROR
 1097 TOKEN SCANNED WAS: FFC0
 1098 'FFC0 HELP' FOR INFO
 1099 TOO MANY DISKS SPECIFIED
 1100 INVALID PACK OR BACKUP FILE NAME
 1101 TOO MANY FILES SPECIFIED
 1102 EXPECTED INTEGER FOR MONTH IN DATE
 1103 EXPECTED INTEGER FOR DAY IN DATE
 1104 INTEGER FOR MONTH OUT OF RANGE
 1105 INTEGER FOR DAY OUT OF RANGE
 1106 INTEGER FOR MONTH MUST BE GREATER THAN
 ZERO
 1107 MALFORMED DATE FIELD
 1108 EXPECTED 2 DIGIT INTEGER FOR YEAR
 1109 GARBAGE FOUND AT END OF GOOD COMMAND
 1110 FFB0 NOT REMOVED - OPEN DENIED
 1111 FFB0 NOT REMOVED - CREATED AFTER
 SPECIFIED DATE
 1112 FFB0 NOT REMOVED - PURGE FAILED
 1113 FFB0 NOT REMOVED - NOT FOUND
 1114 FFB0 NOT REMOVED - IN USE
 1115 NO BACKUP FILES FOUND IN RANGE FFC0-FFC1
 1116 NO BACKUP FILES FOUND IN RANGE FFC0-FFC1
 ON FFC2
 1117 COULD NOT FIND BACKUP FILE FFB0
 1118 NO BACKUP FILES FOUND IN FAMILY FFB0
 1119 NO BACKUP FILES FOUND ON DISK FFC0
 1120 FILEID, EQUALSIGN OR INTEGER EXPECTED TO
 IDENTIFY FILE(S)

RL 1 INVALID DICTIONARY ENTRY FFA0
 2 ILLEGAL PARAMETER LIST FFC0
 3 INVALID CHARACTER IN IDENTIFIER FFC0
 6 NO SPECIFICATION GIVEN
 615 OPEN DISK UNLABELLED NOT IMPLEMENTED
 616 DISK LABEL OF FFC0 NOT IN CMS FORMAT
 627 CANNOT READ SECTOR 0 OF FFC0
 731 NO PPIT ON DISK FFC0
 733 LABEL OF FFC0 DISK IS FFC1 - NOT FFC2
 734 FFC0 IS FIXED DISK FFC1 WITH PPIT
 735 FFC0 DISK FFC1 RELABELLED AS FFC2

SAU.INIT 2 ILLEGAL PARAMETER LIST FFC0

3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
195 FFB0 IS NOT TYPE SOURCE OR DATA
196 SPECIFICATION TOO LONG
503 DISK FFC0 LOCKED
634 BOOTSTRAP FILE FFB0 NOT CMS STANDARD
640 READ ERROR ON BOOTSTRAP FILE FFB0
717 DATA SPACE NOT AVAILABLE ON FFC0.
921 FFB0 IS VERSION FFC1
919 FFB0 CREATED
920 FFB0 NOT CREATED
922 SAU.INIT DEFINE PARAMETERS INCOMPLETE
923 PARAMETERS IN FILE FFB0 - FFC1 DISK FFC2 AS
FFC3 FOR FFC4 FILES,
924 SERIAL NO. FFC0, OWNER FFC1, DATE FFC2
925 READ ERROR ON FFB0
926 WRITE ERROR ON FFB0
927 FFB0 NOT CMS STANDARD
7 DISK FFC0 NOT AVAILABLE
659 READING ALL FFA0 SECTORS ON FFC1.
660 READ ERROR ON FFC0, SECTOR FFD1.
661 ENTER DISK ID TO CONTINUE.
662 ENTER SPACE TO TERMINATE DCR
663 READING CYLINDERS ABORTED.
664 BUFFERED BY SECTOR.
665 MULTI-SECTOR BUFFERED.
666 READING ALL SECTORS ABORTED.
667 READING ALL SECTORS COMPLETED.
668 FFA0 BAD SECTORS READ ON FFC1.
669 SECOND DISK TO BE TESTED? <Y OR N>
670 ERROR ON FFC0 MULTI-SECTOR READ.
671 FFA0 SECTORS READ.
672 READ ERROR ON FFC0 AVAILABLE TABLE.
673 DCR TERMINATED BY USER.
674 NON-NUMERIC INPUT CHARACTER.
675 MULTI-SECTOR BUFFERING NOT AVAILABLE.
676 MULTI-SECTOR BUFFERING WANTED? <Y OR N>
677 FFC0 AVAIL.TABLE WILL NOT BE CHECKED.
678 MORE THAN 60 BAD AREAS IN AVAILABLE TABLE.
679 NOT LISTED AS BAD IN AVAIL.TABLE.
680 LISTED AS BAD IN AVAIL.TABLE.
681 CAPACITIES ON 2ND DISK NOT SAME AS ON 1ST.
682 NO ERROR ON MULTI-SECTOR READ.
683 ERROR READING FDNL ON FFC0.
684 OPEN.FILES TRUNCATED.
685 FFA0 OPENS TRIED ON FFC1.
686 FFC0 ON FFC1 DID NOT OPEN.
687 FFC0 ON FFC1 OPENED.
688 FFA0 SUCCESSFUL, FFA1 UNSUCCESSFUL.

SCR

689 OPENING UP TO 20 FILES ON FFC0.
690 READING FFA0 CYLINDERS ON FFC1.
691 READING CYLINDERS COMPLETED.
692 ENTER TEST NUMBERS OR ENTER HELP.
693 ENTER ON 1 LINE NUMBERS FOR TEST WANTED.
694 1 = READ.ALL
695 2 = OPEN.FILES
696 3 = READ.CYLINDERS
697 4 = WRITE.INDEX
698 5 = READ.RANDOM
699 ALL = ALL TESTS WANTED
700 SPACE = TERMINATE DCR
701 NO FILES ON FFC0.
702 BUT NO ERRORS WHEN SECTORS READ.
703 WRITING INDEX FILE ON FFC0.
704 WRITING INDEX FILE FINISHED.
705 READING FFA0 RANDOM SECTORS ON FFC1.
706 READING RANDOM SECTORS COMPLETED.
708 ENTER NUMBER OF RANDOM SECTORS TO BE
READ FROM EACH DISK.
709 INPUT NUMBER > 16,777,215.
710 ENTER NUMBER OF RECORDS IN INDEX FILE.
711 DCR FINISHED.
712 OR SPACE TO CANCEL TEST.
713 REQUESTED IX FILE SIZE EXCEEDS DISK
CAPACITY
714 FETCH.VALUE = FFD0.
715 OPEN FAILED ON FFC0.
716 WRITE.INDEX ABORTED.
717 DATA SPACE NOT AVAILABLE ON FFC0.
718 INDEX FILE TRUNCATED.
719 READING RANDOM SECTORS ABORTED.
720 DATA MISMATCH ON FFC0, RECORD FFD1.
721 READING INDEX COMPLETED.
722 READING INDEX ABORTED.
723 FFA0 DATA MISMATCHES ON FFC1.
724 WRITE ERROR ON FFC0, KEY = FF01
737 READ ERROR ON FFC0, KEY = FF01.
746 SCR TERMINATED BY USER.
747 INVALID INPUT CHARACTER.
748 1 = DISK CONFIDENCE ROUTINE (DCR)
749 2 = CPU/IO INTERACTION TEST (CPU.IO)
750 3 = MAGNETIC DEVICE EXERCISER (MT.IO)
751 4 = NON.MAGNETIC PERIPHERAL EXERCISER
(OTHER.IO)
752 ENTER NUMBER OF ITERATIONS FOR CPU.IO –
DEFAULT = 1.
754 LINE CONTROL = LINE(1).
767 READY SCRATCH TAPES ON FFA0 MAGNETIC TAPE
DRIVES, FFA1 CASSETTE DRIVES.
755 ENTER NUMBER OF MT DEVICES – DEFAULT = 1.
756 ENTER NUMBER OF CS DEVICES – DEFAULT = 0.

757 ENTER NUMBER OF RECORDS IN FILE - DEFAULT
= 256.
758 ENTER NUMBER OF MT.IO ITERATIONS -
DEFAULT = 1
759 SPECIFY MT.IO PARAMETERS - ENTER SPACE FOR
DEFAULT VALUE.
760 ERROR: MORE THAN 5 DEVICES REQUESTED.
761 FILE FFA0 DID NOT OPEN FOR OUTPUT ON FFC1
762 FILE FFA0 DID NOT OPEN FOR INPUT ON FFC1
763 WRITE FAIL ON FFC0 FILE FFB1, RECORD FFA2
764 READ FAIL ON FFC0 FILE FFB1, RECORD FFA2
765 WRITE/READ MISMATCH ON FFC0 FILE FFB1,
RECORD FFA2
766 SCR MT.IO; ITERATION FFA0:
869 SCR FINISHED.
870 SCR OTHER.IO: ITERATION FFA0, WRITE PRINTER
871 LINE CONTROL = PAGE.
872 LINE CONTROL = LINE(1) BEFORE.
873 LINE CONTROL = LINE(1) AFTER.
874 ENTER NUMBER OF OTHER.IO ITERATIONS -
DEFAULT = 1.
875 PRINTER TO BE TESTED? <Y OR N>
876 CARD READER/PUNCH TO BE TESTED? <Y OR N>
877 PRINTER TEST FINISHED.
878 PRINTER TEST ABORTED.
879 ENTER SPACE FOR DEFAULT VALUES OR 0 TO
TERMINATE SCR
880 OTHER.IO FINISHED.
881 CPU.IO FINISHED.
882 MT.IO FINISHED.
883 SECOND DISK NOT WANTED.
884 REPLY NOT "Y" OR "N".
885 MULTI-SECTOR BUFFERING NOT WANTED.
1078 CARD PERIPHERALS NOT IMPLEMENTED
1088 OTHER.IO - 1 ITERATION
1079 DUPLICATE SCR SUBTEST
1080 ERROR: TOTAL MAGNETIC DEVICES = 0
1081 ERROR: FILESIZE = 0
1082 WARNING: SECOND DISK.ID = FIRST DISK.ID
1083 DEFAULT VALUES ARE: TESTS 1,2,4
1084 DCR - ALL SUBTESTS, 1 DISK
1085 INDEX FILESIZE = 100 RECORDS
1086 READING 1000 RANDOM SECTORS
1087 CPU.IO - 1 ITERATION
1122 SCR ABORTED
1123 SCR SUBTEST NAME EXPECTED
1124 0 IS INVALID PARAMETER
1125 TRUNCATED
1126 INVALID KEYWORD
1127 INVALID PARAMETER
1128 KEYWORD EXPECTED: FFC0
1129 LAST KEYWORD: FFC0
1130 NO. OF SUBTEST ITERATIONS EXPECTED

1131 DISK ID EXPECTED
 1132 SCR PARAMETER PART EXCEEDS 40 CHARACTERS
 1133 SCR INITIATING MESSAGE ERROR
 1134 DCR SUBTEST NAME EXPECTED
 1135 DUPLICATE DCR SUBTEST
 1136 INDEX FILE SIZE EXPECTED
 1137 NUMBER OF RANDOM RECORDS EXPECTED
 1138 'M' OR 'S' EXPECTED
 1139 ENTER NUMBER OF DCR ITERATIONS – DEFAULT
 = 1
 1151 DCR ABORTED

SCR.MPL

714 FETCH.VALUE = FFD0.
 739 P1 COMPUTATION ERROR. I = FFA0
 740 SYSTEM DISK NOT AVAILABLE.
 741 SCR.MPL COPY FFC0 TERMINATED.
 742 WRITE CONDITIONAL FAILS = FFA0, WRITE
 ERRORS = FFA1.
 743 READ CONDITIONAL FAILS = FFA0, READ ERRORS
 = FFA1.
 744 INPUT/OUTPUT MISMATCH, I = FFA0
 745 SCR.MPL COPY FFC0.
 753 TERMINATED.UNABLE TO WRITE 1ST RECORD.

SECURITY

1159 FFC0 IS FFC1 FFC2
 1172 FFC0 SYNTAX IS :
 1178 FFC0 MUST RUN UNDER A USERCODE

SORT-MERGE

197 SORT-MERGE SPEC WARNING= >FFC0
 198 SORT-MERGE SPEC ERRORS= >FFC0
 199 SORT-MERGE INTRINSIC NOT INVOKED
 200 MISSING SORT-MERGE SPECIFICATION
 201 ILLEGAL SORT-MERGE SPECIFICATION
 202 SORT-MERGE SPEC FILE NOT DATA OR SOURCE
 203 SORT-MERGE SPEC FILE RECORD SIZE TOO BIG
 204 FIX AND REPLY "OK" ELSE <NULL>
 205 FFB0 UNAVAILABLE ON DISK
 206 FILE FFC0 UNAVAILABLE
 207 FILE FFC0 UNAVAILABLE ON SYSTEM DISK
 208 SORT-MERGE SPEC CORRECTION APPLIED
 209 SORT-MERGE INTRINSIC ZIP FAILURE
 210 INTRINSIC CODE FILE NOT FOUND
 211 INSUFFICIENT MEMORY TO LOAD INTRINSIC
 212 NO DISK SPACE FOR INTRINSIC VM FILE
 213 NO MIX SLOT AVAILABLE FOR INTRINSIC
 214 INTRINSIC USERCOUNT ERROR
 215 ERROR ON DISK CONTAINING INTRINSIC
 216 INTRINSIC CODE FILE CORRUPT
 217 INTRINSIC DS/DP'ED

SORTINTRINS

886 FF00/SORTINTRINS [178] SORT-MERGE OUTPUT FILE
 NOT CREATED
 887 FF00/SORTINTRINS [188] INITIATING MESSAGE
 INVALID

888 FF00/SORTINTRINS [189] SORT-MERGE VER FFC1
 INITIATED FROM FF02/FFC3
 889 FF00/SORTINTRINS [182] NO INITIATING MESSAGE
 890 FF00/SORTINTRINS [193] INPUT RECORD SIZES
 UNEQUAL – BAD FILE FFC1
 891 FF00/SORTINTRINS [194] IN/OUT RECORD SIZES
 UNEQUAL – OUTPUT SIZE CHANGED
 892 FF00/SORTINTRINS [170] DUPLICATE RECORD FFA1
 893 FF00/SORTINTRINS [184] FILE ERROR FFA1 NEAR
 RECORD FFA2 ON FFC3
 894 FF00/SORTINTRINS [180] SORT-MERGE SOFTWARE
 ERROR
 895 FF00/SORTINTRINS [174] FFA1 RECORDS
 CONTAINING INVALID INDEX KEYS
 896 FF00/SORTINTRINS [444] KEYBUILD NOT FOUND
 897 FF00/SORTINTRINS [446] ALL INPUT RECORDS
 DELETED
 898 FF00/SORTINTRINS [173] FFA1 DUPLICATE RECORDS
 899 FF00/SORTINTRINS [187] DUPLICATE RECORDS –
 KEYFILE NOT BUILT
 900 FF00/SORTINTRINS [196] KEY OVERRUNS RECORD
 END
 901 FF00/SORTINTRINS [185] UNORDERED MERGE INPUT
 FILE FFC1 NEAR RECORD FFA2
 902 FF00/SORTINTRINS [176] FFA1 RECORDS MERGED
 903 FF00/SORTINTRINS [177] FFA1 FILES MERGED
 904 FF00/SORTINTRINS [183] FFA1 RECORDS SORTED
 905 FF00/SORTINTRINS [181] FFA1 RECORDS
 REFERENCED BY KEYFILE/TAGFILE
 906 FF00/SORTINTRINS [175] FFA1 DELETED RECORDS
 907 FF00/SORTINTRINS [198] FFA1 PARITY BLOCKS
 908 FF00/SORTINTRINS [445] MERGER NOT FOUND
 909 FFB0 HEADER RECORD DEVICE TYPE INVALID
 911 FF00/SORTINTRINS [171] ILLEGAL INDEX KEY IN
 RECORD FFA1
 912 FF00/SORTINTRINS [172] RECORDS LOST/GAINED BY
 SORT-MERGE
 913 FF00/SORTINTRINS [179] SORT-MERGE ABNORMAL
 EOJ
 914 FF00/SORTINTRINS [186] TOO MANY RECORDS FOR
 SORT-MERGE
 915 FF00/SORTINTRINS [195] BAD RECORD/BLOCK SIZE
 FOR OUTPUT DEVICE
 916 FF00/SORTINTRINS [197] CANNOT SPLIT INDEX FILE
 917 FF00/SORTINTRINS [454] LOC: FFD1 , FETCH.VALUE
 FFD2
 918 FF00/SORTINTRINS [455] FILE ERROR FFA1 ON
 SORT-MERGE FILE FFC2
 769 *** SQUASH VERSION FFC0 ***
 770 NON-FILE DIRECTORY FULL – “PARTIAL”
 SQUASH REQUIRED
 771 DISK INTEGRITY SUSPECT

SQ

772 USAGE MAP OF THE DISK WILL BE PRINTED
773 SQ ABORTED - I/O ERROR AT DISK ADDRESS
FFD0
774 SQ INVALID - NO INITIATING MESSAGE
775 SQ INVALID - INVALID PACK-ID
776 SQ INVALID - INVALID SYNTAX
777 FAST OPTION - INTEGER MUST NOT BE GREATER
THAN 65535
778 SQ INVALID - INVALID ADDRESS
779 OVERLAP - NO WAY TO SEPARATE THEM, SAVE
AND/OR PURGE AND AT LEAST RERUN SQ VERIFY
780 OVERLAP - THERE IS NO WAY TO SEPARATE
THEM, SAVE ONE OR BOTH, PURGE AND AT
LEAST RERUN SQ VERIFY
781 SQ ABORTED - REQUESTED AREA ALREADY
EXISTS
782 SQ ABORTED - INVALID DISK ALLOC.UNIT = 0
783 IT MAY BE CORRUPTED
784 SQ INVALID - SPECIFIED DISK NOT AVAILABLE
785 SQ NEEDS A PRINTER-DO YOU WANT TO
CONTINUE WITH NO PRINTOUT <Y OR N>
786 SQ ABORTED - NOTHING TO SQUASH IN THAT
AREA
787 IRRECOVERABLE ERROR ON DISK - SAVE AND/OR
REINITIALIZE
788 SQ ABORTED - NO WAY TO GET REQUIRED AREA
789 DIRECTORY DESTROYED - REINITIALIZE THE
DISK
790 ADDRESS MISMATCH - SAVE AND REINITIALIZE
THE DISK
791 TOO MANY DATA AREAS ON DISK - "PARTIAL"
SQ REQUIRED
792 MEMORY INCONSISTENCY OR SOME OTHER
IRRECOVERABLE PROBLEMS - RERUN SQ
793 TOO MANY FILES OPEN OR/AND BAD AREAS -
NO WAY TO SQUASH THIS DISK
794 CANNOT SQUASH TOO FRAGMENTED DISK -
PROGRAM LIMITATION
795 SQ ABORTED - SPECIFIED PACK-ID IS NOT
PHYSICAL UNIT
796 SQ ABORTED - SPECIFIED PACK-ID NOT FOUND
IN PPIT
797 I/O ERROR ON FFC0
798 THIS AREA SKIPPED - EXECUTION CONTINUES
799 BETWEEN DISK ADDRESSES FFD0 TO FFD1
800 READ ERROR WHEN MOVING FILE AREA
801 WRITE ERROR IN NEW LOCATION WHEN MOVING
FILE AREA
802 LAST SQ-EXECUTION WAS ABNORMALLY
TERMINATED WHEN MOVING FILE - FFC0
803 INTEGRITY OF FILE SUSPECT - EXECUTION
CONTINUES
804 AREAS STILL MISSING - RERUN SQ VERIFY

807 BAD SECTORS AND FFC0
808 FFC0 AND BAD SECTORS
809 LARGEST AVAILABLE SPACE IS FFA0 SECTORS
810 TOTAL AVAILABLE SPACE IS FFA0 SECTORS IN
FFA1 AREA(S)
811 *** SQ COMPLETED ***
812 *** VALIDATION PHASE BEGINS ***
813 *** SQUASH PHASE BEGINS ***
828 FFC0 AND FFC1

SYCOPY

2 ILLEGAL PARAMETER LIST FFC0
49 FFC0 NOT A RECOGNIZED DUMP TAPE
130 **** UTILITY LIMIT REACHED ****
138 COMPARISON ERROR ON FFB0
142 AROUND RECORD FFA0
145 DIFFERING RECORD SIZES
814 *** SYCOPY VERSION FFC0 ***
815 *** INITIATING MESSAGE FOR SYCOPY ***
816 INVALID CASE - SYCOPY ABORTED
817 FFA0 TAPE(S) COPIED
818 FFA0 TAPE(S) COMPARED
819 FFA0 TAPE(S) COPIED AND COMPARED
820 FFA0 TAPE(S) MERGED
821 NO ERROR
822 FFA0 TAPE(S) DELETED
823 SYCOPY ABORTED
824 INITIATING MESSAGE INCOMPLETE
825 DISCREPANCY IN DUMP LEVELS
826 END OF FILE
827 NOT END OF FILE

SYS-SUPERUTL

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
5 FFB0 NOT AVAILABLE - IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
8 FFC0 CONTAINS -
9 FFC0 ON FFC1 CONTAINS -
10 FFC0 FFC1 FFC2
11 FFC0 REQUIRES OVERFLOW DISK FFC1
12 ON LINE FFC0 FFC1
13 NOT ON LINE FFC0
14 NO FILES FOUND IN THE FAMILY FFC0
15 FFB0 IS A SYSTEM FILE
16 FFB0 REMOVED
17 INDEXED PAIR FFB0 FFB1 REMOVED
18 INDEXED PAIR FFB0 FFB1 NOT REMOVED -
19 DATA FILE FFB0 NOT FOUND
20 DATA FILE FFB0 IN USE
21 FFC0 NOT REMOVED - NOT FOUND
22 FFC0 NOT REMOVED - IN USE
23 FFB0 NOT REMOVED - SYSTEM FILE
24 NO FILES FOR REMOVAL IN FAMILY FFC0

25 INVALID CHARACTER IN IDENTIFIER FFC0 –
WARNING

26 FFC0 CHANGED TO FFC1

27 FFC0 NOT CHANGED – NOT FOUND

28 FFC0 NOT CHANGED – IN USE

29 FILE IDENTIFIER TOO LONG FFC0

30 FFC0 NOT CHANGED – ILLEGAL REQUEST

31 FFC0 NOT CHANGED – FFC1 ALREADY ON DISK

32 KEY FILE FFC0 NOW POINTS TO DATA FILE FFC1

33 NO FILES FOR CHANGING IN THE FAMILY FFC0

34 FFC0 IS AN INVALID REQUEST

35 FFC0 – FFA1 SECTORS; AVAILABLE FFA2;
TEMPORARY FFA3

36 FFC0 – FFA1 SECTORS; AV. FFA2; TEMP. FFA3

37 FFC0 IS AN UNACCEPTABLE RESPONSE FOR KX

38 TL ZIP FAIL – LOG FILES NOT CONSOLIDATED

39 LOGGING WILL NOT BE SWITCHED ON

40 UNABLE TO OPEN MINIMUM LOG FILES

41 UNABLE TO ZIP FFC0 AT WARMSTART

42 IR REQUIRED BEFORE FFC0 CAN BE DONE

86 PARITY ON RECORD FFA0 IN LOG FILE FFC1

87 CANNOT LOCATE DESIRED LOG ENTRY

88 CANNOT READ RECORD FFA0 IN LOG FILE FFC1

89 RECALL ENTRY TOO LONG – TRUNCATED

90 UNABLE TO OPEN FILE FOR FFC0

91 CANNOT RECALL <FFC0> BEYOND THIS POINT

92 PSEUDO-DISK FFC0 ON DISK FFC1 NOT AVAILABLE

93 PSEUDO-DISK FFC0 ON DISK FFC1 CONTAINS –

94 FFC0 ON PSEUDO-DISK FFC1 ON DISK FFC2
CONTAINS –

95 FFC0FFC1 – FFA2 SECTORS; AVAILABLE FFA3;
TEMPORARY FFA4

96 FFC0FFC1 – FFA2 SECTORS; AV. FFA3; TEMP. FFA4

97 UTILITY LIMIT REACHED ON PSEUDO-DISK FFC0

98 LIMIT AT CHANGE OF FFC0 TO FFC1

99 DISK FFC0 IS A PSEUDO-DISK

100 FFC0 ON PSEUDO-DISK FFC1 INCOMPLETE

101 FILE LIST FOR FFC0 MAY BE INCOMPLETE

102 FILE LIST MAY BE INCOMPLETE

103 COMMENCING LOG FILE CONSOLIDATION

104 COMMENCING LOG FILE REALLOCATION

105 LOGGING IS INITIATED ON FFC0 FFC1 FFC2

109 FFB0 IS NOT A RECOGNIZED SYSTEM FILE

123 END FFC0

132 I/O ERROR DETECTED IN WRITE TO FILE FFB0

194 READ ERROR ON PPIT OF FFC0

195 FFB0 IS NOT TYPE SOURCE OR DATA

196 SPECIFICATION TOO LONG

503 DISK FFC0 LOCKED

604 INSUFFICIENT DISK SPACE TO CONSOLIDATE LOG
FILES

612 SYSTEM DISK IS LOCKED

943 FFB0 NOT AVAILABLE – DISK LOCKED

	940	READ ERROR ON KEYFILE FFB0 - KEYFILE/ DATAFILE PAIR NOT REMOVED
TAPELR	1069	INVALID IMPLEMENTATION LEVEL OF FFC0
	1	INVALID DICTIONARY ENTRY FFA0
	2	ILLEGAL PARAMETER LIST FFC0
	3	INVALID CHARACTER IN IDENTIFIER FFC0
	4	FFB0 NOT FOUND
	5	FFB0 NOT AVAILABLE - IN USE
	6	NO SPECIFICATION GIVEN
	7	DISK FFC0 NOT AVAILABLE
	49	FFC0 NOT A RECOGNIZED DUMP TAPE
	195	FFB0 IS NOT TYPE SOURCE OR DATA
	196	SPECIFICATION TOO LONG
	943	FFB0 NOT AVAILABLE - DISK LOCKED
TAPEPD		
	1	INVALID DICTIONARY ENTRY FFA0
	2	ILLEGAL PARAMETER LIST FFC0
	3	INVALID CHARACTER IN IDENTIFIER FFC0
	4	FFB0 NOT FOUND
	5	FFB0 NOT AVAILABLE - IN USE
	6	NO SPECIFICATION GIVEN
	7	DISK FFC0 NOT AVAILABLE
	10	FFC0 FFC1 FFC2
	49	FFC0 NOT A RECOGNIZED DUMP TAPE
	50	FFC0 TAPE FFC1 <FFC2> DUMPED ON FFC3 FFC4 CONTAINS -
	123	END FFC0
	195	FFB0 IS NOT TYPE SOURCE OR DATA
	196	SPECIFICATION TOO LONG
	943	FFB0 NOT AVAILABLE - DISK LOCKED
TL		
	1	INVALID DICTIONARY ENTRY FFA0
	2	ILLEGAL PARAMETER LIST FFC0
	3	INVALID CHARACTER IN IDENTIFIER FFC0
	4	FFB0 NOT FOUND
	5	FFB0 NOT AVAILABLE - IN USE
	7	DISK FFC0 NOT AVAILABLE
	16	FFB0 REMOVED
	72	NO LOG FILES FOUND FOR CONSOLIDATION
	73	NO ACTIVE LOG FILE FOUND
	74	TRANSFER COMPLETED
	75	NO READY-TO-TRANSFER FILES FOUND
	76	CONTINUING PROCESS
	77	ILLEGAL USE OF RECOVER PARAMETER
	78	ACTIVE FILE NOT CONSOLIDATED
	83	FFC0 OPTION ALREADY SET
	110	FILE FFB0 HAS BEEN ALTERED
	132	I/O ERROR DETECTED IN WRITE TO FILE FFB0
	133	I/O ERROR DETECTED IN READ FROM FILE FFB0
	134	ERROR AT RECORD NUMBER FFA0
	195	FFB0 IS NOT TYPE SOURCE OR DATA
	196	SPECIFICATION TOO LONG
	221	SECONDARY LOG FILE FFB0 CREATED
	503	DISK FFC0 LOCKED

604 INSUFFICIENT DISK SPACE TO CONSOLIDATE LOG
FILES
612 SYSTEM DISK IS LOCKED
656 CANNOT OPEN FILE FFB0 – DIRECTORY FULL
943 FFB0 NOT AVAILABLE – DISK LOCKED
1069 INVALID IMPLEMENTATION LEVEL OF FFC0
1077 INVALID HARDWARE CONFIGURATION AREA IN
FILE FFB0

TRANSLATOR

1 INVALID DICTIONARY ENTRY FFA0
2 ILLEGAL PARAMETER LIST FFC0
3 INVALID CHARACTER IN IDENTIFIER FFC0
4 FFB0 NOT FOUND
5 FFB0 NOT AVAILABLE – IN USE
6 NO SPECIFICATION GIVEN
7 DISK FFC0 NOT AVAILABLE
43 FUNCTION ABORTED
44 INVALID ESCAPE CODE IN FIX
45 AU REVOIR UNTIL THE NEXT TIME
46 TEXT LENGTH EXCEEDS UTILITY LIMIT
47 REQUEST FOR INVALID RECORD FROM FILE FFB0
48 BREAK COMMAND RECEIVED
106 WORK FILE(S) ARE PRESENT
107 RECOVER OR REMOVE REQUIRED
108 INVALID REQUEST FFC0
109 FFB0 IS NOT A RECOGNIZED SYSTEM FILE
110 FILE FFB0 HAS BEEN ALTERED
111 WORK FILE(S) BEING REMOVED
112 UNABLE TO RECOVER
113 WORK FILE(S) ALREADY IN USE
114 LAST OPERATION WAS FFC0
115 FFB0 FILE ALREADY PRESENT
116 SAVE OR REMOVE REQUIRED
117 MESSAGE FILE REQUIRED
118 FFC0 MESSAGE FILE LOADED
119 UNKNOWN REQUEST FFC0
120 NO FILE NAME IN PARAMETERS
121 ENTER NEW LANGUAGE TYPE
122 LANGUAGE TYPE HAS TOO MANY CHARACTERS
123 END FFC0
124 FFA0 VERB ENTRIES UPDATED
125 THIS ITEM CANNOT BE TRANSLATED
126 MIS-MATCH FOUND IN NEW TEXT
127 RECOVERY COMPLETED
128 FFC0 WORK FILE LOADED
129 FFA0 RECORDS SAVED IN FILE FFB1
130 **** UTILITY LIMIT REACHED ****
131 COMMAND IN PROCESS ...
132 I/O ERROR DETECTED IN WRITE TO FILE FFB0
133 I/O ERROR DETECTED IN READ FROM FILE FFB0
134 ERROR AT RECORD NUMBER FFA0
503 DISK FFC0 LOCKED
607 ENTER FF00 TO FINISH OR FF01 TO CONTINUE

612 SYSTEM DISK IS LOCKED
 943 FFB0 NOT AVAILABLE - DISK LOCKED
 921 FFB0 IS VERSION FFC1
 941 WORK FILES REMOVED
 942 LIST OF FFC0 SECTION EXHAUSTED AFTER FFA1
 ENTRIES
 944 WORK FILES NOT AVAILABLE - DISK LOCKED

UPDATE

- 1 INVALID DICTIONARY ENTRY FFA0
- 2 ILLEGAL PARAMETER LIST FFC0
- 3 INVALID CHARACTER IN IDENTIFIER FFC0
- 4 FFB0 NOT FOUND
- 5 FFB0 NOT AVAILABLE - IN USE
- 6 NO SPECIFICATION GIVEN
- 7 DISK FFC0 NOT AVAILABLE

 195 FFB0 IS NOT TYPE SOURCE OR DATA
 196 SPECIFICATION TOO LONG
 284 ILLEGAL PARAMETER LIST - BAD ATTRIBUTES
 286 FFB0 FILE TYPE IS NOT DATA
 287 ILLEGAL PARAMETER LIST - TABS ERROR
 288 OUTPUT FILE FFB0 TOO SMALL
 289 PERMANENT ERROR ON OUTPUT FILE FFB0
 503 DISK FFC0 LOCKED
 612 SYSTEM DISK IS LOCKED
 656 CANNOT OPEN FILE FFB0 - DIRECTORY FULL
 943 FFB0 NOT AVAILABLE - DISK LOCKED

WL

- 4 FFB0 NOT FOUND
- 5 FFB0 NOT AVAILABLE - IN USE

 132 I/O ERROR DETECTED IN WRITE TO FILE FFB0
 133 I/O ERROR DETECTED IN READ FROM FILE FFB0
 134 ERROR AT RECORD NUMBER FFA0
 612 SYSTEM DISK IS LOCKED
 943 FFB0 NOT AVAILABLE - DISK LOCKED
 1069 INVALID IMPLEMENTATION LEVEL OF FFC0

XD

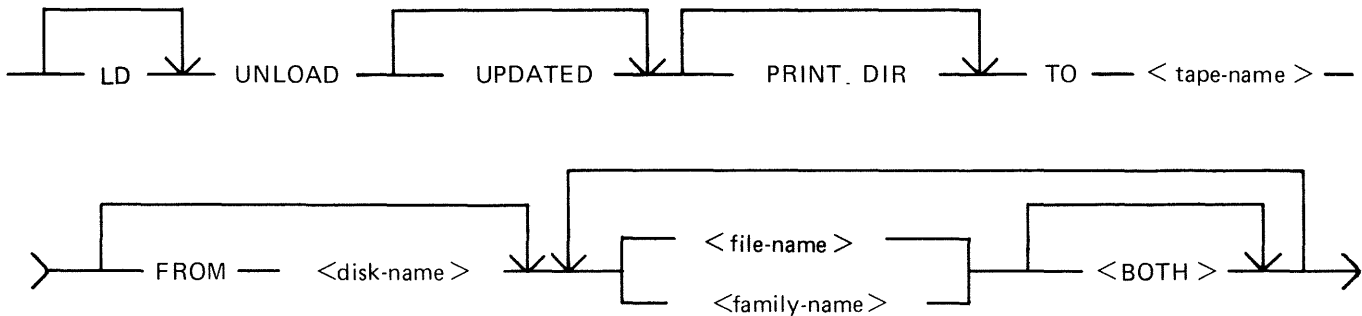
- 1 INVALID DICTIONARY ENTRY FFA0
- 2 ILLEGAL PARAMETER LIST FFC0
- 3 INVALID CHARACTER IN IDENTIFIER FFC0
- 6 NO SPECIFICATION GIVEN
- 7 DISK FFC0 NOT AVAILABLE
- 99 DISK FFC0 IS A PSEUDO-DISK

 264 SECTOR FFC0 BEYOND END OF DISK FFC1
 268 SECTORS FOR XD NOT AVAILABLE
 269 CANNOT DELETE SECTORS FROM FFC0
 270 AVAILABLE TABLE FULL - ENTRY FFC0 FFC1
 LOST
 271 FFC0 SECTORS FROM FFC1 DELETED
 503 DISK FFC0 LOCKED
 523 PLEASE RUN SQ ON FFC0

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.

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.

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 on 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

Message	Description
NO FILE <file-name> ON DISK <disk-name> FOR UNLOAD	This file is not present on the disk. The utility continues looking for other file(s) to dump, if specified.
NO FILES IN THE FAMILY <family-name> ON DISK <disk-name> FOR UNLOAD	No files have been found by UNLOAD in the family <family-name>. The utility continues to look for other files to dump, if specified.
NO FILES TO DUMP	The utility has found that none of the files specified will be dumped. The utility goes to End of Job. Check that the disk has the required file(s).
<file-name> DUMP DISCREPANCY	The end of file has been reached sooner than expected, and implies that there is erroneous information in the Disk File Header for that file. The utility will continue trying to load the rest of the file(s) specified. Try loading this file using another tape unit.
<file-name> REMOVED	This file has been copied to the tape and now removed from the disk.
<file-name> DUMPED	This file has been dumped onto the specified tape.
<file-name> NOT FOUND	The file containing the <parameter-list> (star file) is not present. The utility goes to End of Job. Make the star file available or specify the parameters in the initiating message.
DISK <disk-name> NOT AVAILABLE or with a pseudo-disk arrangement PSEUDO-DISK 'pseudo-disk-id' ON DISK	The specified disk is either not on the system or not ready. The utility will go to End of Job. Ensure that the required disk is on the system

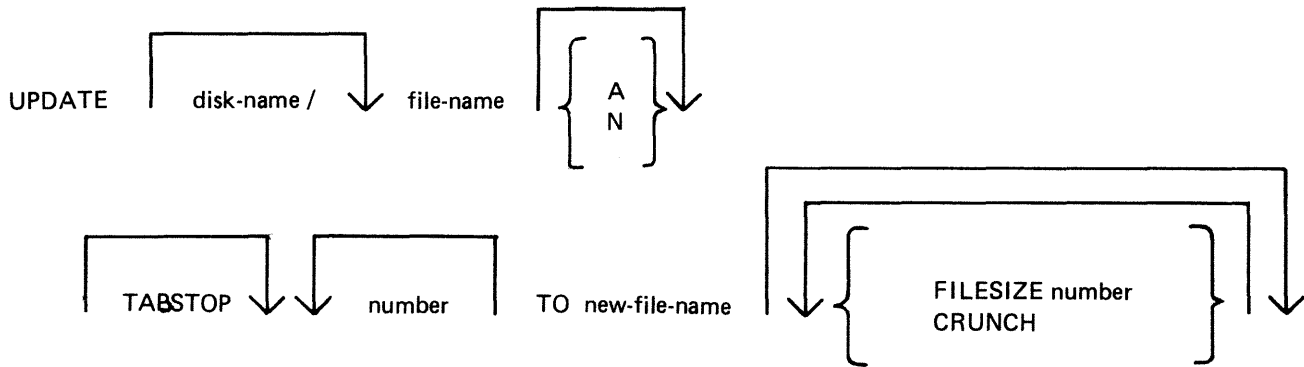
(continued)

Message	Description
<disk-name> NOT AVAILABLE and if the pseudo-disk is unrestricted, then FILE LIST FOR <family-name> MAY BE INCOMPLETE or FILE LIST MAY BE INCOMPLETE	and ready, before re-running the utility.
<file-name> NOT DUMPED - DATA FILE NOT ON LINE	The data file for the given key file is not present on the disk. The data file is not dumped. Note that the key file has been dumped. The utility will continue. Retry the utility with a disk which contains both data and key files.
UTILITY LIMIT REACHED AT FILE <file-name>	The dump is from an unrestricted pseudo-pack, the number of files to be dumped has reached the limit of 2804 files. The utility will now stop collecting any more file names, and commence the actual dumping.
INVALID CHARACTER IN IDENTIFIER <file-name> - WARNING	The utility has generated a file name which contains invalid characters. The utility will continue.
<file-name> NOT DUMPED - <reason> where <reason> can be: IN OUTPUT USE or HAS BEEN REMOVED or HAS BEEN ALTERED or IS INCOMPLETE or REASON UNKNOWN or DISK WAS LOCKED In this case, the additional message below will be displayed: LOAD ABANDONED - ERROR IN DUMP	The requested file was not dumped for the reason explained in the message. If the reason was that the disk was LOCKED, that is, unavailable to LD, then the utility goes to End of Job as there are no further files on the tape. For any other reason, the utility will continue trying to load any further requested file(s).
DUPLICATE - <file-name> ALREADY BEING DUMPED	More than one request has been made to dump this file in the initiating message. The utility will continue. Note this is usually caused by specifying individual files and the family group containing the files.
COMMUNICATE FAILURE DUMP ABANDONED - TAPE SHOULD BE PURGED	LD is running on a B 90 with an MCP version equal to or greater than 3.04. The utility has requested to select the speed at which it uses the tape drive. The MCP has reported an error to the utility. The utility terminates.

UPDATE (Disk File Update)

This utility allows the operator to construct new disk files from existing files. "CREATE" and "AMEND" use many similar features.

Format diagram:



The existing file must be a source or data file. Attributes such as Record and Block 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).

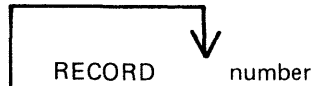
OCK4 will output a display of the above options. If OCK4 is selected while in MODIFY, SELECT or INSERT mode, an asterisk (*) will be displayed preceding that mode.

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:

Format diagram:



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 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 (data hexadecimal)
                                           301 351 401 420 451
```

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 ABCDZXXXXXXXXZZZOPQRST
```

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

Error Message	Description
<file name> NOT AVAILABLE – DISK LOCKED	Another program has locked the disk required by the utility.
<file name> NOT AVAILABLE – IN USE	Another program is using the file required by the utility.
<file name> NOT FOUND	The file <file name> is not present.
I/O ERROR DETECTED IN <WRITE TO> or <READ FROM> FILE <file name> followed by ERROR AT RECORD NUMBER <record number>	An error has been detected by the utility when trying to read/write to <file name>.
SYSTEM DISK IS LOCKED	The utility tried to access the System disk, and found that it was locked by another program.
INVALID IMPLEMENTATION LEVEL OF SYSCONFIG	The level of SYSCONFIG is not compatible with this version of WL. Run CONFIGURER from this release to create the required SYSCONFIG.

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 of 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

Note: 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.
SECTORS FOR XD NOT AVAILABLE	Requested sectors are allocated to a file, missing, or previously XD'd.	Refer to a KA listing to determine which file, if any, can be RM'd to make the sectors available.
CANNOT DELETE SECTORS FROM @address@	Only the part of the area specified for deletion is available. The utility will remove the available area only.	None.
ADDRESS @address@ BEYOND END OF DISK disk-name	The sector address specified is greater than the address of the last physical sector on the disk.	None.
DISK disk-name IS A PSEUDO DISK	The specified disk is a Pseudo disk. The utility will go to End of Job.	None

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 in-place 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 15K bytes.

The workfile buffer default size for index sorts and regular sorts is 720 bytes.

INVOKING THE SORT

The sort is executed by entering the name SORT preceded by disk-name if not on the system disk, and followed by the sort language specifications or an asterisk plus "star-file" name. The "star-file" contains the sort-language statements, and may reside on card, cassette or disk. The star-file name may be omitted, in which case the sort statements must be on a system disk file named SORTSPEC.

Examples:

To invoke the sort using a star-file named SRTLANG on the system disk:

```
SORT *SRTLANG
```

To execute the sort from disk PB4 using a star-file SORTSPEC on the same disk:

```
PB4/SORT *PB4/SORTSPEC
```

If the sort specification is given in the initiating message, it cannot be longer than 255 characters. If the sort statement is zipped from a user program it cannot be longer than 716 characters. If it is not possible to specify a complex sort or merge within these limitations, a star-file should be used.

For a one-part star-file name of 7 characters or less, the file will be searched for first on cards, then on cassette, then on the system disk. For a two-part name, the file will be searched for on a user disk. For a one-part name of more than 7 characters, the file will be searched for on the system disk.

If the required file is not found, the sort displays:

```
FILE file-name UNAVAILABLE  
FIX AND REPLY "OK" ELSE <NULL>
```

and waits on an ACCEPT.

There are two alternative responses:

1. Make the file present and enter OK to the ACCEPT, to resume execution.

or

2. Enter a null response (terminator only) to the ACCEPT, to cause EOJ.

If the specification statements are provided in the initiating message, control characters such as carriage return and line feed are treated as space characters.

A star-file on cassette must be created by the COPY utility, not the LD utility.

A star-file on disk must be of type data or source, and should not be in use by other programs.

Input statements may be printed on the printer, unless inhibited by a user option (see later), or if provided in the initiating message.

THE SORT LANGUAGE

The specification for a sort consists of three statements:

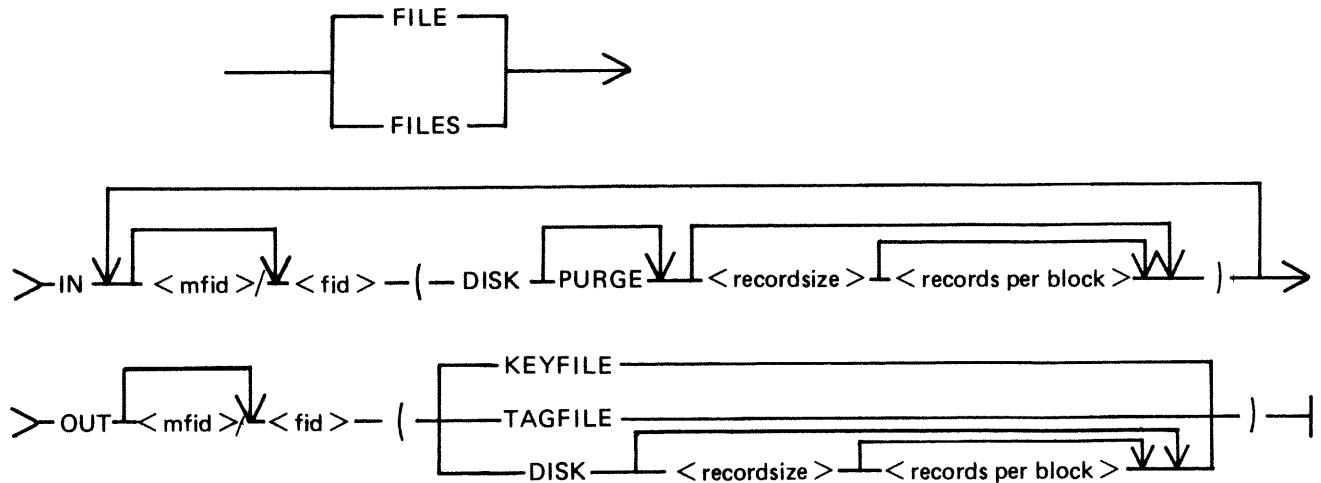
1. The FILE Statement
2. The KEY Statement
3. The USER-OPTION Statement

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.

Railroad diagram:



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.

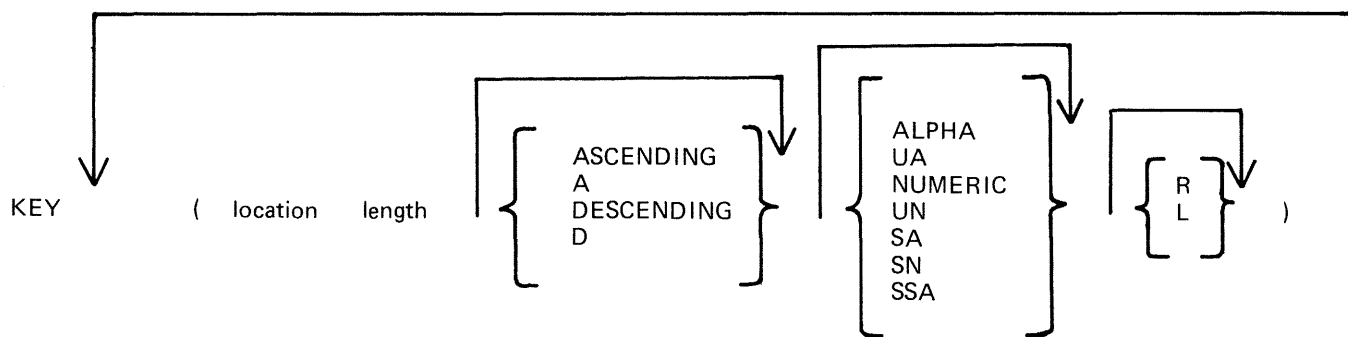
For a tagfile creation sort, the output specification enclosed in parentheses must be the single word TAGFILE. The output will be disk and record and block sizes are not user definable.

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.

Format 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.

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”.

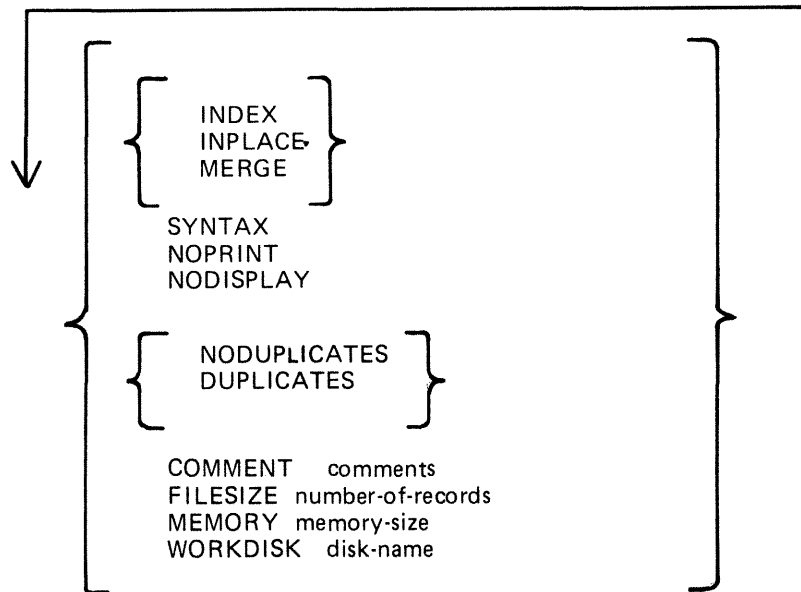
The User-Option Statement

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.

Format 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.

Note that if the INPLACE option is specified on a system that has not implemented the inplace sort, then a regular full record sort will be performed.

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 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 or 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
```


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.

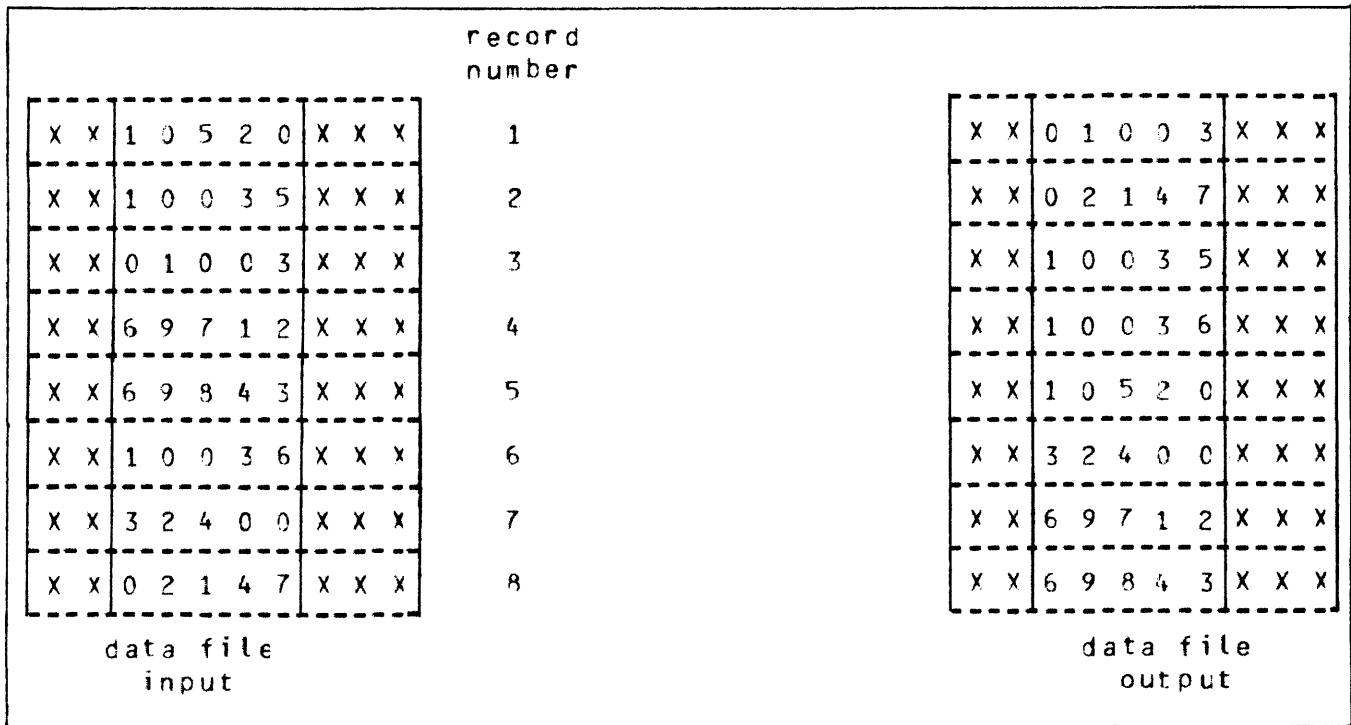


Figure 5-1. Regular Record Sort

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 WORKDISK 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 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.

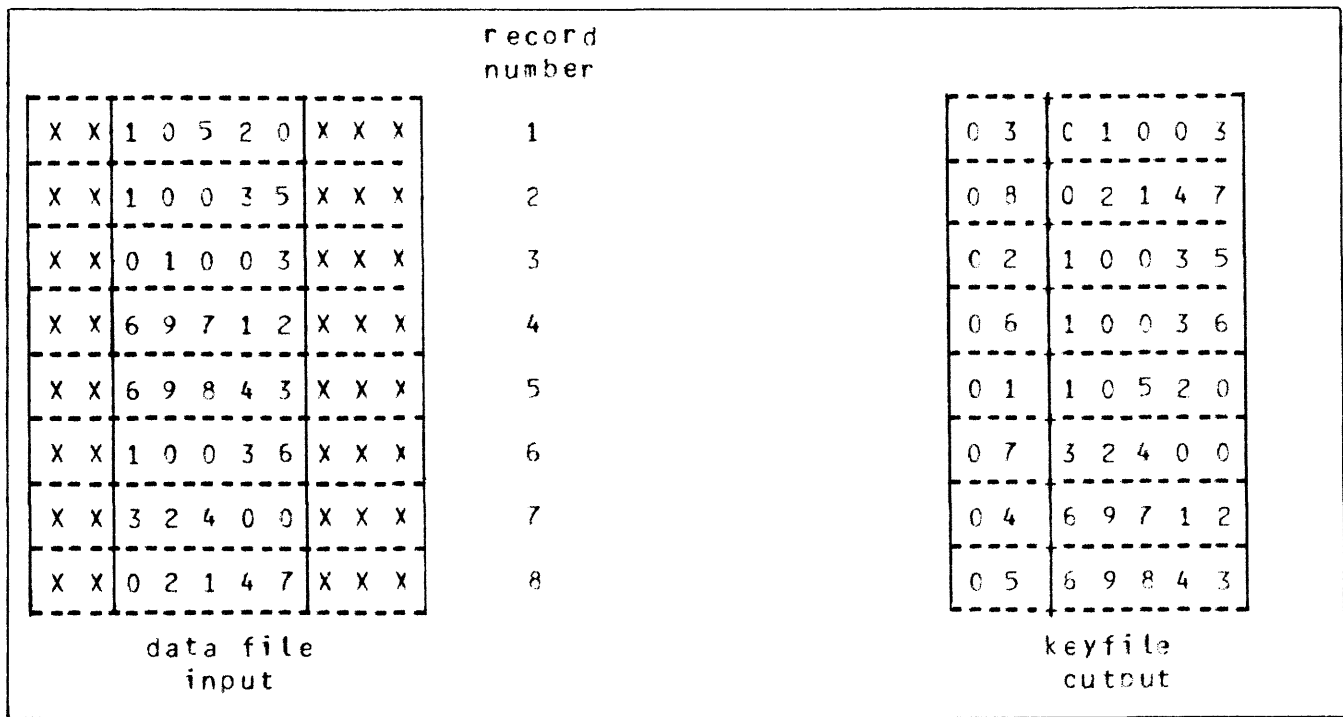


Figure 5-2. Keyfile Creation

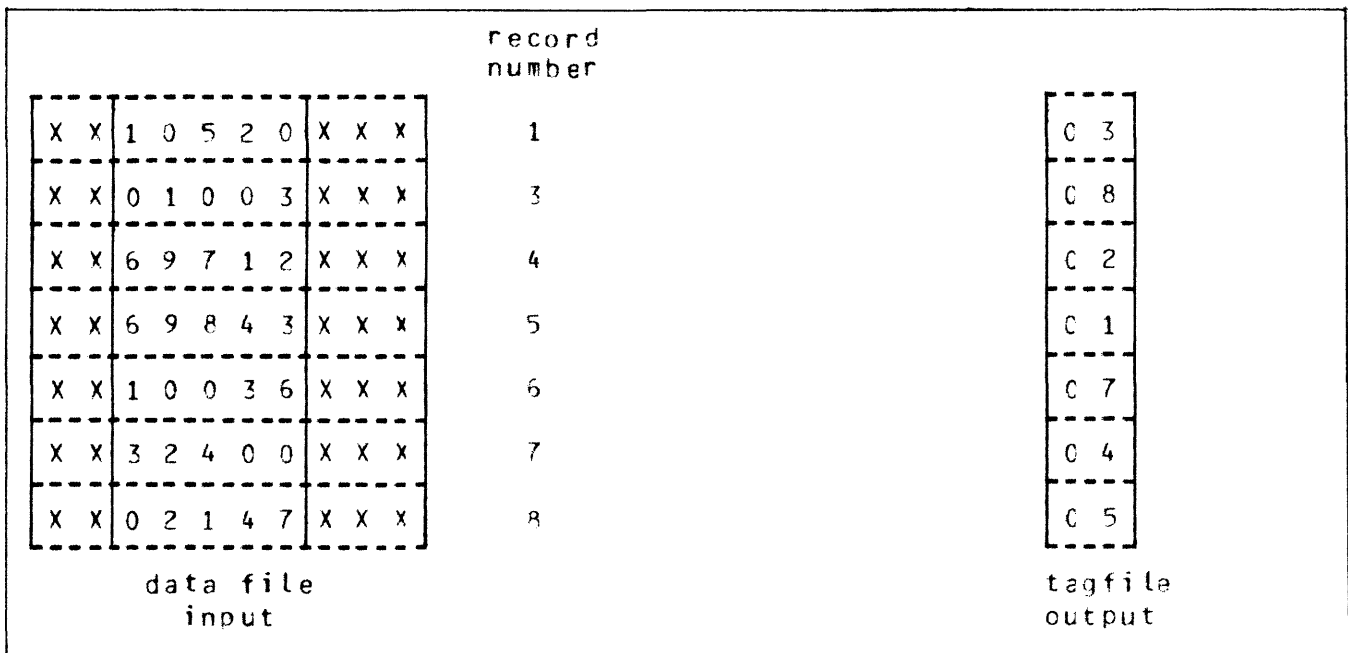


Figure 5-3. Tagfile Creation

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).

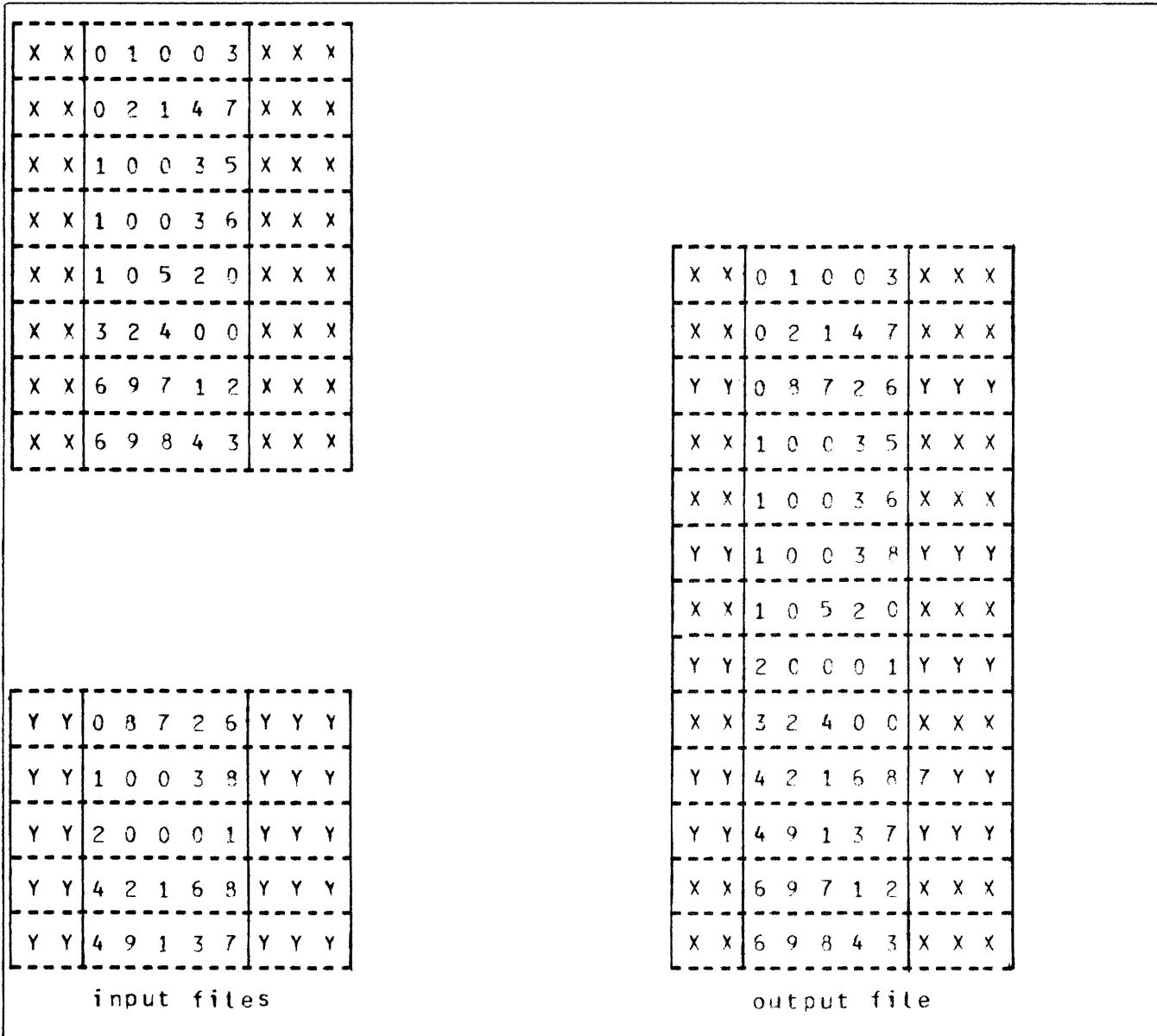


Figure 5-4. File Merge

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.

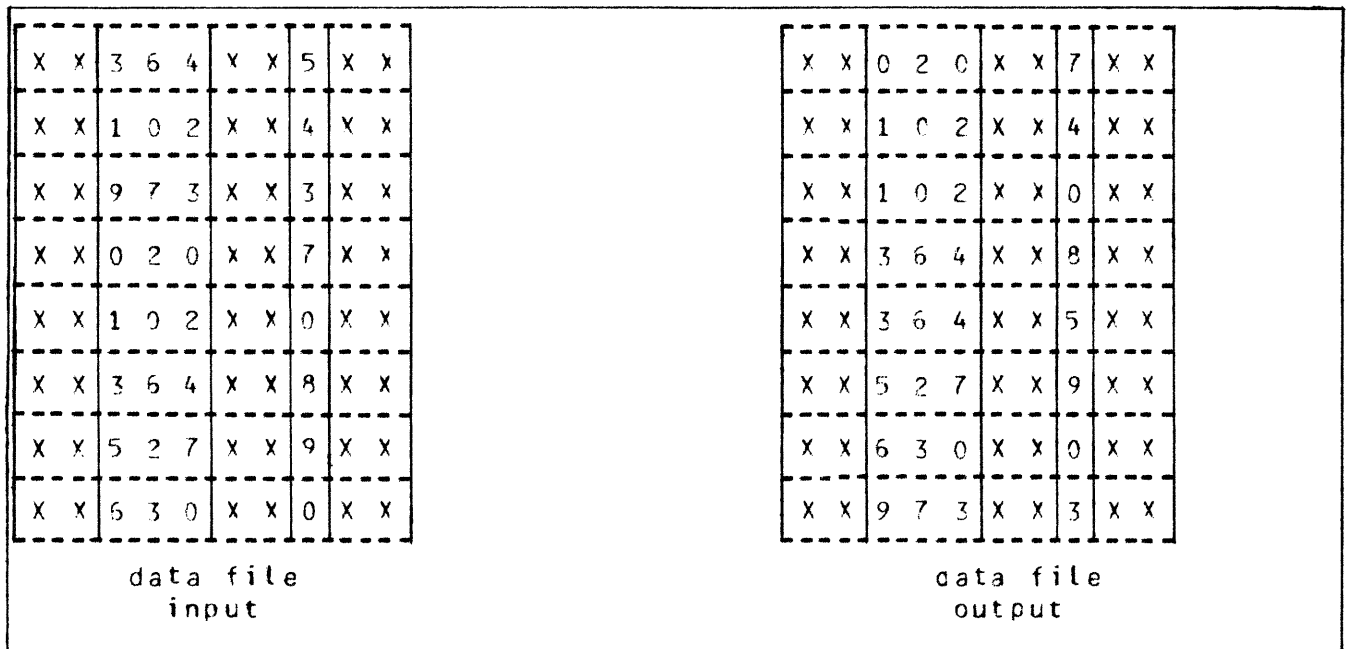


Figure 5-5. Multiple Key Sort

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, with only the error or warning number being given (no explanatory text). 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...

Number	Message
11	MERGE <SORT TYPE OPTION> NOT SPECIFIED
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

(continued)

Number	Message
75	ALPHANUMERIC KEY LENGTH NOT EVEN NUMBER OF 4-BITS...
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 | 7 - output file error |
| 2 - parity on input file | 8 - parity on sort workfile |
| 3 - EOF on sort workfile | 9 - parity on input file (block ignored) |
| 4 - parity on output file | |
| 5 - sort workfile error | |
| 6 - input file error | |

SECTION 6

COMPILATION FACILITIES

INTRODUCTION

Compilation of programs written in CMS COBOL, RPG and MPL can be performed with the CO (compile) utility. CO is a normal utility program residing on disk. It is used to co-ordinate the various parts of the CMS COBOL, RPG and MPL compilers. Each compiler consists of a number of programs, or passes, which must be executed in a specified order. CO controls the execution of these passes, the order being determined by the passes themselves. CO also controls the allocation of intermediate workfile names, to enable multiple compilations of the same language to take place in parallel. The term Compile Tag is used to uniquely identify a compilation.

No more than 26 concurrent compiles are allowed.

CO creates and maintains a master file, CO.MASTER, in which is stored information about the operating environment and the progress of each compilation. The compiler passes have access to this file which acts as the interface between them and CO. Information also in this file allows the CO utility to perform restarts if the system halts during a compilation. This restart facility eliminates the need to rerun a compilation from pass one if one or more passes have already completed successfully.

The file CO.MASTER resides on the same disk as the CO controlling the compile. If a CO.MASTER already exists, CO attempts to use it rather than create a new one. However, if a CO.MASTER is found whose attributes do not match those expected, CO displays the following message:

```
EXISTING FILE CO.MASTER IS INCOMPATIBLE WITH THIS VERSION OF CO. TO REMOVE CO.MASTER AND CONTINUE ENTER "Y" TO ACCEPT. ANY OTHER ENTRY WILL CAUSE CO TO TERMINATE.
```

DETAILS OF "CO.MASTER" FOR RELEASE 3.4

FID	= CO.MASTER
MFID	= same as for controlling CO
FILETYPE	= @45@
FILESIZE	= 27
RECORD	= 180
BUFFER	= 180
IMPLEMENTATION LEVEL NUMBER	= 0 (binary)

CO uses some standard names for input and output files, which can be changed by the initiating CO message. The basic CO operation is given in figure 6-1.

Initial input to CO is either from the initiating SPO message or through macro (star) files or through a disk file called 'CO.STARTUP' on the system disk. There is an option to produce a CO listing. Information provided to CO enables the user to describe the following:

- input patch
- input source
- output source file
- output object program
- output compilation listing
- compiler workfiles

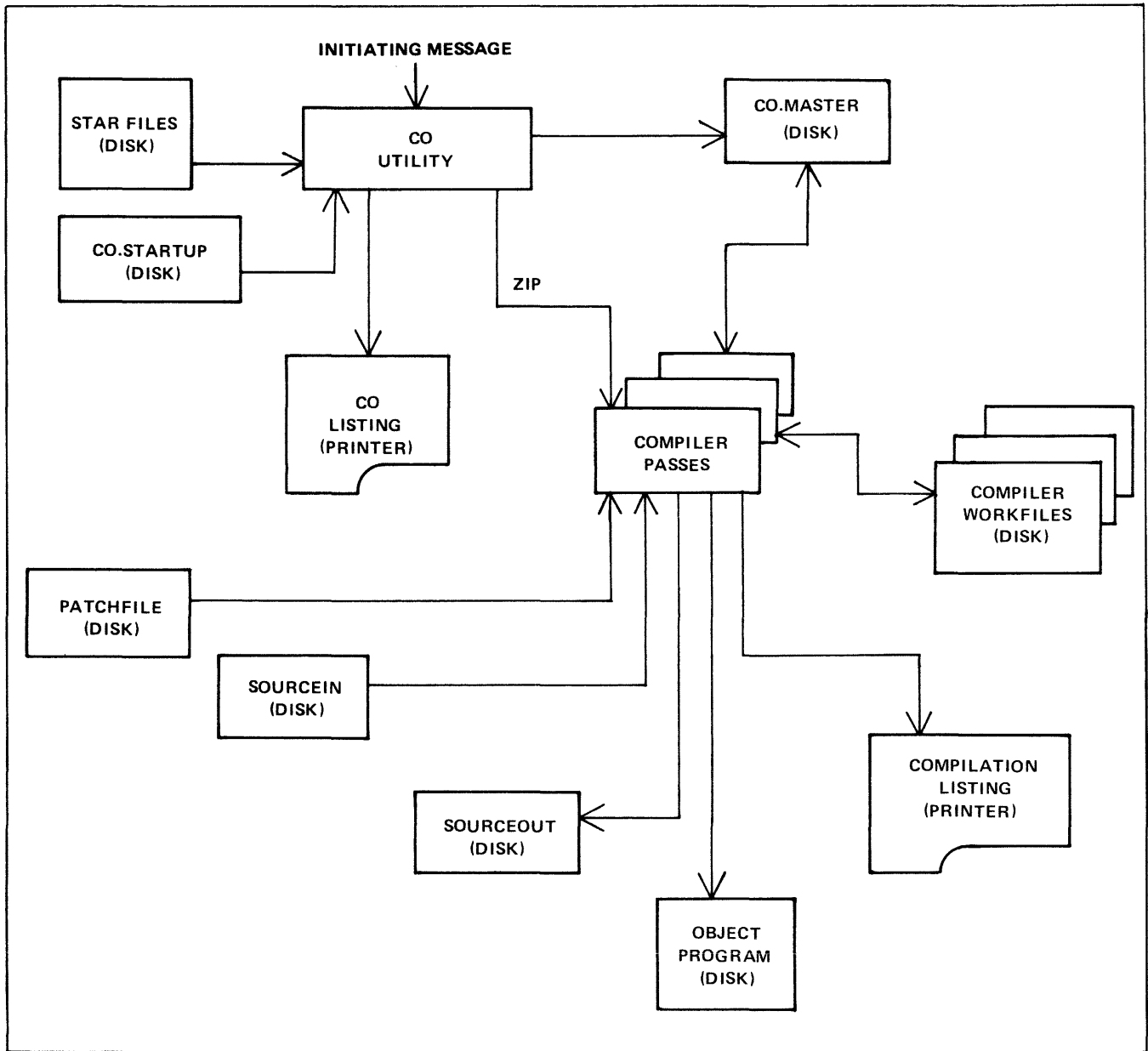


Figure 6-1. Operation of CO Utility

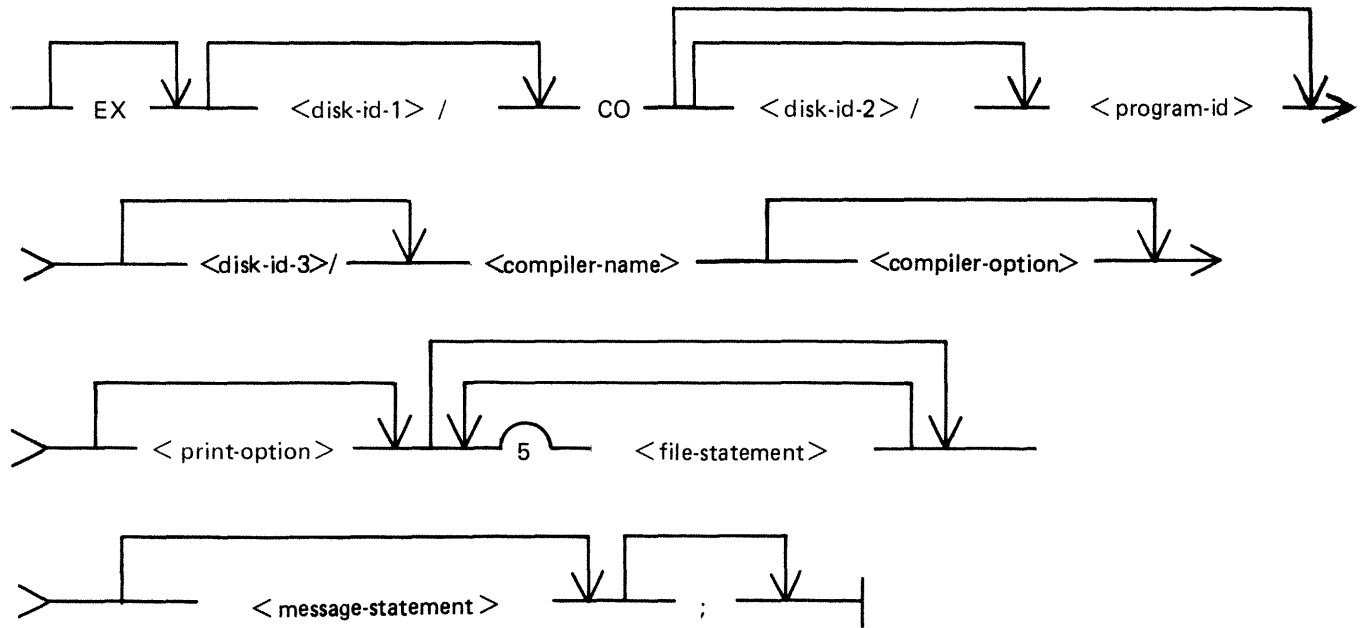
CO provides the ability to set “dollar options” for the compilation.

CO operates in two basic ways:

1. To initiate and control a single compilation.
2. To interrogate compilation status, and reset or clear an aborted compilation.

TO INITIATE A SINGLE COMPILATION

Railroad diagram:



General Parameters

< disk-id-1 >

The disk-name of the disk on which the CO utility resides, and on which "CO.MASTER" will reside if no disk-name is given, the system disk is assumed.

< disk-id-2 >

The disk-name of the disk on which the generated object program is to be placed. If no disk-name is given, the system disk is assumed.

< program-id >

The name of the generated object program. If no program name is specified, then the object program will be created on the system disk with a name as follows:

- "COBJECT" – for COBOL compilations
- "RPGOBJECT" – for RPG compilations
- "MPLOBJECT" – for MPL compilations

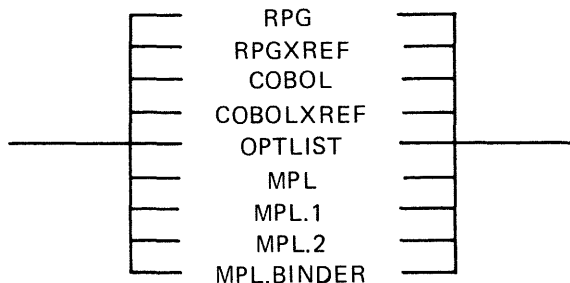
< disk-id-3 >

The disk-name of the disk from which all passes of the requested compiler are to be taken. If no disk name is given, then the compiler is assumed to reside on the system disk.

2015228

<compiler-name> **Railroad diagram:**

The name of the compiler to be used.



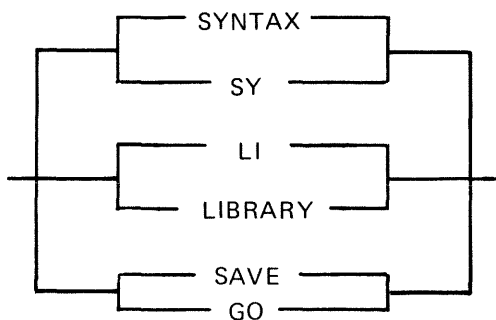
<compiler-name>

Meaning

RPG		Execute RPG Compiler.
RPGXREF		Execute RPGXREF (Stand-alone).
COBOL		Execute COBOL Compiler.
COBOLXREF		Execute COBOLXREF (Stand-alone).
OPTLIST		Execute OPTLIST (Stand-alone).
MPL		Execute MPL Compiler.
MPL.1	(*)	Execute only Pass 1 of MPL Compiler.
MPL.2	(*)	Execute only Pass 2 of MPL Compiler.
MPL.BINDER	(*)	Execute only MPL Binder.

(*) Applicable to Separate Compilation Use of the MPLII Compiler.

<compiler-option> **Railroad diagram:**



SYNTAX or SY

No object program is generated. COBSVERTER is not invoked in the case of RPG and COBOL; in the case of an MPLII compilation, processing stops at the end of pass 3.

LIBRARY or LI

An executable object program is created and saved on disk.

SAVE or GO

An executable object program is produced and saved on disk. This object program is also executed. CO goes to EOJ after zipping the object program.

<print-option>

Railroad diagram:

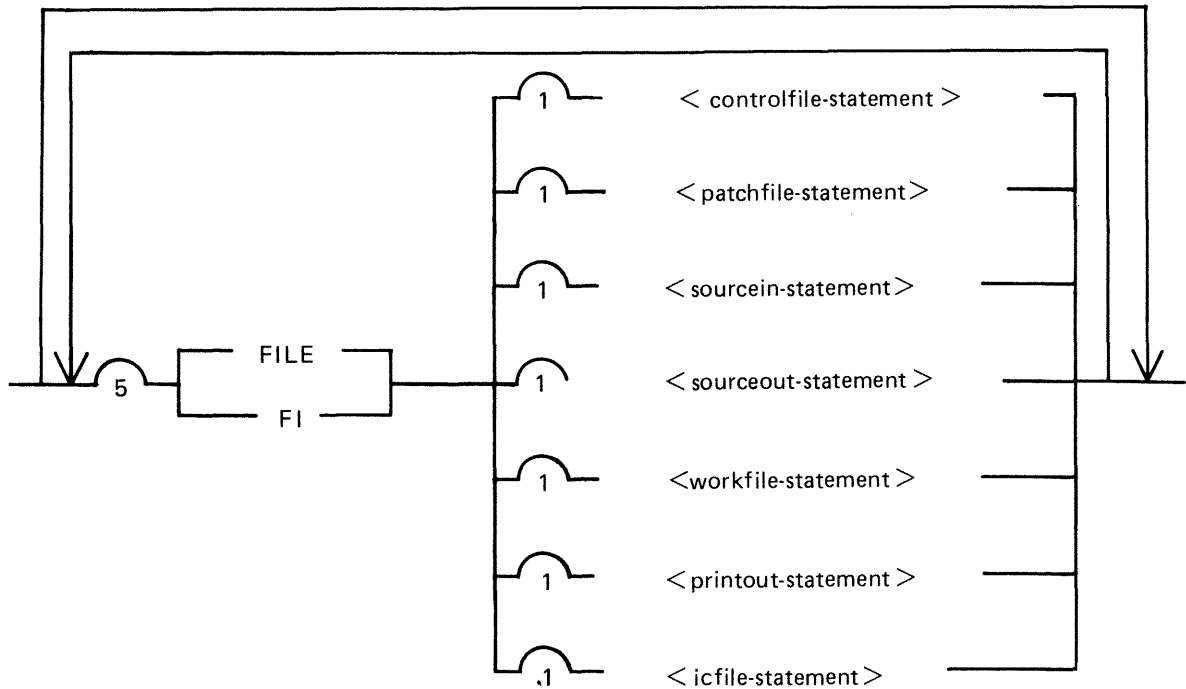


PRINT

An edited listing of the Startup Message will be output, along with a list of settings for all the file parameters which can be modified through CO for the compiler in question. A list of default dollar card settings is output along with a list of any dollar options entered via CO. The output listing also contains a log of all error messages displayed.

<file-statement>

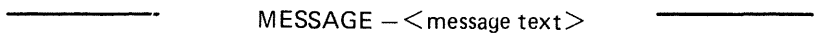
Railroad diagram:



The occurrence of FI or FILE in a CO initiating message indicates that what follows is a file modification. The words FI and FILE have no other significance.

<message-statement>

Railroad diagram:



The reserved word MESSAGE indicates the start of the list of dollar options for that compile. <message-text> consists of a list of one or more dollar cards, taken from the list in Compiler Dollar Options later in this section.

Example:

MESSAGE \$LIST \$XMAP \$SEGMENT FRED

would be valid for an MPLII compilation.

Use of the <message-statement> is not valid for COBOL compilations.

Refer to Compiler Dollar Options later in this section for a list of which dollar options may be set/reset via CO.

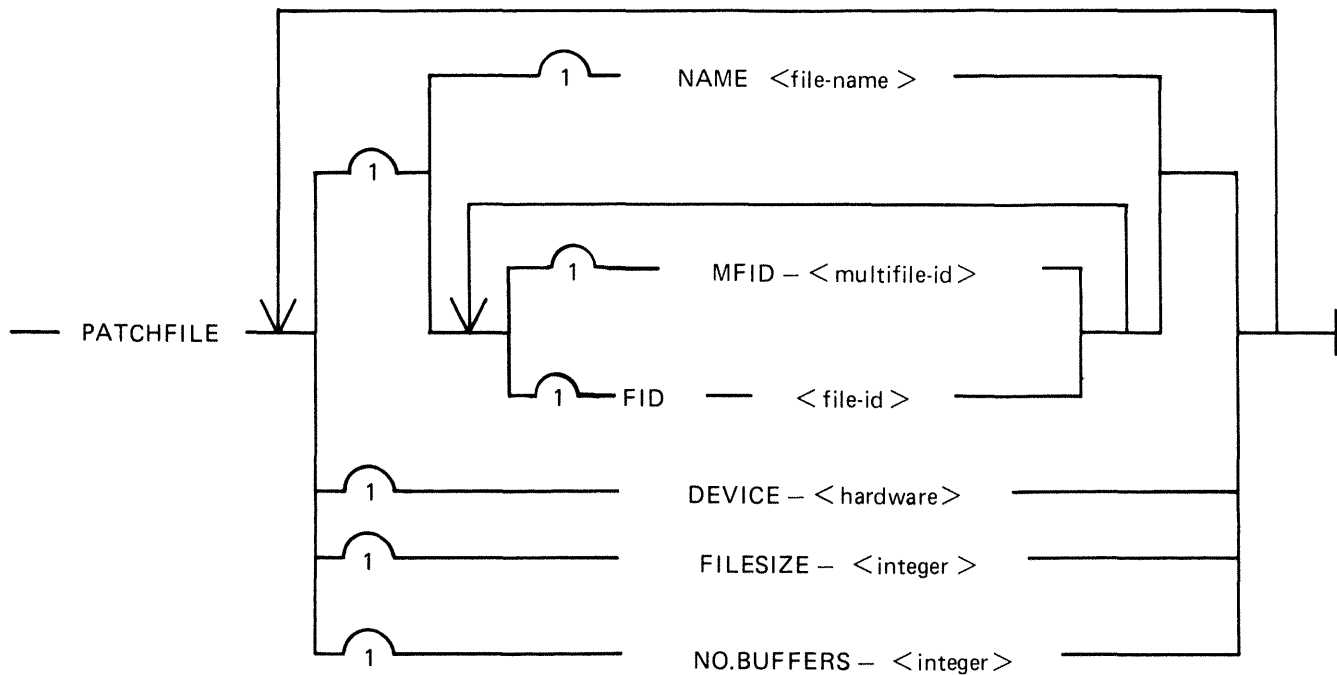
File Modification Parameters

PATCHFILE

PATCHFILE is the primary source input file, and contains dollar records and source records which may optionally be merged with a secondary input file to produce an output source file.

<patchfile-statement>

Railroad diagram:



NOTE

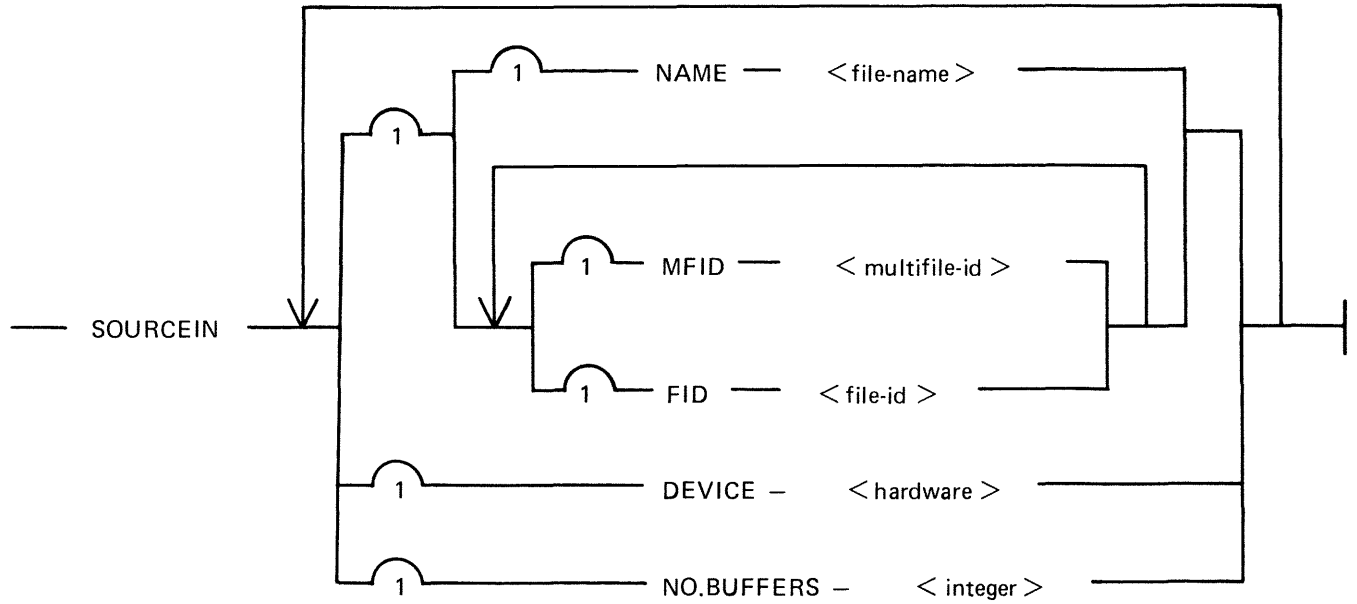
Filesize specification is only applicable for COBOL and COBOL XREF, RPG and RPGXREF. For COBOL and COBOL XREF it indicates a total size of all library files to be included in the source. Consult the RPG Reference Manual form no. 2007274 for a description of the use of FILESIZE when specified for PATCHFILE.

SOURCEIN

SOURCEIN is the optional secondary input file.

< sourcein-statement >

Railroad diagram:



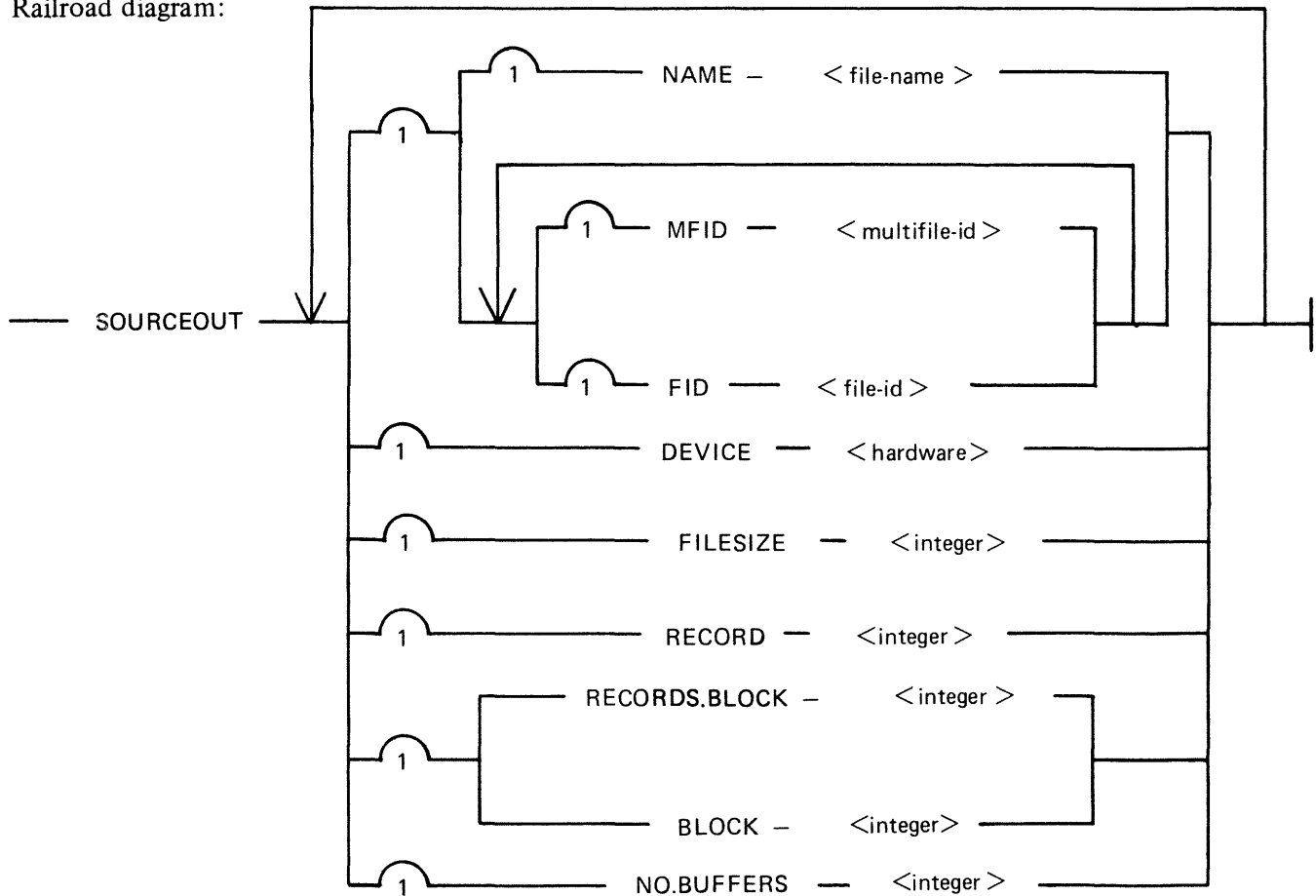
SOURCEOUT

SOURCEOUT is the optional source output file produced by merging PATCHFILE and SOURCEIN.

< sourceout-statement >

See next page

Railroad diagram:

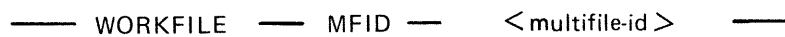


WORKFILE

WORKFILE refers to the intermediate workfiles produced by the compiler during a compilation. Specification of MFID causes these workfiles to be placed on the disk whose name is given by <multifile-id>.

<workfile-statement>

Railroad diagram:

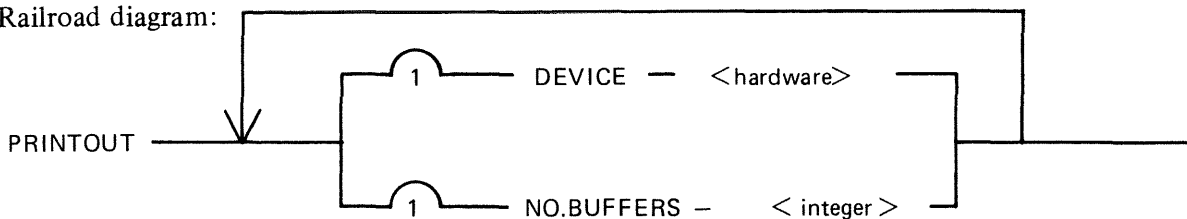


PRINTOUT

Specification of device type for PRINTOUT enables the listings from a particular compilation to be directed to a given type of printer device.

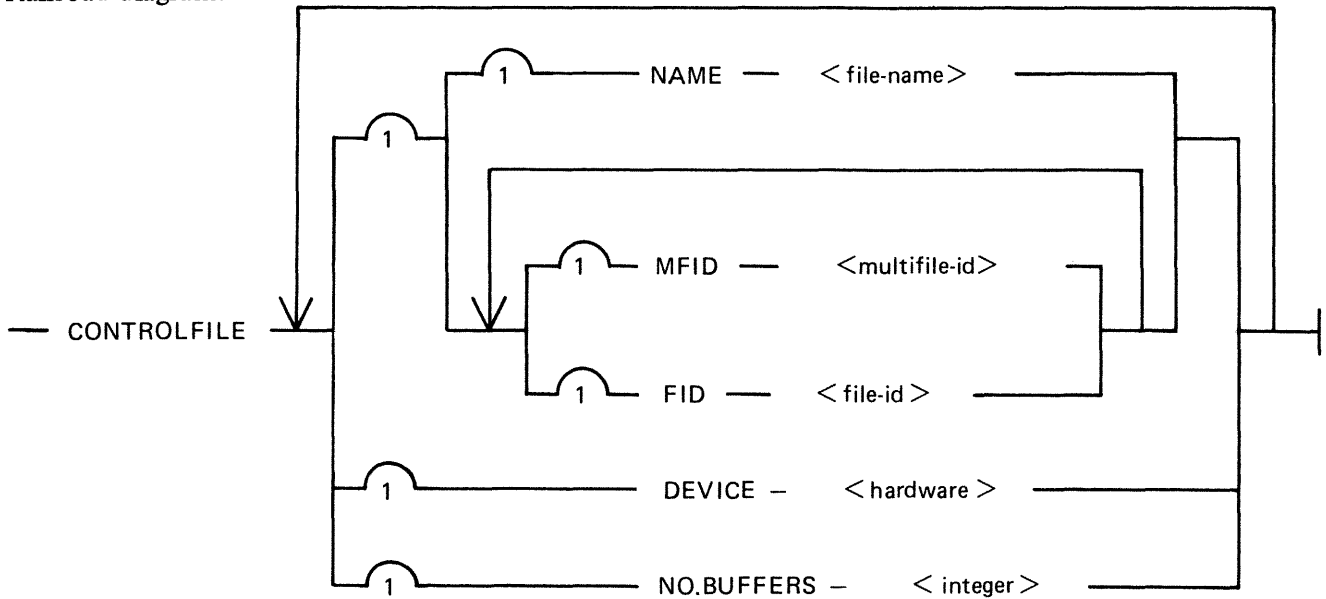
<printout-statement>

Railroad diagram:



CONTROLFILE

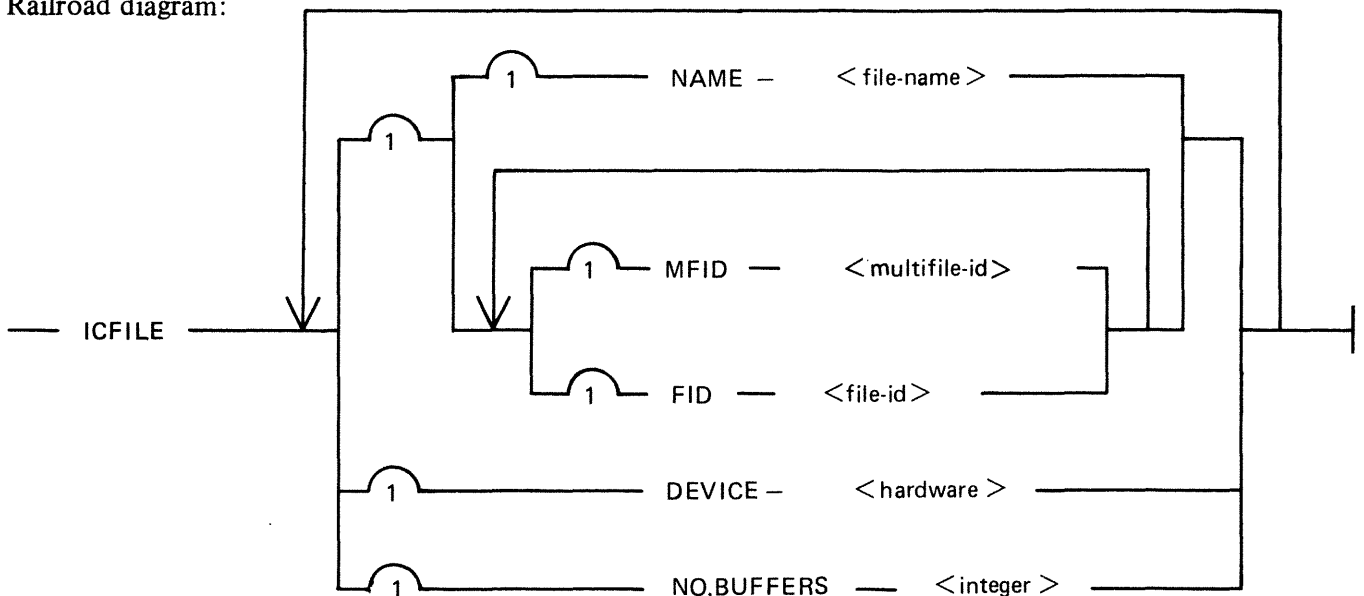
Railroad diagram:



CONTROLFILE may be used only when executing MPL.2 or MPL.BINDER, where it contains control information.

ICFILE

Railroad diagram:



ICFILE specification is only applicable when executing MPL.2, and identifies the intermediate code file (or semi-compiled file).

NOTE

Only those attributes shown in the above syntax diagrams may be modified for each type of file.

Description of the Attributes

NAME <file-name>

This specifies the CMS file-name of the Patchfile/Sourcein/Sourceout file.

MFID <multifile-id>

This specifies the name of the disk on which an input file is to be found, or an output file is to be placed.

MFID may be used in conjunction with FID as an alternative to NAME, except in the case of WORK-FILE modification, where MFID only may be specified.

FID <file-id>

This specifies the file-id of the Patchfile/Sourcein/Sourceout file.

FID may be used in conjunction with MFID as an alternative to NAME.

DEVICE <hardware>

This specifies the peripheral on which an input file is to be found, or an output file is to be placed. It may be any valid CMS device as shown in table 6-1, and is disk by default.

Table 6-1. Valid CMS Devices

Device	Mnemonic
Card Readers:	
Any card reader	AR
Any multi-function unit	AM
80-column reader	R8
80-column reader/punch	M8
96-column reader	R9
96-column reader/punch	M9
Card Punches:	
Any card punch	CP
Any multi-function unit	AM
80-column punch	P8
80-column reader/punch	M8
96-column punch	P9
96-column reader/punch	M9
Tape and Cassettes (NRZ or PE):	
Any tape	AT
Write-disabled reel	MT
Write-disabled cassette	CT
	Alternatives: CS, CASSETTE
Note: If a write-disabled device is specified for an output file, CO will substitute a write-enabled device in the specification, and issue warning.	
Printers:	
Any printer	AP
Serial printer (console)	SP

Table 6-1. Valid CMS Devices
(continued)

Device	Mnemonic
Line printer	LP
Disks:	
Any disk, default cartridge	DK Alternatives: DC, DISK
Burroughs Super Mini	DM
Fixed disk	DF
Pack	DP

NOTE

The mnemonics AT, MT, CT (CASSETTE and CS) refer only to the type of tape device. CO determines from the context whether or not write permit is required, and substitutes the device code accordingly.

FILESIZE <integer> or FILE.SIZE <integer>

In the case of SOURCEOUT, this specifies the maximum number of records in the output file. If used with PATCHFILE, then the specified value is taken to be the total number of source lines in the source program to be compiled. This includes "COPY statements" in the case of a COBOL compilation. The integer may be of the form

<integer> K

where K denotes a multiplication factor of a thousand. At least one space must separate the "K" from the integer value.

RECORD <integer>

This allows specification of the record size, in bytes, of the SOURCEOUT file. The valid range of values is 1-65535. If a number outwith this range is entered, an error message is output and the default value substituted.

If the device specified for SOURCEOUT is one of the tape family, then the maximum record/block sizes are 256 for cassettes, and 2047 for reels.

RECORDS.BLOCK <integer>

This specifies the blocking factor for the SOURCEOUT file. The block size is calculated by multiplying the record size by the blocking factor. If the resultant block size is zero or greater than 65535, an error message is displayed and the block size is set equal to the record size.

BLOCK <integer>

This specifies the block size, in bytes, of the SOURCEOUT file (CO allows the block size to be specified either as a size in bytes or as a multiple of the record size). The valid range of values for BLOCK is 1-65535. If a number outwith this range is entered, an error is displayed and the block size is set equal to the record size. The specified block size must be an integral multiple of the record size, otherwise an error message is displayed and the block size set equal to the record size.

NO.BUFFERS <integer>

This specifies to the compiler the number of buffers to be used for the file in question. The valid range of values is 1-16.

Refer to table 6-2 for a list of the default settings for these parameters.

Table 6-2. Default Settings for Parameters Modifiable Through CO

Parameter Name	Default Setting	Comment
CO Pack Id	“0000000”	< disk-id-1 >
Program Pack Id	“0000000”	< disk-id-2 >
Program Id:		< program-id >
COMPILER = RPG	“RPGOBJECT”	
COMPILER = RPGXREF	space filled	
COMPILER = COBOL	“COBOBJECT”	
COMPILER = COBOLXREF	space filled	
COMPILER = OPTLIST	“CODE-IN”	
COMPILER = MPL	“MPLOBJECT”	
COMPILER = MPL.1	“I.C.FILE”	
COMPILER = MPL.2	“MPLOBJECT”	
COMPILER = MPL.BINDER	“I.C.FILE”	
Compiler Pack Id	“0000000”	< disk-id-3 >
Compiler Option	LIBRARY	
PATCHFILE Parameters:		
MFID	“0000000”	
FID	“PATCHFILE”	
DEVICE	DK	
FILESIZE	0	
NO.BUFFERS	0	If zero, the individual compiler defaults will be used.
SOURCEIN Parameters:		
MFID	“0000000”	
FID	“SOURCEIN”	
DEVICE	DK	
NO.BUFFERS	0	If zero, the individual compiler defaults will be used.
SOURCEOUT Parameters:		
MFID	“0000000”	
FID	“SOURCEOUT”	
DEVICE	DK	
FILESIZE	4K	
RECORD	80	
BLOCK	720	
NO.BUFFERS	0	If zero, the individual compiler defaults will be used.
CONTROLFILE Parameters:		
MFID	“0000000”	
FID	“CONTROLFILE”	
DEVICE	DK	
NO.BUFFERS	0	If zero, the individual compiler defaults will be used.
ICFILE Parameters:		
MFID	“0000000”	
FID	“ICFILE”	

Table 6-2. Default Settings for Parameters Modifiable Through CO

Parameter Name	Default Setting	Comment
NO.BUFFERS	0	If zero, the individual compiler defaults will be used.
WORKFILE Parameter: MFID	“0000000”	
PRINTOUT Parameter: DEVICE	LP	
NO.BUFFERS	0	If zero, the individual compiler defaults will be used.

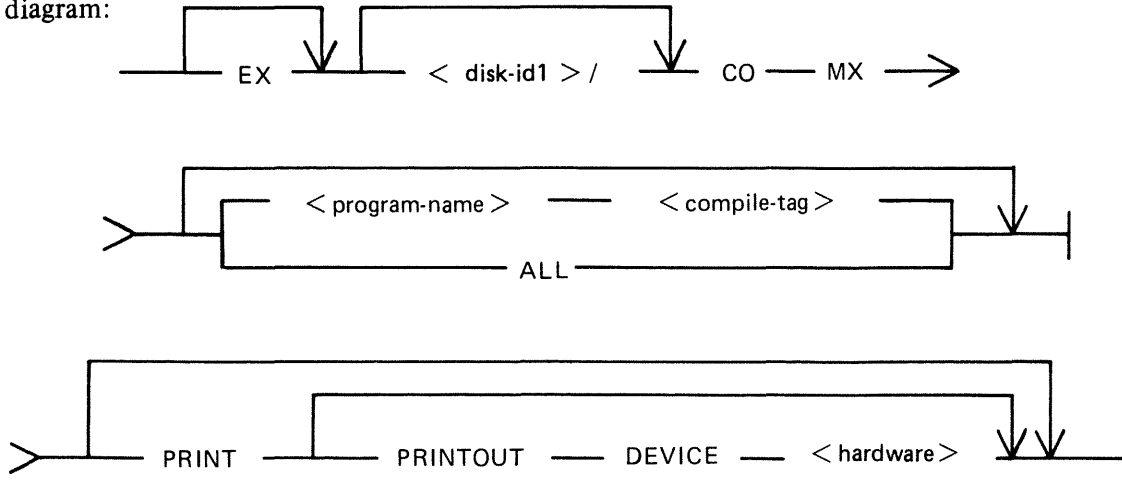
NOTE

“0000000” denotes the system disk, when used as the default setting for MFID.

TO INTERROGATE THE STATUS OF COMPILATIONS

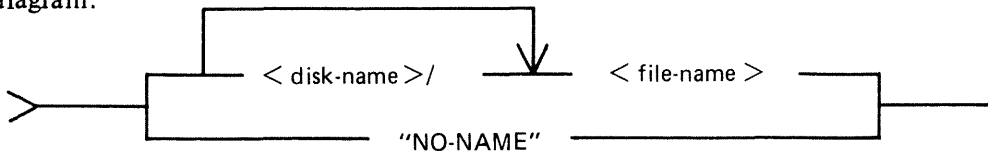
CO may be used to determine the status of any compilation(s). This is done by interrogating the CO.MASTER file. Output may optionally be directed to a serial or line printer. The initiating syntax is of the form:

Railroad diagram:



Where < program-name > is defined:

Railroad diagram:



The string "NO-NAME", including the quotes, represents a file-name of ASCII spaces. Such a filename may exist, as it is the default object program name for COBOLXREF and RPGXREF. In these cases the object program name is purely documentary, and if no name is entered it defaults to spaces.

This function can be initiated at any time, provided that the mix is not already full. It provides information of the following form for individual compiles, groups of compiles, or all compiles known to CO :

PROGRAM	NAME	disk1/fred
PATCHFILE	FID	fredp
SOURCEIN	FID	fredin
SOURCEOUT	FID	fredout
COMPILER	NAME	cobol
CURRENT	PASS	cobol2
COMPILE	TAG	a
CO	MIX-NUMBER	3
PROGRAM	NAME	disk2/joe
ICFILE	FID	joec
CONTROL FILE	FID	joectrl
COMPILER	NAME	mpl.2
CURRENT	PASS	mpl.2

COMPILE
CO

TAG q
MIX-NUMBER 15

1. "CO MX" and "CO MX ALL" are equivalent and provide information about all compiles known to CO.
2. "CO MX <program-name>" provides information about:
 - 1) 1 compile if <program-name> is unique within CO.MASTER or
 - 2) however many compiles this <program-name> has. In this case the MX output is prefixed by a statement of the form:

"<integer> COMPILES EXIST WITH PROGRAM NAME <program-name>,
THEIR DETAILS ARE AS FOLLOWS:"

3. "CO MX <program-name> <compile-tag>" provides information about a single compile. The compile-tag ensures uniqueness.

In any of the above cases, if there are either no compiles known to CO, or no compile with the specified <program-name>, (and <compile-tag>), a suitable message is displayed.

By default, MX output appears on the SPO device. However, it can be directed to the serial printer using the key-word PRINT; or to any named printer using the PRINTOUT clause.

CO has no way of distinguishing between "active" (that is, executing), and "inactive" (that is, stopped due to a system or other failure) COs. Therefore, a CO MX provides information about both types.

It can happen that more than one compile may at some time have had the same CO mix number. A compile suffering a zip failure, or which has been DS'ed or DP'ed, or which was active when a system failure occurred is still known to CO, and information about it still appears in CO MX output. Although CO does not know if any particular compile is still executing, it can tell which of several compiles with the same CO mix number was the most recent compile to have had that mix number. If a CO with this mix number is in the mix, then CO can identify it.

Where there is more than one compile with the same CO mix number, the string, (OLD), appears beside the mix number of all but the most recent compile with that mix number.

Example :

```
PROGRAM NAME prog1
.
.
CO MIX-NUMBER 2 (OLD)

PROGRAM NAME prog3
.
.
CO MIX-NUMBER 2

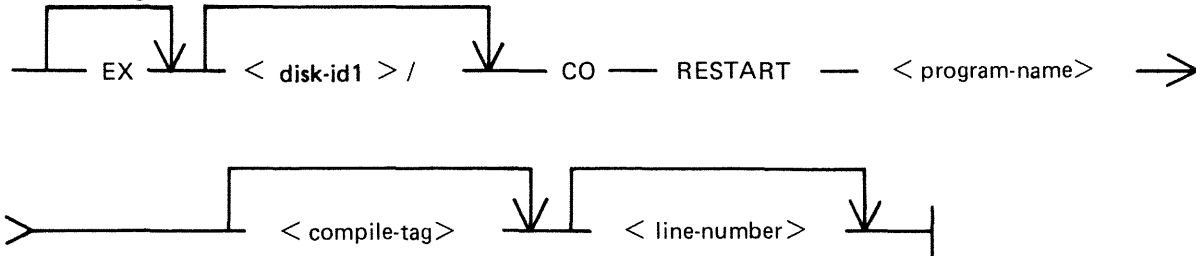
PROGRAM NAME prog2
.
.
CO MIX-NUMBER 2 (OLD)
```

In the above example, compiles with program name prog1, prog2 and prog3 have all at some time had CO mix number 2. However, prog3 had CO mix number 2 most recently, and therefore if there is an executing CO with mix number 2, then it is the one controlling the compile with program name prog3.

TO RESTART AN ABORTED COMPILATION

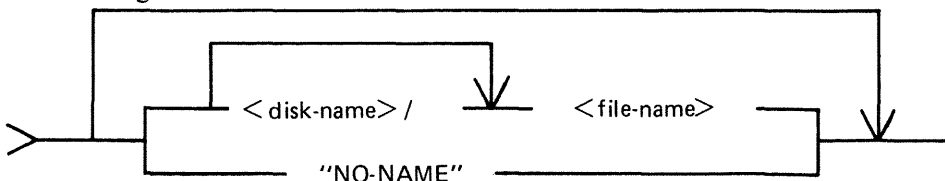
This function may be used to restart a compilation which has been terminated prematurely due to a system failure, for example, a system failure or a ZIP failure.

Railroad diagram:



Where < program-name > is defined:

Railroad diagram:



The string "NO-NAME", including the quotes, represents a file-name of ASCII spaces. Such a filename may exist, as it is the default object program name for COBOLXREF and RPGXREF. In these cases the object program name is purely documentary, and if no name is entered it defaults to spaces.

If the Restart is successful the compilation is resumed at the beginning of the pass in which the failure occurred.

As for MX and CLEAR functions, compiles are identified for RESTART purposes by < program-name > optionally qualified by < compile-tag >.

1. "CO RESTART < program-name >" either

- 1) RESTARTs the compile with the given < program-name > providing that the name is unique within "CO.MASTER", or
- 2) if < program-name > is not unique, a message is displayed saying how many compiles exist with that < program-name >.

"< integer > COMPILES EXIST WITH PROGRAM NAME < program-name >, THEIR DETAILS ARE AS FOLLOWS:"

It then provides the same information as is given for MX for each of the compiles with that < program-name >. The operator is then asked to identify, by entering in response to an ACCEPT, the compile-tag of the compile he wishes to Restart. The request reads as follows:

"ENTER COMPILE TAG OF REQUIRED COMPILE OR NULL ACCEPT TO TERMINATE RESTART ATTEMPT."

If the character entered is not in the range A thru Z, or is not one of the appropriate compile-tags, the following message is displayed:

“COMPILE TAG NOT APPROPRIATE. PLEASE TRY AGAIN.”

A Null Accept to any of these prompts causes CO to go to EOJ. No RESTART will be done.

2. “CO RESTART <program-name> <compile-tag>” RESTARTs the compile with that combination of program-name and compile-tag.

If the appropriate compile is found, one of the following messages will be output, depending on whether a compile-tag was entered.

“COMPILE WITH PROGRAM NAME <program-name> RESTARTED”, or

“COMPILE WITH PROGRAM NAME <program-name> AND COMPILE TAG <compile-tag> RESTARTED”

Although only one compile can be Restarted in one invocation of CO, any number, up to a maximum of 26 compiles, may have been Restarted at any time.

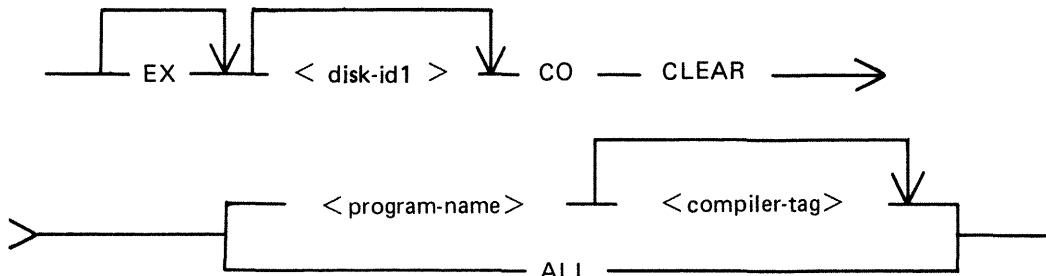
If Restarting a COBOL compilation, a line-number may be specified. This is the line-number at which printing of the output listing should be resumed.

As CO cannot distinguish between “active” and “inactive” COs, it cannot prevent the operator from Restarting or Clearing an executing CO. However, when the original version of the pass, that was executing when the RESTART was done, terminates and returns to CO, CO will recognize that the Compilation Block has been initialized, display the message “CANNOT CONTINUE COMPILATION – COMPILATION BLOCK INITIALIZED”, and terminate the job. Restarting executing COs is not advised, and CO cannot be held responsible for any problems resulting from, for example, duplicate files.

TO CLEAR AN ABORTED COMPILATION

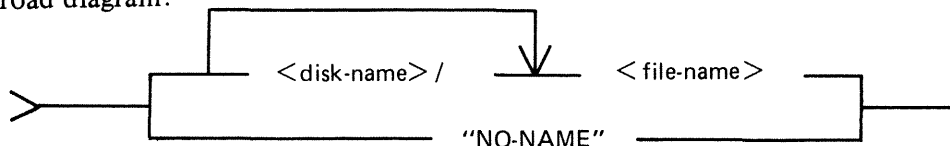
If a failure has occurred, and the operator chooses not to Restart the compilations that were running prior to the failure, then he may decide to remove all references to a specific compilation, or all compilations known to CO. As for MX and RESTART functions, compiles are identified for CLEAR purposes by program-name optionally qualified by compile-tag.

Railroad diagram:



Where <program name> is defined:

Railroad diagram:



The string "NO-NAME", including the quotes, represents a file-name of ASCII spaces. Such a filename may exist, as it is the default object program name for COBOLXREF and RPGXREF. In these cases the object program name is purely documentary, and if no name is entered it defaults to spaces.

1. "CO CLEAR <program-name>" either
 - 1) CLEARs the compile with the given program-name providing that name is unique within "CO.MASTER", or
 - 2) if program-name is not unique, a message of the following form is displayed indicating how many compiles exist with that program-name.

"<integer> COMPILES EXIST WITH PROGRAM NAME <program-name>, THEIR DETAILS ARE AS FOLLOWS:"

It is followed by the same information as is given for MX for each of the compiles with that program-name. The operator is then asked to identify by entering, in response to an Accept, the compile-tag of the compile he wishes to CLEAR. The request reads:

"ENTER COMPILE TAG OF REQUIRED COMPILE, OR NULL ACCEPT TO TERMINATE CLEAR ATTEMPT"

If the character entered is not in the range A thru Z, or is not one of the appropriate compile tags, the following message is displayed and the operator is requested to try again.

"COMPILE TAG NOT APPROPRIATE. PLEASE TRY AGAIN"

A Null Accept to any of these prompts causes CO to go to EOJ; no CLEAR will be done.

-
2. “CO CLEAR <program-name> <compile-tag>” CLEARs the compile with that combination of program-name and compile-tag.
 3. “CO CLEAR ALL” CLEARs all compiles known to CO.

As each compile is CLEARed, one of the following messages is displayed, depending on whether the compile-tag was specified:

“COMPILE WITH PROGRAM NAME <program-name> CLEARED” or
“COMPILE WITH PROGRAM NAME <program-name> AND COMPILE TAG <compile-tag> CLEARED”

When a CLEAR command is executed, all workfiles belonging to the CLEARed compilation that are still on the disk are removed by CO, unless the compiler requests that they are saved.

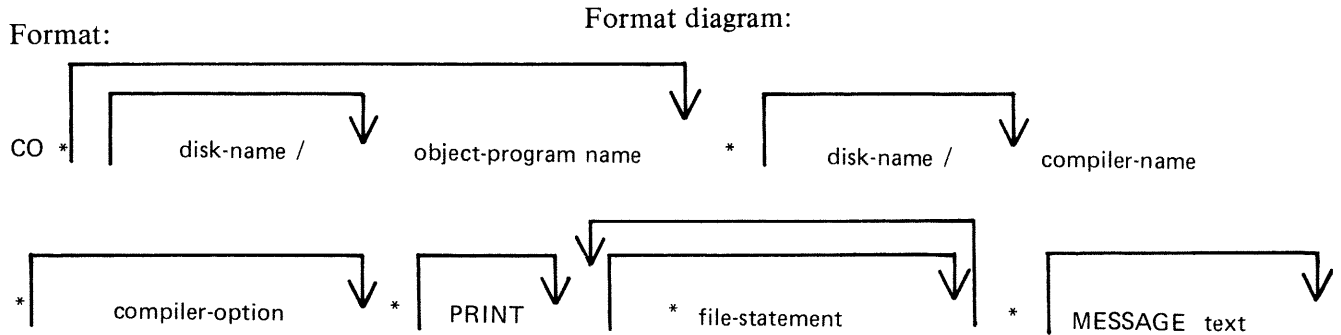
The same rules apply to CLEARing executing COs as apply to Restarting executing COs, with the added complication that CO removes any workfiles belonging to a compilation when it CLEARs it. Therefore, the previously executing pass may fail to find certain of its workfiles.

As with Restart, CLEARing one CO leaves all other COs undisturbed.

USE OF MACRO CALLS

All or part of the initiating message to CO may be provided as data in disk files. This is indicated by an asterisk (star) in the message.

Following the asterisk must be a valid file name, including the disk-name if not on the system disk. When the initiating message is scanned, the contents of a star-file are included in the scan. At the end of the star-file contents, scanning continues with the primary message. The complete initiating message, or individual clauses, may be included in the star-files. The message format for CO has been repeated below with an asterisk where a star-file may be used:



Example:

Consider four star-files, with names and contents as follows:

Name	Contents
FILE1	AA RPG GO
FILE2	PATCHFILE NAME BB
FILE3	DEVICE AR
FILE4	SOURCEOUT FID CC

Then the following two initiating messages to CO are valid:

```
CO *FILE1 PRINT *FILE2 DEVICE AR
CO *FILE1 PRINT *FILE2 *FILE4
```

but the following two messages are invalid, because the contents of FILE3 is wrong, as it starts in the middle of a file-statement:

```
CO *FILE1 PRINT *FILE2 *FILE3
CO *FILE1 PRINT PATCHFILE NAME BB *FILE3
```

COMPILER DOLLAR OPTIONS FOR CO

RPG Compiler

\$LIST	(D)	\$NLIST
\$SEQ		\$NSEQ
\$XMAP		\$NXMAP
\$SUPR		\$NSUPR
\$LOGIC		\$NLOGIC
\$MERGE		\$NMERGE
\$SEVERE	(D)	\$NSEVERE
\$MAP		\$NMAP
\$NAMES		\$NAMES
\$NEW		\$NNEW
\$XREF		\$NXREF

MPLII Compiler

\$LIST	\$NOLIST
\$LISTE	
\$LISTP	
\$SEQUENCE	
\$XMAP	\$NOXMAP
\$NOWARNING	
\$CODE	
\$NEWTAPE	
\$FORMAT	
\$SEGMENT	
< file-name >	
\$SEGSIZE	
< integer >	
\$XREF	\$NOXREF

Stand-Alone OPTLIST

\$MAP
\$COP-TABLE

Stand-Alone COBOLXREF

\$LIST or \$SET LIST| \$RESET LIST

COBOL Compiler

No Dollar options may be Set/Reset via CO.

NOTE

(D) indicates the options which are set by default. This notation is also used in CO's output listing, to distinguish default from deliberate dollar card settings.

Dollar Option Precedence

RPG Compiler

Any dollar options specified in CO's startup message override any occurrences of that option appearing in either the PATCHFILE or SOURCEIN input files.

MPLII Compiler

Any dollar option specified through CO overrides any occurrences of that option appearing in either the PATCHFILE or SOURCEIN input files.

COBOLXREF

Any setting/resetting of the LIST option in the CO message overrides any occurrence of that option appearing in either the PATCHFILE or SOURCEIN input file.

For a more detailed description of the available dollar options, consult the relevant compiler manual.

CO ERROR MESSAGES

The printout file for CO may contain error messages from the CO utility itself. These are listed in table 6-3, and are largely self-explanatory.

If errors have been detected in CO's initiating message, a message will be displayed to indicate whether or not the compiler will be zipped. If no errors have been flagged, it is assumed that the compiler will be zipped, and no message is displayed.

Table 6-3. CO Error Messages

Error Number	Error Message
1.	First word in Initiating Message is not a valid disk or file name.
2.	Invalid object program name or compiler name specified.
3.	Warning – Compiler Pack Id. truncated to 7 characters.
4.	No compiler specified.
5.	Warning – Program Pack Id. truncated to 7 characters.
6.	Warning – Program Id. truncated to 12 characters.
7.	Reserved Word not found when expected.
8.	FI/FILE not followed by one of PATCHFILE,SOURCEIN,SOURCEOUT, CONTROLFILE, ICFILE, WORKFILE,PRINTOUT.
9.	Reserved Word used out of context.
10.	Warning – Compiler Option already modified – previous modification ignored.
11.	Warning – Print Option specified more than once.
12.	Record size * Blocking-factor > 65535, Block size set equal to Record size.
13.	MESSAGE not used by COBOL – Message Text ignored.
14.	Warning – Message Text already processed – this Message Text ignored.
15.	Modification not valid for COBOLXREF – ignored.
16.	\$SEGMENT file-name parameter not a valid file name.
17.	CO.MASTER no longer on disk – job terminated.
18.	Warning – \$SEGMENT pack-id truncated to 7 characters.
19.	Warning – \$SEGMENT program name truncated to 12 characters.
20.	\$SEGSIZE parameter not numeric.
21.	Modification not valid for SOURCEIN – modification ignored.
22.	Warning – MESSAGE contains no valid dollar cards.
23.	Warning – \$card not valid for this compiler.
24.	Warning – Invalid \$card in Message Text.
25.	Warning – PATCHFILE modification(s) already done – these modifications are ignored.
26.	Warning – PATCHFILE modifications incomplete.
27.	Modification not valid for PATCHFILE – ignored.
28.	Warning – No PATCHFILE modifications.
29.	MFID/FID modification already done for this file – modification ignored
30.	Pack name/File name in NAME specification invalid.
31.	Pack name in NAME specification truncated to 7 characters.
32.	File name in NAME specification truncated to 12 characters.
33.	Pack name in MFID clause invalid.
34.	Pack name in MFID clause truncated to 7 characters.
35.	File name in FID clause invalid.
36.	File name in FID clause truncated to 12 characters.

-
37. RECORD size of zero specified – modification ignored – default used.
 38. DEVICE modification already done for this file – this modification ignored.
 39. Specified Device Type invalid – default device type used.
 40. FILESIZE already modified for this file – this modification ignored.
 41. FILESIZE parameter not numeric.
 42. BLOCK size already modified – this modification ignored.
 43. RECORDS.BLOCK/BLOCK parameter not numeric.
 44. RECORD parameter not numeric.
 45. SOURCEIN modification(s) already done – these modifications ignored.
 46. Warning – SOURCEIN modification incomplete.
 47. No SOURCEIN modifications.
 48. SOURCEOUT modification(s) already done – these modifications ignored.
 49. Warning – SOURCEOUT modifications incomplete.
 50. Warning – No SOURCEOUT modifications.
 51. (Unused)
 52. (Unused)
 53. (Unused)
 54. (Unused)
 55. Null Initiating Message – no file CO.STARTUP – job terminated.
 56. Nested Macro Call found.
 57. Invalid file name in Macro Call – <macro-name>.
 58. Pack name in Macro Call truncated to 7 characters.
 59. File name in Macro Call truncated to 12 characters.
 60. Record size for card device > 96, RECORD and BLOCK size set to 96.
 61. File CO.STARTUP empty – No Initiating Message – job terminated
 62. (Unused)
 63. CO.MASTER not found – job terminated.
 64. Compiler not zipped – errors in Initiating Message.
 65. (Unused)
 66. (Unused)
 67. No COs to RESTART/CLEAR – job terminated.
 68. (Unused)
 69. SYSTEM STATUS Communicate failure – job terminated.
 70. WORKFILE modification(s) already done – this modification ignored.
 71. WORKFILE not followed by MFID – modification ignored.
 72. Specified RESTART line-number for listing not numeric.
 73. Specified Restart Line-number > 65535 – Line-number set to 0.
 74. (Unused)
 75. No parameter specified for CLEAR function.
 76. Line-number parameter only valid for COBOL – ignored.
 77. Illegal character encountered in Initiating Message.
 78. Cannot continue compilation – Compilation Block initialized.
 79. Non-input Device specified for PATCHFILE – ignored.
 80. Non-input Device specified for SOURCEIN – ignored.
 81. Macro file not found – <macro-name>.
 82. Non-output Device specified for SOURCEOUT – ignored.
 83. BLOCK size for SOURCEOUT is not a multiple of the RECORD size.
 84. Specified BLOCK size for Tape Device too large – default value used.
 85. Specified RECORD size for Tape Device too large – default used
 86. 80 column card device RECORD size > 80, RECORD and BLOCK size set to 80.
 87. 96 column card device RECORD size > 96, RECORD and BLOCK size set to 96.
 88. Warning – PRINTOUT modification(s) already done – this modification ignored.
 89. Device specified for PRINTOUT not Printer Type – ignored.

-
- 90. PRINTOUT not followed by DEVICE or NO.BUFFERS – ignored.
 - 91. BLOCK size or Blocking-factor of zero specified – default used
 - 92. SOURCEOUT FILESIZE specified as zero – ignored – default used.
 - 93. Specified PATCHFILE FILESIZE > 65535 – default used.
 - 94. Specified SOURCEOUT FILESIZE > 65535 – default used.
 - 95. Specified RECORD size > 65535 – default used.
 - 96. Specified BLOCK size or Blocking-Factor > 65535 – default used
 - 97. When compiling MPL on Cumbernauld Support System, level number required.
 - 98. If 80 Column card output device used, record truncation will occur.
 - 99. RECORD size already modified – this modification ignored.
 - 100. Macro file present but empty – <macro-name>.
 - 101. NO.BUFFERS already modified – this modification ignored.
 - 102. NO.BUFFERS parameter not numeric.
 - 103. Value specified for NO.BUFFERS not in range 1 to 16 – default used.
 - 104. Warning – NO PRINTOUT modifications.
 - 105. Warning – PRINTOUT modifications incomplete.
 - 106. Modification not valid for PRINTOUT – ignored.
 - 107. ICFILE modification not valid for this compiler – ignored.
 - 108. CONTROLFILE modification not valid for this compiler – ignored
 - 109. CONTROLFILE modification(s) already done – these modifications ignored.
 - 110. NO CONTROLFILE modifications.
 - 111. CONTROLFILE modifications incomplete.
 - 112. Modification not valid for CONTROLFILE – ignored.
 - 113. ICFILE modification(s) already done – these modifications ignored.
 - 114. No ICFILE modifications.
 - 115. ICFILE modifications incomplete.
 - 116. Modification not valid for ICFILE – ignored.
 - 117. Non-input device specified for CONTROLFILE – ignored.
 - 118. Non-input device specified for ICFILE – ignored.
 - 119. PATCHFILE modification not valid for this compiler – ignored.
 - 120. SOURCEIN modification not valid for this compiler – ignored.
 - 121. SOURCEOUT modification not valid for this compiler – ignored.
 - 122. Invalid entry enclosed in quotes in < RESTART > statement.
(CLEAR)
(MX)
(CLEAR)
 - 123. < RESTART > program name invalid.
(MX)
(CLEAR)
 - 124. Pack name in < RESTART > statement truncated to 7 characters.
(MX)
(CLEAR)
 - 125. File name in < RESTART > statement truncated to 12 characters.
(MX)
(CLEAR)
 - 126. Invalid entry in < RESTART > statement.
(MX)
 - 127. No compile with program name <program-name> exists.
 - 128. No compile with program-name <program-name> and compile tag <compile-tag> exists.

If errors have been detected in CO's Initiating message, a message is displayed to indicate whether or not the compiler will be zipped. If no errors have been flagged, it is assumed that the compiler will be zipped, and no message is displayed.

CO Other Messages

1. COMPILER BEING ZIPPED.
2. COMPILER NOT ZIPPED.
3. CANNOT INITIATE THIS CO – MAXIMUM NUMBER OF COS ALREADY IN PROCESS
4. NO FILE <file-name> TO EXECUTE.
5. ZIP FAILURE WHEN ZIPPING <pass-name>.
6. RPG COMPILER ATTEMPTED TO ZIP RPGCOPY – NOT IMPLEMENTED.
7. JOB TERMINATED.
8. CO – MIX EMPTY.
9. FEATURE NOT IMPLEMENTED – JOB ABORTED
10. <integer> COMPILES EXIST WITH PROGRAM NAME <program-name>, THEIR DETAILS ARE AS FOLLOWS:
11. ENTER COMPILE TAG OF REQUIRED COMPILE, OR NULL ACCEPT TO TERMINATE
 (CLEAR)
 < RESTART > ATTEMPT.
 (MX)
12. COMPILE TAG NOT APPROPRIATE, PLEASE TRY AGAIN.
13. COMPILE WITH PROGRAM NAME <program-name> RESTARTED.
14. COMPILE WITH PROGRAM NAME <program-name> AND COMPILE TAG <compile-tag> RESTARTED.
15. COMPILE WITH PROGRAM NAME <program-name> CLEARED.
16. COMPILE WITH PROGRAM NAME <program-name> AND COMPILE TAG <compile-tag> CLEARED.
17. EXISTING FILE CO.MASTER IS INCOMPATIBLE WITH THIS VERSION OF CO.
18. TO REMOVE CO.MASTER AND CONTINUE, ENTER Y TO ACCEPT. ANY OTHER ENTRY WILL CAUSE CO TO TERMINATE.
19. COMPILE WITH PROGRAM NAME <program-name> AND COMPILE TAG <compile-tag> SHOULD BE RESTARTED OR CLEARED.

ZIP FAILURES

If a zip failure occurs, or a particular compiler pass is DS'ed or DP'ed, CO displays one of the messages in table 6-4, indicating the reason for the failure, then takes one of the following two actions:

If no Restart is in operation or pending:

One of the following messages is displayed:

“CO SHOULD BE RESTARTED OR CLEARED”

“CO MIX-NUMBER n MUST BE RESTARTED OR CLEARED”.

The CO utility is forced into “Restart Mode” which prevents any new COs or any other RESTARTS being performed until the CO in question has been either restarted and completed, or cleared.

If a Restart is in operation or pending:

One of the following messages is displayed:

“CO SHOULD BE RESTARTED OR CLEARED”

“CO MIX-NUMBER SHOULD BE RESTARTED WHEN EXISTING RESTART COMPLETE”.

The CO utility is not forced into “Restart Mode” in this case. It is required to RESTART the job as soon as possible after the existing restart is complete.

Table 6-4. Zip Failure Messages

ZIP FAILURE DUE TO PROGRAM FILE NOT FOUND FOR <program.name>
ZIP FAILURE DUE TO INTERPRETER FILE NOT FOUND FOR <program.name>
ZIP FAILURE DUE TO INSUFFICIENT MEMORY FOR <program.name>
ZIP FAILURE DUE TO NO USER DISK OR DIRECTORY FULL FOR <program.name>
ZIP FAILURE DUE TO FULL MIX FOR <program.name>
ZIP FAILURE DUE TO USER COUNT ERROR <program.name>
ZIP FAILURE DUE TO DUPLICATE PACK <program.name>
ZIP FAILURE DUE TO INVALID LOAD REQUEST <program.name>
ZIP FAILURE DUE TO MCS ALREADY PRESENT <program.name>
ZIP FAILURE DUE TO DISK ERROR <program.name>
ZIP FAILURE DUE TO CODE FILE ERROR <program.name>
ZIP FAILURE DUE TO ILLEGAL DATA COMM LOAD REQUEST <program.name>
COMPILER PASS DS'ed <program.name>
COMPILER PASS DP'ed <program.name>
ZIP FAILURE DUE TO NULL MIX REQUIRED FOR <program.name>
ZIP FAILURE DUE TO DUAL ALPHABET/REVERSE ESCAPEMENT NOT SUPPORTED
<program.name>
ZIP FAILURE DUE TO INSUFFICIENT REAL STORE FOR <program.name>
ZIP FAILURE DUE TO DISK LOCKED FOR <program.name>
ZIP FAILURE DUE, REASON UNKNOWN <program.name>

RESTARTING EXECUTING CO VERSIONS

Note that CO does not prevent restarting or clearing a currently-executing compilation. However, when the original version of the compiler pass, that was executing when the RESTART was done, terminates and returns to CO, CO will recognize that the block of information in the CO.MASTER file has been overwritten. The message:

“CANNOT CONTINUE COMPILATION-COMPILATION
BLOCK INITIALIZED”

is displayed and the job is terminated. This may result in problems such as the occurrence of duplicate files. If an executing compilation is CLEARed, there is an added complication that any workfiles for that compilation are removed.

Restarting or clearing any currently-executing compilation is therefore not advised.

LIST OF RETURN VALUES FOR CO

Byte2	Meaning
@00@	Compilation Successful.
@01@	No Initiating Message.
@02@	Syntax errors in Initiating Message – Compiler not ZIPPED.
@03@	Compiler not Zipped – Reason unconnected with the Initiating Message. For example: 1. All workfile characters marked as in use – none available for this job. The message “CANNOT INITIATE THIS CO – MAXIMUM NUMBER OF COs ALREADY IN PROCESS – JOB TERMINATED” will be displayed. 2. Requested feature not implemented. 3. An error has occurred within CO.
@10@	ZIP Failure when attempting to ZIP Compiler – RESTART/CLEAR required.
@12@	Compiler Pass DS/DP’ed – RESTART/CLEAR required.
@13@	Compilation apparently successful but object file not found for execution.
@14@	Compilation successful – object file found – but ZIP Failure when attempting to execute object program.
@20@	Compilation unsuccessful – message(s) indicating reason for failure output by terminating compiler pass.

RESERVED WORDS

All keywords used in the initiating message to CO are reserved words, that is, they cannot be used for file names or other user-defined parts of the initiating message. A complete list of the reserved words is given in table 6-5.

Table 6-5. CO Reserved Words

MPL		SOURCEOUT	
COBOL		WORKFILE	
RPG		PRINTOUT	
RPGXREF		NAME	
OPTLIST		MFID	
COBOLXREF		FID	
SAVE	(a)	DEVICE	
GO	(a)	FILE.SIZE	(e)
SYNTAX	(b)	FILESIZE	(e)
SY	(b)	RECORD	
LIBRARY	(c)	BLOCK	
LI	(c)	RECORDS.BLOCK	
PRINT		N0.BUFFERS	
MESSAGE		MX	
FILE	(d)	RESTART	
FI	(d)	CLEAR	
PATCHFILE		ALL	
SOURCEIN			
CONTROLFILE			
ICFILE			

NOTE

The pairs of words marked with the same letter, for example, (a), are synonymous.

SECTION 7
NUMBERED SYSTEM SOFTWARE OUTPUT MESSAGES
TO BE SUPPLIED

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).

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 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 power-on sequence are given later in this section, System Load Errors.

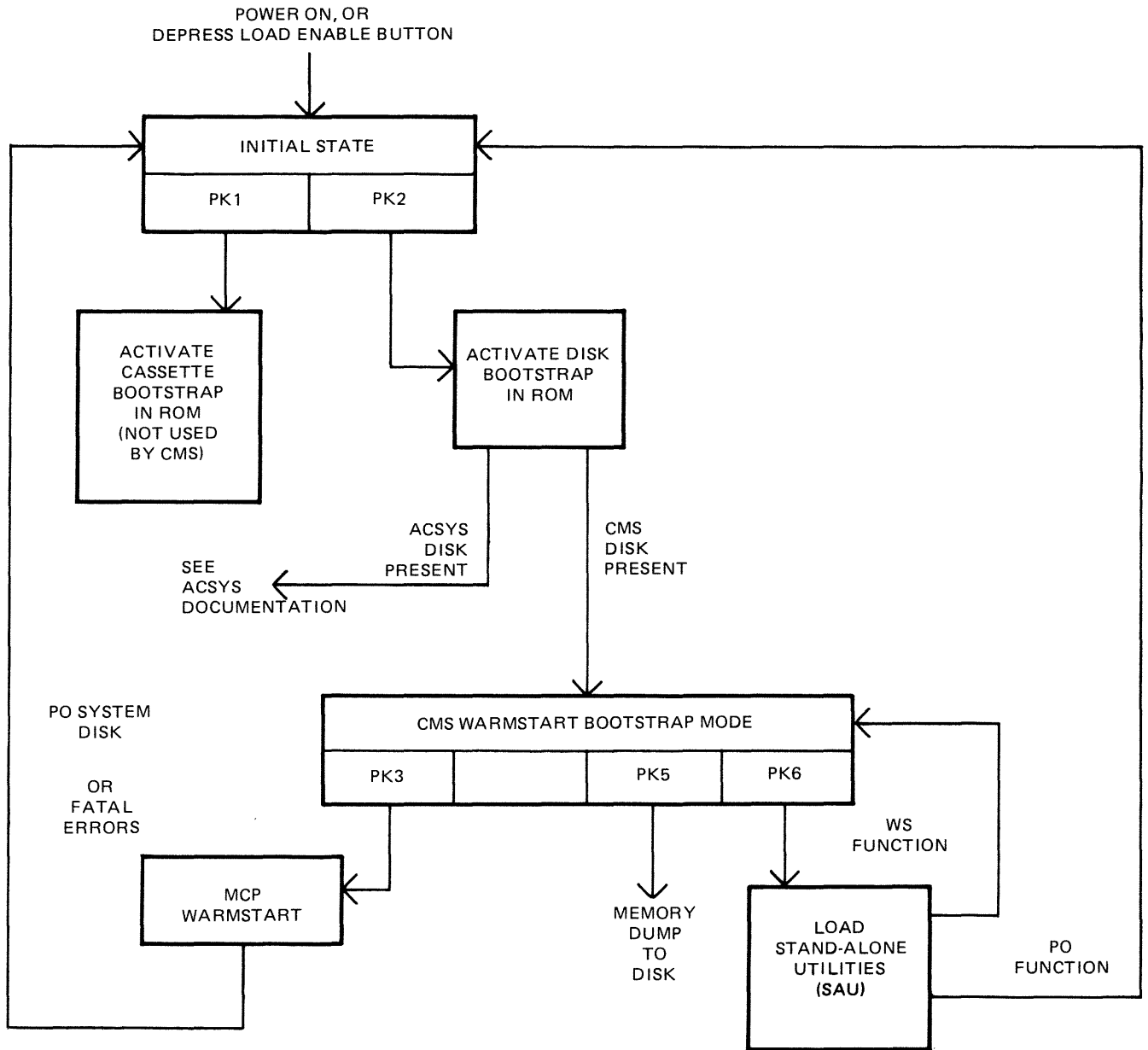


Figure 8-1. B 90 Warmstart ROM Level 4

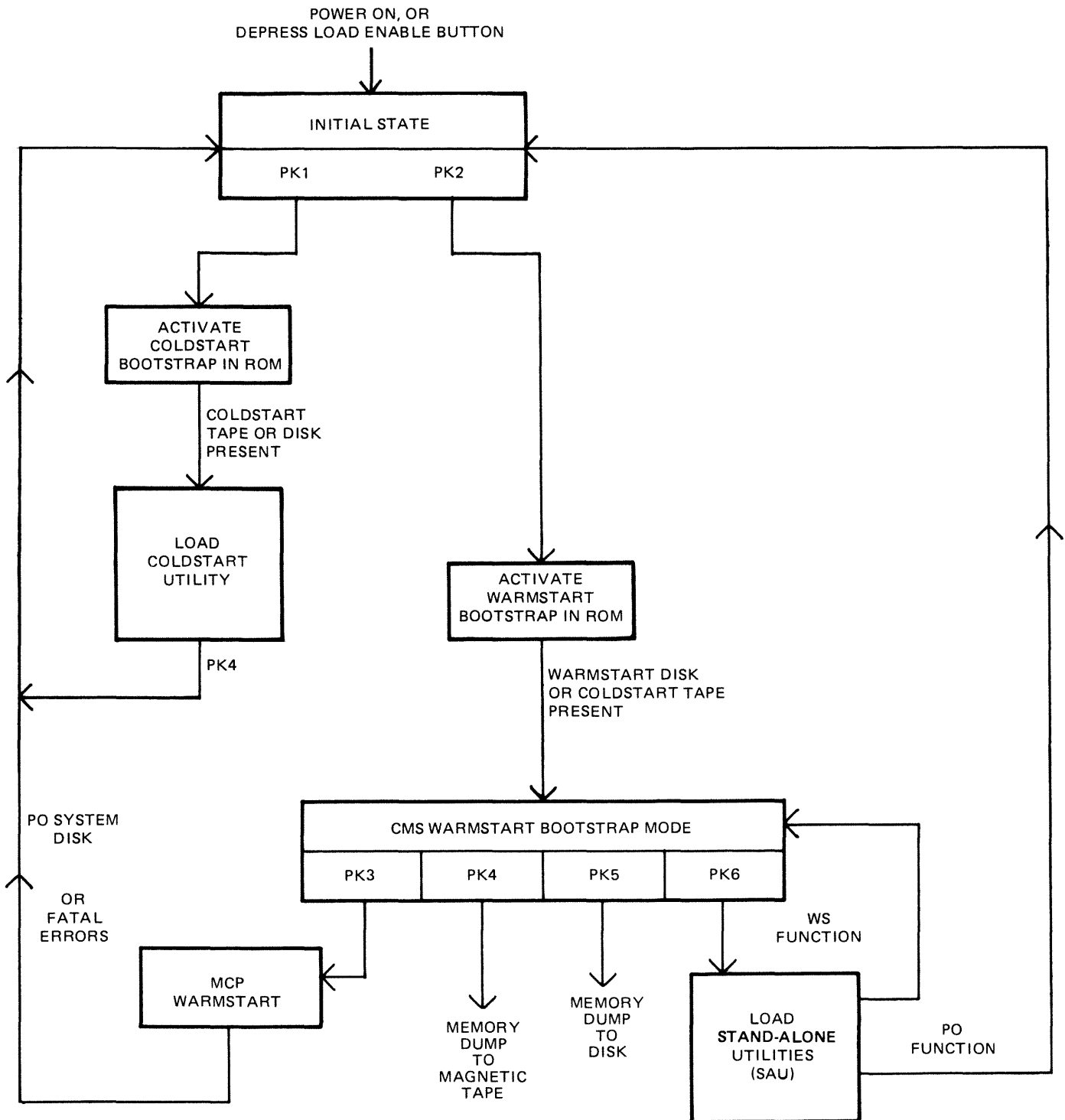


Figure 8-2. B 90 Coldstart and Warmstart ROM Level 5

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).

A Note On 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.

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 COLDTART utility. A minimum of level 5 ROM is required to support the COLDSTART 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 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

Single reel tapes only 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

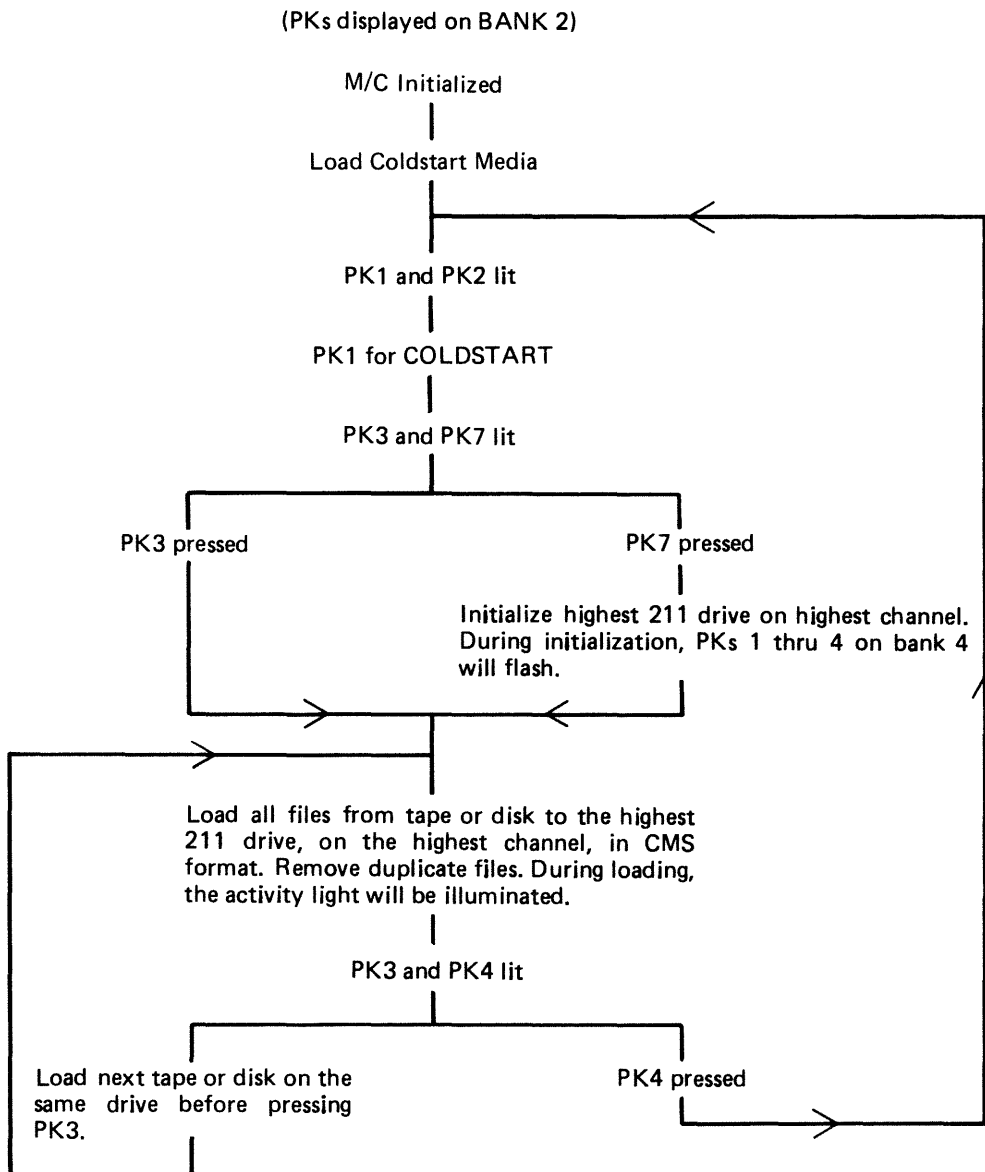


Figure 8-3. Outline of COLDSTART Utility

Initialization Parameters for 211 Fixed Disk

If the 211 is to be initialized, the following pre-defined set of parameters are used :

Disk name	SYSTEM
Number of files	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 (BSMII 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
X O O X O O O O	No tape named “CLDSTRT” on-line (Magnetic tape only).	See note 2

Bank 4 Lights 1 2 3 4 5 6 7 8	(continued) Description	Recovery
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 only).	See note 2
X X X X O O O O	Disk expiring (BSMDII only).	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:

	1	2	3	4	5	6	7	8	
	O	O	O	O	X	X	O	O	BANK 2
MSD									LSD

(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 or a function when SAU requests it.

Terminal Configuration for SAU

SAU requires a TD 830 device or equivalent (MT 983) to be installed on channel 5 at 9600 bits per second using Burroughs Two Wire Direct Interface (TDI) and poll/select line discipline. The physical address of the TD 830/MT 983 must be 01 (ASCII 30 31).

The required TD 830 system registers for use with SAU are as follows:

0080	0A FB 0A 00 17 4F 1F 1E 10 30 31 04 09 00 04 04
0090	04 2F 4C 17 4F FF 20 47 17 40 00 08 03 50 0F FF
00A0	00

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 DEVICE 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 BSMII, F = FIXED, K = CARTRIDGE)

y identifies unit
(A, B, C etc.)

Disk I/O Errors During SAU

These errors are identified by the messages:

O/P ERROR, I/P 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 (I/P error) or write to (O/P 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.

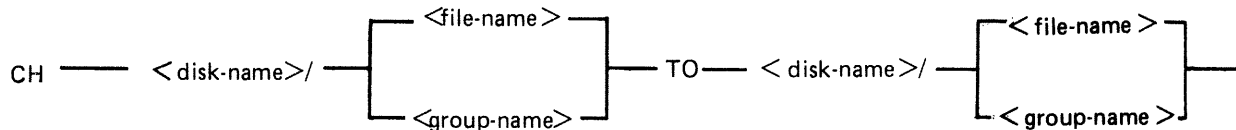
The terms “master file” and “overflow file” are sometimes used to distinguish the two parts of a dual pack file. It should be remembered, however, that either part of the file may be regarded as the master file. In this section the term “master” file is used to indicate the part of the file mentioned 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.

CH (Change disk file name)

This function allows the operator to change the name of disk files.

Railroad diagram:



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 O/P 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 O/P 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.

Message	(continued) 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).

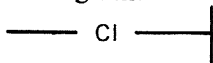
The CI function performs this disk initialization. When all disk parameters have been requested, CI checks the recording surface of the disk by writing and reading test patterns to each sector of the disk. Any "bad sectors" (unusable or unreadable) are removed from the disk's available area table, and 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:

Railroad diagram:



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.

Message	(continued) 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 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 NO OF BAD SECTORS EXCEEDS 50	A hardware error has occurred. This disk is unusable.	Place disk with CMSBOOT into another drive and try CI again. Select another disk to be initialized.
NO COLDSTART BOOTSTRAP IN CMSBOOT, LOAD CORRECT FILE AND HIT OCK1/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.

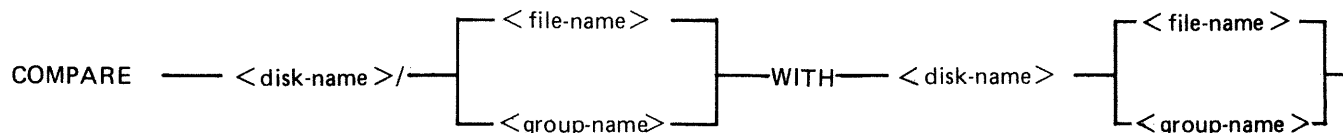
WARNING

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.

Railroad diagram:



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

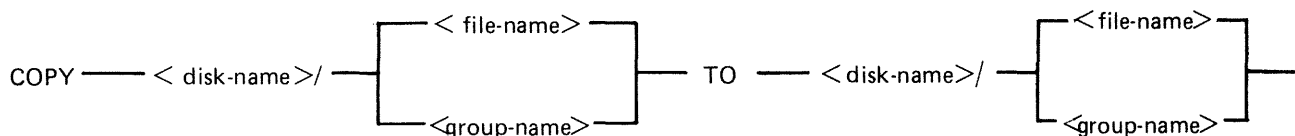
Message	Possible Cause	Suggested Action
file-name NOT FOUND	Self-explanatory.	Check input.
FILE SIZE CONFLICTS – END COMPARE	One file contains more records than the other.	None.
CONFLICTIONS IN NUMBER OF AREAS – END COMPARE	The number of areas assigned to the two files are different.	None.
disk-name/file-name 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.

Message	(continued) Possible Cause	Suggested Action
file-name NOT FOUND file- name NOT COMPARED INCOMPLETE DUAL PACK FILE	Comparison of dual pack file was required and the overflow pack was present but did not contain the required file.	Investigate why the file was not present on the overflow pack.
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.

Railroad diagram:



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.

Message	(continued) Possible Causes	Suggested Action
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 input (source) disk may be corrupted.	1. Change the position of the disks and retry. 2. For a group copy, some files may be recovered using single file copy. If these attempts fail, then type of error should be analyzed with DA utility and disk re-initialized before re-use.
O/P 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 END COPY	Typing error. Copy successful (EOJ); Specified group name to be copied is not on disk.	Check input and re-enter. None if copy successful; Check input – re-enter if necessary; Check for correct disk.
file-name COPIED CANNOT ALLOCATE AREAS FOR – file-name	COPY successful. Insufficient available space on disk to which operator is attempting to copy this file.	None. 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.
I/P 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.

Message	(continued) Possible Causes	Suggested Action
file-name NOT COPIED: ILLEGAL FILE-ID	Specified file-name contains an illegal character (that is, / ?), therefore file will not be copied. Legal characters are 0-9, A-Z, . (dot), - (dash).	Check input and re-enter if necessary.
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 (BSMII) to another BSMII.

Format:

DISCOPY DM_x TO DM_x
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 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".

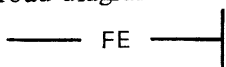
The "xxxxx" characters are ignored by the utility; only the first seven characters are compared when the on-line disks are searched (that is, the file specifications searched for are equivalent to CMSBOOT=).

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:

Railroad diagram:



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.

Message	(continued) 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 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.
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.

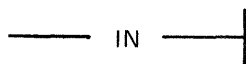
The function also writes a disk label containing information supplied by the operator to the appropriate prompts (see Output Messages), and creates a disk directory of the appropriate size required for the number of files specified, plus a single, SYSMEM, entry. Sectors 1 through 31 on track zero are 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:

Railroad diagram:



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).

Message	(continued) 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 NO OF BAD SECTORS EXCEEDS 50	A hardware error has occurred. This disk is unusable.	Place disk with CMSBOOT into another drive and try IN again. 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 HIT OCK1 OR HIT ANY KEY THEN OCK1 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.

Message	(continued) 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.

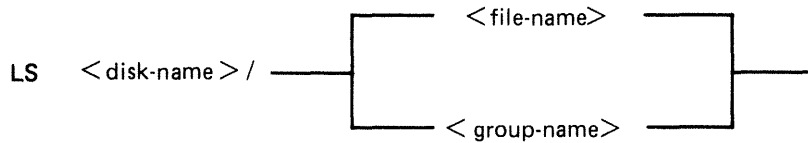
LS (List File Sizes)

This function allows the operator to print name 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 “UN-LABELLED”.

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 Message

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:

RL AP2

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.

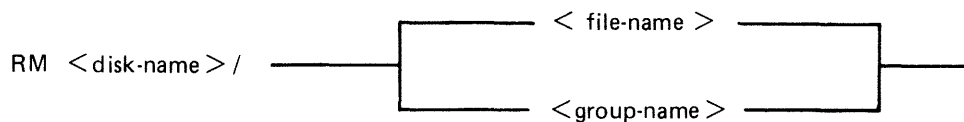
There are two ways to resolve the problem if it does occur:

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.

Railroad diagram:



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) lights 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:

```
B90 CMS MCP 03.04.12 82116 + + + + + + + + + + +
... + + +
DMA SYSTEM1/ SYS DISK 2 FILES OPEN
DMB M1/ 0 FILES OPEN
ENTER DATE AS MM/DD/YY
5/25/82
25 MAY 82 82145
COMMENCING LOG FILE CONSOLIDATION
TRANSFER COMPLETED
LOGGING IS INITIATED ON 05/25/82 (MCP VE
...RSION 03.04.12)
```

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.

The MCP has to be told which terminal to use as the SPO terminal. This is achieved by typing in (to a ready terminal) “?.LI PASSWORD” at the top left hand corner. Transmit using the XMT key. Note that “PASSWORD” can be changed, and the current password should be used instead. The following is a sample output:

```
SPO INTERFACE MODULE – RELEASE 03.04.08 – COMPILED 82026
STATION TD830XA01 LOGGED-IN AS SPO
WARNING: DATE HAS NOT BEEN INITIALISED
```

```
B90 CMS MCP 03.04.12 82116 + + + + + + + + + + + + + + + + +
```

```
LOGGING IS INITIATED ON 11/11/11 AT 00:01:23 MCP VERSION 03.04.12)
```

```
DT 5/25/82 0900
```

```
25 MAY 82 82145 TUE 1545
```

```
(operator input date and time)
```

```
(MCP output to date and time)
```

SCL Page 2

NOTE

If the terminal has a two page capability, the user may move from the SPO page to the application page by pressing the SPCFY key. To move from the application page to the SPO page, the user presses the SPCFY key.

For possible system error conditions, see System Load Errors and Hardware Error Conditions later in this section.

BASIC OPERATION UNDER MCP CONTROL

D-Lights (Bank 5)

During MCP execution, D-lights D1 to D8 (Bank 5) give an indication of system activity, as shown in figure 8-4.

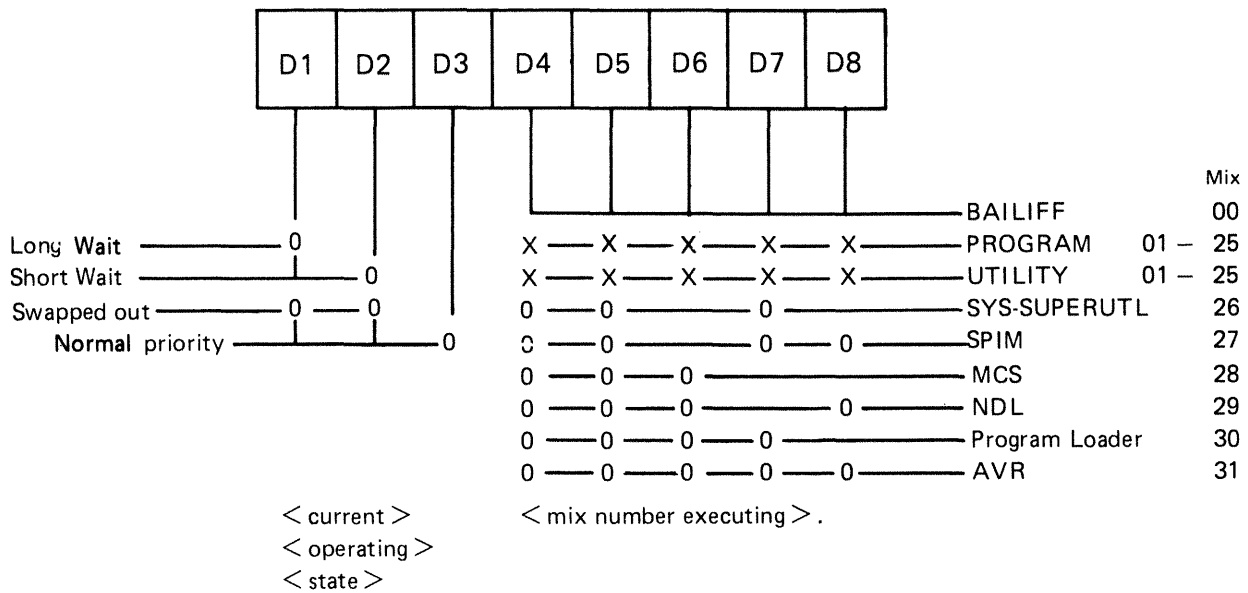


Figure 8-4. D-Light Display Interpretation

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 immediately 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 (2011 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 drive 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 rewind.)

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.

Hardware Error Conditions

If a hardware error is detected while the ROM is scanning or loading a bootstrap, the ERROR light is lit and the channel number is displayed on PK1 through PK8 (Bank 2) by illuminating all the PKs except the one representing the channel number. The system waits for keyboard input.

Similarly, if a hardware error occurs while a Bootstrap is searching for, or loading, a file, the ERROR light is lit and the channel is displayed on PK1 through PK8 (Bank 2) by illuminating the appropriate PK. The system waits for keyboard input.

The following table shows valid keyboard input.

Key	ERROR Lit	Bank 2 Displays (PK1..8)	Bank 4 Displays (PK17..24)
OCK1	ON	Primary device status	Secondary device status 1
OCK2	ON	Secondary device status 2	Secondary device status 3
OCK3	ON or OFF	Not used	ROM version number
OCK4 (i)	ON or OFF	BOOTSTRAP release level	BOOTSTRAP mark level
OCK4	ON or OFF	Not used	BOOTSTRAP patch level
RESET	OFF	Not used	Not used

NOTE

Pressing the RESET key causes the scan to continue at the next lowest numbered drive or channel. If the RESET key is pressed when no Bootstrap/file has been found, the scan process re-starts from channel 7.

Bootstrap version numbers are displayed in three parts "MARK.RELEASE.PATCH". If the Bootstrap is loaded, press OCK4 to display the two major components. Pressing OCK4 a second time displays the least significant component on Bank 4.

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 RESET after COLDSTART not found on hardware error have 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 then 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 a 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.

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 "Hardware Error Conditions" and "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

The system must be in the CMS Bootstrap state. If the system has clear-started, press PK2 to get to the CMS Bootstrap state (PK3 to PK6 lit).

Press PK5. The system searches for a disk file called MEMDUMP. The search proceeds as outlined in Bootstrap Scanning Algorithm later in this section.

The utility GEN.DUMPFL may be used to create a suitable file on the system disk. This must have been done before a dump to disk is attempted. It is recommended that for minidisk-based systems a spare minidisk is kept with a MEMDUMP file on and this disk is the only disk loaded when the memory dump to disk is taken. This avoids any possible confusion between MEMDUMP files on different disks.

The contents of RAM are written to the MEMDUMP file. At the end of the dump, the CMS Bootstrap state is entered (PK3 to PK6 lit).

For possible errors see System Load Errors.

NOTE

Valid memory dumps are produced only if the pertinent error condition was the last event on the system. Invalid memory dumps occur if the machine main cabinet has just been switched on, or the error condition did not occur under MCP control, or PK1 was pressed while in initial state (PK1 and PK2 lit).

SYSTEM LOAD ERRORS

The following table of errors covers all symptoms found during start-up of the B 90, 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	No bootstrap found (disk not initialized correctly)	Press RESET to continue scan.
(b) One of PK1-PK8 (Bank 2) extinguished, all other (Bank 2) lit	Disk media or drive fault.	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.

Symptom	(continued) Possible Causes	Suggested Action
Memory dump to disk: system warmstarts.	PK3 was depressed instead of PK5.	Memory dump cannot now be taken. Allow warmstart to complete.
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.	Check forms transport closed. Check printer cover down. Check levers in correct position. Utility should continue.

Diagnosis of Disk Errors at System Load Time

Caelus Device Primary Status

Bit	Interpretation
0	Drive / ddressed (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 incomplete
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	1 = Tape corrected error	Transport address *	1 = Formatter busy *
1	1 = Tape hard error	Transport address *	1 = Data busy *
2	1 = Parity error from formatter	1 = Write Inhibited tape	1 = high speed 0 = low speed
3	1 = Controller detected error	1 = Drive online	Tape streaming *
4	1 = 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.
	Extended memory not correctly used.	Run CONFIGURER and warmstart if necessary.
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.

Symptom	(continued) Possible Causes	Suggested Action
Logging not taking place.	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 (for example, on a monthly basis).
	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.INTERP not present on system disk.	Make sure the correct interpreter (for example MICROPOL) is named NDL.INTERP and located on the system disk.
	No MCS program.	Ensure that the correct MCS is executing as well as any user data comm program.
Degraded performance	Lack of memory due to incorrect use of Extended memory, wrong SYSCONFIG file on system disk.	Run CONFIGURER, repeat after warmstart.
	Lack of memory due to memory fault at warmstart time.	Call field engineer.
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.

Symptom	(continued) Possible Causes	Suggested Action
System not active : programs in mix but D-light pattern does not change; keyboard input (for example, MX) causes error bell to ring.	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 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, call field engineer.
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. Call field engineer.

Symptom	(continued) Possible Causes	Suggested Action
Non-functional disk drive.	Hardware fault on drive.	Load a good disk on drive (EXCEPT IF DRIVE HEADS HAVE CRASHED), run 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.
	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, send the KA plus recent SPO message log to field engineer. Run DA to give further analysis of disk.
		Use backup disk for system files. Re-cover any other files by attempting to COPY then one-by-one from the corrupt disk to a good disk.

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

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 C_i , 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 a 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

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

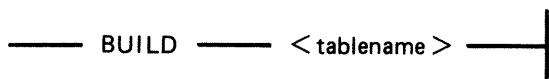
The output will be directed to the device from which the function was requested. However, by specifying PRINT, output will be directed to a hard copy device.

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

Railroad diagram:



BUILD allows a new user table to be defined, providing that a table of that name does not already exist in the current workfile.

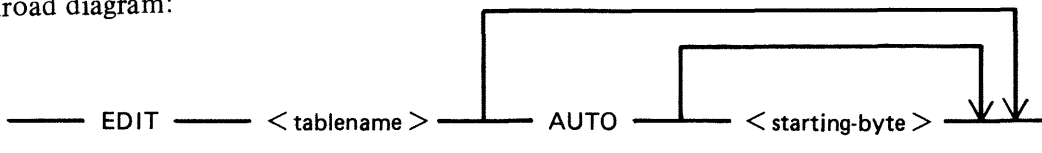
Details about the translation mode and, if using mode 1, the length of the translation elements, are requested.

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

Railroad diagram:



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.

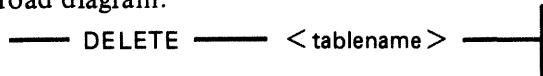
NEXT causes the current entry to be retained and the next value to be translated will be output. If the AUTO option is not selected, then two hex numbers must be entered, separated by at least one space, after BUILDTRANS has issued a prompt. The first number is taken as the byte to be translated,

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

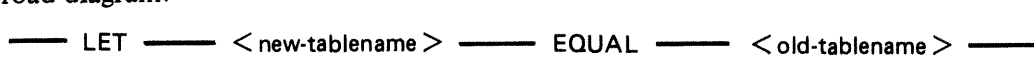
Railroad diagram:



DELETE removes the specified user table, if it exists, from the workfile.

LET

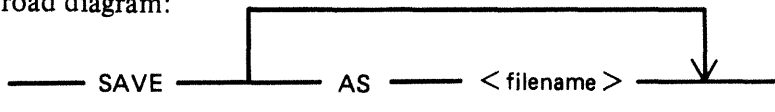
Railroad diagram:



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

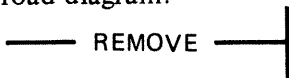
Railroad diagram:



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

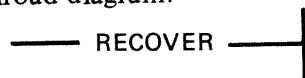
Railroad diagram:



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

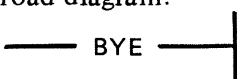
Railroad diagram:



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

Railroad diagram:



BYE causes BUILDTRANS to go to EOJ.

BUILDTRANS Message Format

Message	Description
TEXT LENGTH EXCEEDS UTILITY LIMIT	An attempt has been made to input text which exceeds the restrictions for this input device.
SYSTEM DISK LOCKED or DISK <disk-name> LOCKED or FILE <file name> NOT AVAILABLE - DISK LOCKED or WORK FILE NOT AVAILABLE - DISK LOCKED	Any one of these messages may be output if the disk which BUILDTRANS is attempting to access has its SYSMEM file locked by some other program.
WORK FILE(S) ALREADY IN USE	BUILDTRANS has found a file present with the same name as its workfile but the file is not available for use by this copy of BUILDTRANS.
RECOVER OR REMOVE REQUIRED	BUILDTRANS has found a file with the same name as its workfile. The user has the option to RECOVER and continue from a previous session, or to REMOVE the file and start again.
RECOVERY COMPLETED	RECOVER function has successfully recovered the workfile.
TRANSLATION VALUES ARE <value>	The EDIT function is recovered by the workfile.
FILE <filename> HAS BEEN ALTERED, WORK FILE(S) BEING REMOVED	The RECOVER function has been unable to recover the workfile. The user can only do a GET, MAKE or BYE.
<file-name> ALREADY PRESENT, SAVE OR REMOVE REQUIRED	The user has attempted a GET or MAKE after modifying the current file and before SAVING or REMOVING it.
TRANSLATION VALUES ARE <value> BYTE(S)	The EDIT function has been requested on a table of type 0 or 1. For these modes, the correct number of bytes must be entered for the entry to be accepted.
ENTER TRANSLATION MODE	This must be in the range 0 to 2 (entered as a decimal value).
ENTER MAXIMUM SIZE OF TRANSLATION VALUES	This number is the maximum translation entries for a single internal code in a mode 1 table. This value is defined to be in the range 1 to 10 (entered as a decimal value).
ENTER ANALYSIS MODE	This value is defined to be in the range 0 to 255. (entered as a decimal value). This is for mode 2.
ENTER SHIFT CODES	Exactly two bytes of hexadecimal information is required. Each shift code is 1 byte in length. The first byte is the "shift out" code the second is the "shift out" code.
ENTER MACHINE RESTRICTION (1=B 90, 2=B 900, 3=B 1800)	Up to three different values from the stated range can be entered in any order. The numbers are separated by spaces.

(continued)

Message	Description
<offending option> ALREADY SET	Duplicate machine restriction has been entered. Request is re-issued.
<table name> IS A SYSTEM TRANSLATION TABLE	The user has attempted to DELETE a table which is defined to be a SYSTEM table.
WARNING – TABLE <table name> 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 realize that the table is “addressed” by other directory entries as well as the one specified in the EDIT request.
NUMBER OF TABLES <value> EXCEEDS UTILITY LIMIT <value> or NUMBER OF DIRECTORY ENTRIES <value> EXCEEDS UTILITY LIMIT <value>	Either one of these messages can be output if the user has generated a large number of entries without requesting a SAVE. Requesting a SAVE and then a GET will overcome the problem.
SIZE OF TABLE ENTRY (<value> BYTES) EXCEEDS UTILITY LIMIT (<value> BYTES)	A table of mode 1 or 2 has been discovered which has entries greater than the utility can handle.
***** NO ENTRIES IN TABLE *****	There are no entries in the mode 2 table being listed. BUILDTRANS has processed the table and discovered that all the entries are zero length.

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 92494, B 924930
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 ODEC/4 line printer : LP.B92494.GRMNY.V5
6. Version 6 3050 line printer : LP.B924930.BRZIL.V6
7. Version 7 keyboard : KB.KB.SPAIN.V7

CONFIGURER (Configure B 90 System)

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.

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? –
<Y OR N>
This requests 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 3.04, see section 4, LOGGING IN THE 3.04 RELEASE. Valid responses are : "Y" (yes) or "N" (no). If the response is "N" goto 2.0. |
| 1.1 | £ | SPO LOGGING IS SWITCHED ON
or
SPO LOGGING IS SWITCHED OFF |
| 1.1.1 | £ | DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is "N" goto 1.2. |
| 1.1.2 | U | SPO LOGGING REQUIRED? – <Y OR N>
This requests whether or not logging of system messages is required at the next warmstart. |

-
- Valid responses are : “Y” (yes) or “N” (no).
If the response is “N” goto 1.3.
- 1.2 £ LOGGING OF SYSTEM MESSAGES IS CYCLIC
or
LOGGING OF SYSTEM MESSAGES IS NON-CYCLIC
depending upon the current form that system messages are to be
logged in.
- 1.2.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 1.3.
- 1.2.2 U IS LOGGING OF SYSTEM MESSAGES TO BE CYCLIC? – <Y
OR N>
This requests whether system messages are to be logged in CYCLIC
or NON-CYCLIC form.
Valid responses are : “Y” (cyclic) or “N” (non-cyclic).
- 1.3 £ CURRENT SIZE OF PRIMARY LOG FILES IS <integer>
SECTORS
(where <integer> = the size, in sectors, of the 4 primary log
files).
- 1.3.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 1.4.
- 1.3.2 ENTER SIZE OF PRIMARY LOG FILES IN SECTORS
<MINIMUM 32, MAXIMUM 16383>
This requests the size, in sectors, of the 4 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 1.5.
- 1.4.2 U ENTER SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IN
SECTORS <MINIMUM 32, MAXIMUM 262140>
A valid response is an <integer> in the range 32 to 262140.
- 1.5 £ CURRENT SIZE OF SECONDARY MAINTENANCE ENTRY
FILE(S) IS <integer> SECTORS
- 1.5.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 1.6.
- 1.5.2 U ENTER SIZE OF SECONDARY MAINTENANCE ENTRY FILE(S)
IN SECTORS <MINIMUM 32, MAXIMUM 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 2.
- 1.6.2 U 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> or
 2. <blank>, in which case backup log files will reside upon 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” 470.
- 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 ID OF MESSAGE FILE IS <file-name>
where <file-name> is the name of the current local language message file. See TRANSLATOR for creation of customized language message file, if the standard language message file, SYSLANGUAGE, is not suitable.
- 4.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 5.0.
- 4.2 U ENTER FILE-ID OF REQUIRED MESSAGE FILE
This requests the <file-id> of the required local language message file.
Valid responses are :
1. a <file-id> or
2. <blank>, in which case the message file will be “SYSLANGUAGE”.
- 5.0 £ DO YOU WANT TO CHANGE ANY PRINTER BACKUP OPTIONS? – <Y OR N>
This requests whether or not items in the SYSCONFIG file concerned with printer backup files are to be updated.
Valid responses are : “Y” (yes) or “N” (no).
If the response is “N” goto 6.0.

-
- 5.1 £ CURRENT PRINTER BACKUP FILE SIZE IS <integer>
RECORDS
(where <integer> is the maximum size, in records, of a printer
backup file)
- 5.1.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 5.2
- 5.1.2 ENTER PRINTER BACKUP FILE SIZE IN RECORDS
<MINIMUM 1, MAXIMUM 1048560>
This requests the maximum size, in records, of printer backup files.
A valid response is an <integer> in the range 1 to 1048560.
- 5.2 £ PRINTER FILES ARE SENT TO BACKUP
or
PRINTER FILES ARE NOT SENT TO BACKUP
- 5.2.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If response is “N” goto 5.3.
- 5.2.2 ARE PRINTER FILES TO GO TO BACKUP? – <Y OR N>
This requests whether or not printer files are to go to backup.
Valid responses are : “Y” (yes) or “N” (no).
- 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 6.0.
- 5.3.2 ENTER ID 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> or
2. <blank>, in which case printer backup files reside on the system
disk.
- 6.0 £ DO YOU WANT TO CHANGE ANY SPO OPTIONS? – <Y OR
N>
This requests whether or not items in the SYSCONFIG file
concerned with the local SPO and the remote SPO are to be
updated.
Valid responses are : “Y” (yes) or “N” (no).
If the response is “N” goto 7.0
- 6.1 £ DOES THE SYSTEM HAVE A LOCAL SPO? – <Y OR N>
This requests whether or not a local SPO will be present at the next
warmstart.
Valid responses are : “Y” (yes) or “N” (no).
If the response is “N” goto 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 6.2.
- 6.1.3 ENTER LOCAL SPO OPTION – <SCA OR SPA>
This requests whether the local SPO is a SCREEN (SCA) or a SERIAL PRINTER (SPA).
Valid responses are : “SCA” or “SPA”.
The final choice of the type of local SPO will be determined by the MCP at warmstart, if the user’s choice is not applicable to the machine’s configuration.
- 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 WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 7.0.
- 6.2.2 DOES THE SYSTEM HAVE A REMOTE SPO? – <Y OR N>
Valid responses are : “Y” (yes) or “N” (no).
- 6.2.3 £ CURRENT REMOTE SPO MCS IS x
(where x is the file-id and initiating message of the remote SPO’s MCS)
- 6.2.4 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N”, goto 7.0.
- 6.2.5 ENTER FILE-ID 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.
OR
2. <blank>, in which case the MCS and initiating message will be “MCSOBJ DC CC”.
- 7.0 £ DO YOU WANT TO CHANGE ANY TRANSLATION TABLES?
– <Y OR N>
This requests whether or not the current set of translation tables is to be updated.
Valid responses are : “Y” (yes) or “N” (no).
See BUILDTRANS utility for details about translation.
If the response is “N” goto 8.0
- 7.1 £ CURRENT NUMBER OF TRANSLATION TABLES IS <integer>
- 7.1.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 7.2 if <integer> greater than zero or 8.0 if <integer> equal to zero.

-
- 7.2 ENTER NUMBER OF TRANSLATION TABLES <MAXIMUM 255>
A valid response is an <integer> not greater than 255.
- 7.3 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.
- 7.3.1 £ CURRENT DEVICE KIND AND NUMBER IS abn
abn is the CMS DEVICE KIND and DEVICE NUMBER of a translation table.
ab is a CMS DEVICE MNEMONIC.
n is a DEVICE NUMBER.
- 7.3.2 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 7.4
- 7.3.3 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 :
1. ab is a CMS DEVICE MNEMONIC and
2. n is a DEVICE NUMBER so that “A” <= n <= “Z”.
Valid DEVICE MNEMONICS are: “SP”, “LP”, “SC” and “KB”.
- 7.4 £ CURRENT TRANSLATOR NAME IS <identifier>
- 7.4.1 £ DO YOU WISH TO CHANGE IT? – <Y OR N>
If the response is “N” goto 8.0.
- 7.4.2 ENTER NAME OF TRANSLATOR
A valid response is an <identifier> of not more than 20 characters.
If all tables have been completed, goto 8.0, otherwise goto 7.3
- 8.0 JOB COMPLETED
This is the normal job terminating message. The utility has now released the file under the name “SYSCONFIG” onto the system disk.

Error Message	Meaning	Response
** INVALID SELECTION – RETRY **	Self explanatory. Request is repeated.	Use a valid response, see request.
ILLEGAL FILE SIZE – x	x is an illegal response. Request will be repeated.	Use a valid response, see request.
INVALID CHARACTER IN IDENTIFIER <identifier>	<identifier> has invalid character or characters in it. Request is repeated.	Valid characters are “A” – “Z”, “0” – “9”, “.”, “-”.
ILLEGAL – x <request is repeated>	x is an illegal response	Use a valid response, see request.

Error Message	(continued) Meaning	Response
TEXT LENGTH EXCEEDS UTILITY LIMIT <request is repeated>	More than 80 characters in text.	Re-enter text cut down to 80 characters or less.
ILLEGAL NUMBER OF TRANSLATION TABLES – x <request is repeated>	x is an illegal response.	Enter an integer in the range 0- 255.
I/O ERROR DETECTED IN WRITE TO FILE <file- name>	An I/O error has occurred when writing to a file. The utility terminates and no NEW SYSCONFIG file is created.	
I/O ERROR DETECTED IN READ FROM FILE <file- name>	An I/O error has occurred when reading from a file. The utility terminates and no NEW SYSCONFIG file is created.	
ERROR AT RECORD NUMBER <integer>	An I/O error has occurred at record number <integer>	

B 90 DUMPANALYSE

The B 90 DUMPANALYSE utility is for use by CMS COBOL, RPG and MPL programmers. It provides a method of obtaining B 90 program dumps from programs that have been abnormally terminated and gives an analysis to be used as an aid to locating logic errors.

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 B 90 CMS MCP Reference Manual (form number 2007555).

NOTE :

1. Program dump analyzers are not common items to all CMS products.
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

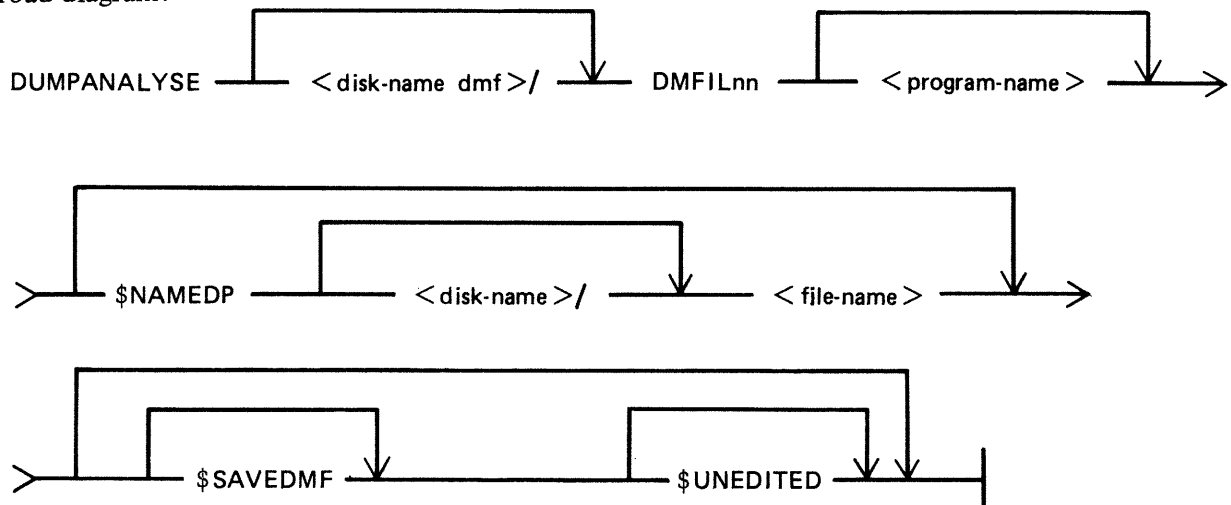
Alternatively, if a program is known to be looping, a DP of the program can be forced in order to identify the conditions associated with the loop.

The dumpfile 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, and the overlayable segments of code and data and certain memory tables used by the MCP for this task (the locked slice).

Dumpanalyse produces a printed analysis of this dump file. This is achieved by using the following initiating message:

Railroad diagram:



disk-namedmf	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.
\$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.

Examples

1. DUMPANALYSE DMFIL01 AM202

Here, DUMPANALYSE analyzes the dumpfile of the DP'd task 01/AM202. The dump file is removed when DUMPANALYSE has finished using it.

2. DUMPANALYSE DMFIL01 AM202 \$SAVEDMF

Here, DUMPANALYSE analyzes the dump file of the DP'd task 01/AM202 and the dump file is not removed when DUMPANALYSE has finished using it.

NOTE

Because DUMPANALYSE has the capability of opening codefiles OTHERUSE LOCK.ACCESS, multiple executions of the utility are possible on the same file.

Analysis Common to all Program Languages

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> consists of 32 hex byte values divided into groups of four. 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 contain 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 below.

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 B 90 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. This field does not appear in a COBOL dump.

Reason for Dump

This is the decoded reason for the abnormal termination of the task.

Task Name

This is a task originator identifier of the aborted task. A breakdown of this byte could be useful to the user. This field does not appear in a COBOL dump.

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) is analyzed. The file-name is printed in the heading for each analyzed FIB. (Details of files and file attributes can be found in the

B 90 CMS MCP Reference Manual, form number 2007555).

NOTE : Data comm is not handled on a file basis.

Closed Files

The only information given for closed or half-closed files is the data segment number of the FIB and the data segment number of the File Parameter Block (FPB) as well as the file state (closed or half-closed).

Open Files

If the file was open at the time of the dump, as well as the information given for a closed file, the following information is supplied:

1. file state

- usage (input, output or both)
- organization
- access mode
- end-of-file encountered

2. work area

If a work area exists then:

- 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

3. records per buffer

This is the number of records contained in a buffer or block.

4. record size

This is the size of records in bytes.

5. block size

The size of the buffer or block in bytes.

6. buffer pointer

Holds an offset into the current buffer to the next available record or for stream access mode to the next available byte.

7. device kind

As defined in the CMS MCP Reference Manual (form number 2007555).

Analysis of disk files provides the following extra information:

8. sectors per buffer

Self explanatory.

9. maximum record written

This is the actual number of records in the file.

10. maximum record declared

This is the maximum number of records the file can contain (maximum file size).

11. disk address details

Unit and sector numbers (absolute sector address).

12. area number log being processed

This number relates to the entry in the disk file header for the particular disk area in use at the time of the dump (a disk file can have a single area or be spread over a maximum of 16 areas).

13. for stream access files

- 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:

other types of access

- current record number
This is the logical record number at the start of the current buffer.

Finally a dump is printed of all the file buffers which exist. 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 hex dump of the locked slice for this task. It contains the Task Control Block (TCB), FIBs and all locked segments.

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 B 90 CMS COBOL Reference Manual (form number 2007266).

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.

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. Each new code segment number which is not on a linear search through the text is allocated the next code segment.

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
UN\$4 – 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

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 above 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.

For more detailed information on these dollar options, refer to the CMS RPG Language Reference Manual (form number 2007274).

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 con-

sole 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 (form number 2007266), CMS RPG Language Reference Manual (form number 2007274). 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 SEG-MAP 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.

8. 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.

9. FETCH VALUE

This is a 24-bit register which holds the result of the previous MCP communicate (the FCM).

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.

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, except for the registers REG1 AND REG2. Their functions are similar to that 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).

Where the DUMPANALYSE listing is requested using this namefile, no actual COP table will be listed. However the DMFILnn file can be saved after using DUMPANALYSE by use of the SAVDMF option in the initiating message. The file will be sequential with 60 byte records. It will consist of several record types as described below.

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)

(continued)

Bytes	Content
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

WARNING BIL DUMP FL, SECOND PARAMETER IN INIT.MESS IGNORED	Program name has been specified for a dump file from an MPL program.	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.

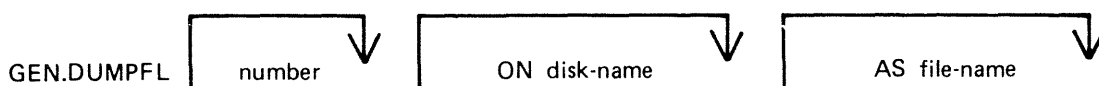
NOT FOUND <reason>	A specified file could not be accessed. <reason> is an interpretation of the fetch value returned to DUMPANALYSE specifying why the file could not be accessed.	None.
MFID NOT PRESENT FILE IN USE, OTHERUSE CONFLICTS BAD FILE TYPE DISK NOT FOUND DUPLICATE PACK DISK ERROR DISK LOCKED 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 SUPPORTED". DUMPANALYSE will attempt to generate an unedited dump.	None.
NO SPECIFICATION GIVEN	This is output when the initiating message given is incomplete.	Check input and re-enter.
NO USER DISK	The MCP is unable to load the utility.	Use backup copy of the utility.

GEN.DUMPFL (Create Empty B 90 Memory Dump File)

Before the contents of memory can be dumped to disk, an empty disk file called MEMDUMP must be created. This file must be large enough to take the contents of all the memory of the system on which the dump is being taken.

The size of the file is specified by “number” in the range 64 to 11492. If “number” is not given, the size defaults to 1024, which corresponds to 1024K bytes of memory. The contents of the created file are initialized, and a repeating pattern of ASCII A through L is written throughout each record of the file.

Format diagram:



Examples:

To create an empty file called MEMDUMP on the system disk for a B 90 with 80K bytes of memory:

```
GEN.DUMPFL 80
```

To create an empty file called MEMORY.DUMP on disk PRB for a 128K B 90:

```
GEN.DUMPFL 128 ON PRB AS MEMORY.DUMP
```

Note that this file-name must be changed to “MEMDUMP” before a successful memory dump can be taken. Any existing file of the same name will be removed when the new file is created by this utility.

Output Messages

Message	Possible Causes	Suggested Action
SPACE RESERVED	Successful EOJ	None.
SIZE TOO SMALL	“number” is less than 64.	Re-input.
SIZE TOO LARGE	“number” is greater than 11492.	Re-input.
DISK NAME TOO LARGE	Disk name is greater than 7 characters.	Re-input.
FILE NAME TOO LONG	File name is greater than 12 characters.	Re-input.

NOTE

If no number is specified in the initiating message, then 1024 will be assumed. This creates a dumpfile for a 1 Megabyte system.

INSTALL

(Copying the contents of an executing Bootstrap disk to a fixed disk).

The INSTALL utility provides a simple method of copying a Bootstrap system disk to any disk except 1Mb and 3Mb disks. This is achieved by having the INSTALL utility resident on a Bootstrap disk which is a Burroughs Mini disk. When INSTALL is executed, it copies the contents of the Bootstrap system disk to the named disk.

Syntax:

INSTALL <disk-name>

where <disk-name> is the name of the disk to which the files are to be copied.

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 (see Install Environment Error Messages) and the utility terminates.

Providing conditions 1 through 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, one of the messages shown in Install Copy Error Messages 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 remove the duplicate file on the output disk. The following message is displayed for each file removed from the output disk:

<file-name> REMOVED

When the copy phase is complete, the following message is displayed:

COPY COMPLETED

Before terminating, INSTALL displays one of the following messages recommending the appropriate action to be taken:

1. If all the files have been copied successfully:
“NOW POWER OFF THE SYSTEM DISK AND RE-WARMSTART FROM DISK <disk-name>”
2. If one or more files have not been copied successfully:
“THE FOLLOWING <number> FILES WERE NOT COPIED – <list of file-name(s)>”

“PLEASE USE ‘COPY’ TO COPY THESE FILES FROM SOME OTHER DISK TO DISK <disk-name> BEFORE POWERING OFF DISK AND WARMSTARTING FROM DISK <disk-name>”

where <disk-name> is that supplied in the initiating message.

Install Environment Error Messages

These are error messages produced due to unsuitable environment.

The utility terminates after displaying one of the following messages, together with the message:

INSTALL TERMINATED

Error Message	Description
THE SYSTEM IS NOT A B90	INSTALL can only be run on a B 90.
THE SYSTEM DISK IS NOT A MINI	The system disk must be a Burroughs Mini disk.
SPECIFIED OUTPUT DISK <disk-name> IS A MINI	The output disk can only be one of the following types: a. 211 fixed disk b. 201I fixed disk c. cartridge
INSTALL NOT EXECUTED FROM THE SYSTEM DISK	The INSTALL utility must reside on the system disk.
SPECIFIED OUTPUT DISK <disk-name> IS NOT AVAILABLE	The specified disk cannot be found. Possible cause is that the disk requested is not present on the system.
SPECIFIED OUTPUT DISK <disk-name> IS IN USE	The requested output disk is being used by another utility.
SPECIFIED OUTPUT DISK <disk-name> IS LOCKED	The requested output disk is in use by another utility.
SPECIFIED OUTPUT DISK <disk-name> IS A DUPLICATE	There are two disks with the same name present on the system.
DIRECTORY OF SPECIFIED DISK <disk-name> IS FULL – DISK SHOULD BE REFORMATTED FOR A LARGER NUMBER OF FILES	The directory of the output disk is full.
NO SPECIFICATION GIVEN	No output disk-name has been specified in the initiating message.

(continued)

Error Message	Description
ILLEGAL PARAMETER LIST <illegal parameter list>	The initiating message is not in the format INSTALL <disk-name>.
INVALID CHARACTER IN IDENTIFIER <invalid character(s)>	The initiating message disk-name contained invalid character(s)

In the preceding messages, "MINI" refers to a Burroughs Mini disk.

Install Copy Error Messages

These are error messages produced if a file cannot be copied.

If a file cannot be copied the following message is displayed:

<file-name> NOT COPIED - <reason>

where <reason> is one of the following:

<reason>	Description
READ ERROR ON DISK <system disk-name>	While reading <file-name> from the system disk, an error was reported.
WRITE ERROR ON DISK <output disk-name>	While writing <file-name> to the output, an error was encountered.
IN USE	<file-name> is being used by another utility.
NOT FOUND	<file-name> is no longer present in the system disk directory.
FILE IS A KEYFILE <output disk-name>	INSTALL does not copy key files.

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:

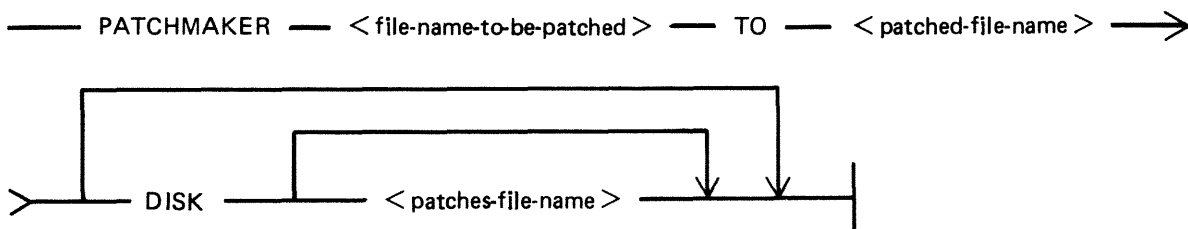
For B 90 MCP	MCPXn.nn.nn
For B 90 BILINTERP	BILXn.nn.nn
For B 90 COBOLINT	COBXn.nn.nn
For B 90 SORTINTRINSIC	SINTXn.nn.nn
For B 90 NDLINTERP	NDLXn.nn.nn
For B 90 MICROPOL	MICXn.nn.nn
For B 90 SAU	SAUn.nn.nn

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.

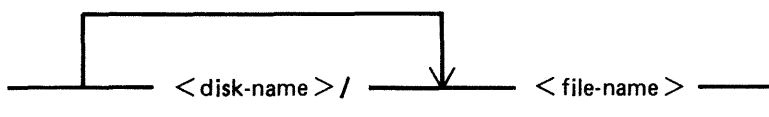
PATCHMAKER is executed via an initiating message which contains all the information concerning the mode of entry of the patches (displays and accepts or patch file) and the names and locations of the files involved. The initiating message for PATCHMAKER is as follows:

Railroad diagram:



Where <file-name-to-be-patched>, <patched-file-name> and <patches-file-name> are defined as:

Railroad diagram:



< 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.

NOTE

PATCHMAKER has no option to create a patch file. Under certain conditions the patch file will be created by the utility. These conditions are: 1. If a fatal error has occurred and the patches have been successfully entered via accepts. 2. If a non-fatal error has occurred and the patches have been successfully entered via accepts and the operator has elected to stop the utility at that point.

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.....

 :
 :
 :
 :
 :

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 2015228.

See also B 90 System Flash No. 3.04-00 and B 90 CMS 3.04 Release Documentation.

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.

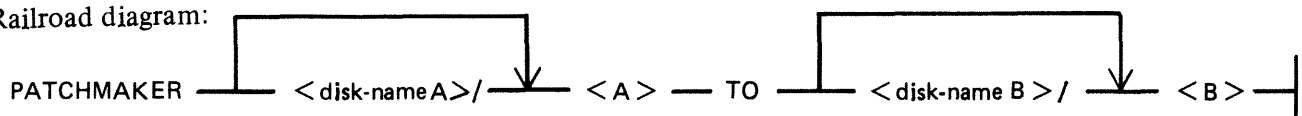
The instructions contained in the following procedure contain "labelled blanks" (A through E). When applying a patch using this procedure, the "labelled blanks" are replaced with the text of the corresponding "labelled inserts" of the Flash.

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

Railroad diagram:



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.

The following error messages are produced if an error is detected when entering patches via the SPO.

7. ERROR IN LAST LINE ENTERED – RESUBMIT
The line just entered, checksum has failed. Check the line for errors and re-enter.
8. ERROR IN PATCHES ENTERED – RUN ABORTED
The overall checksum for the patches entered has failed. Check the input to PATCHMAKER, then restart patch procedure.
9. 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.

10. 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.

11. 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.

12. <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.

13. 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, saving the operator inputting the patches again. The following message will be displayed to inform the operator that this has been done:

FILE PATCHES SAVED

14. 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.
15. PATCHING ERROR - PATCH TOO LARGE FOR <file-name>
Request technical assistance.
16. PATCHING ERROR - CANNOT FIND RECORD IN <file-name> FOR PATCH.
Request technical assistance.
17. PATCHING ERROR - TERMINATING CHECKSUM FOR <file-name>
INCORRECT
Request technical assistance.

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:

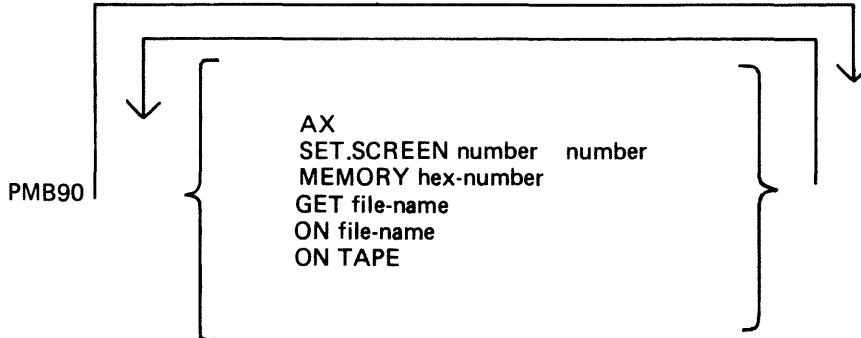
- PMB90 – object code file
- PMBHELP – data file of prompt messages
- PMBERROR – data file of error messages
- PMBM.xxxxx – data file for information on MCP xxxxx
- PMBO.xxxxx – data file for more information on MCP xxxxx

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:

Format diagram:



The meaning of these options is as follows:

AX

The program will use the SPO via DISPLAYS and ACCEPTS to communicate with the operator. If this option is not specified, the console will be used for communication.

SET.SCREEN

This option sets the screen page and line sizes from the numbers specified. These must be set if the DISPLAY option is used (see later). The input medium is the console keyboard but echoing of input is on the SELF-SCAN screen.

MEMORY

If this option is specified, the program will read only the first part of the memory dump, up to the byte given by the hexadecimal number. Note that use of this option could cause the program to fail

on certain print options if the analysis requires a part of the dumped memory that has been excluded.

GET

This option can be used to specify the name of the memory dump disk file. If not used, the file "MEMDUMP" on the system disk is opened. If this file is not present, a cassette labelled "MEM-DUMP/MEMORY" will be opened.

ON

This option can be used to specify the disk on which resides the memory dump file, if not the system disk.

ON TAPE

This option directs the utility to access memory dump files on cassette.

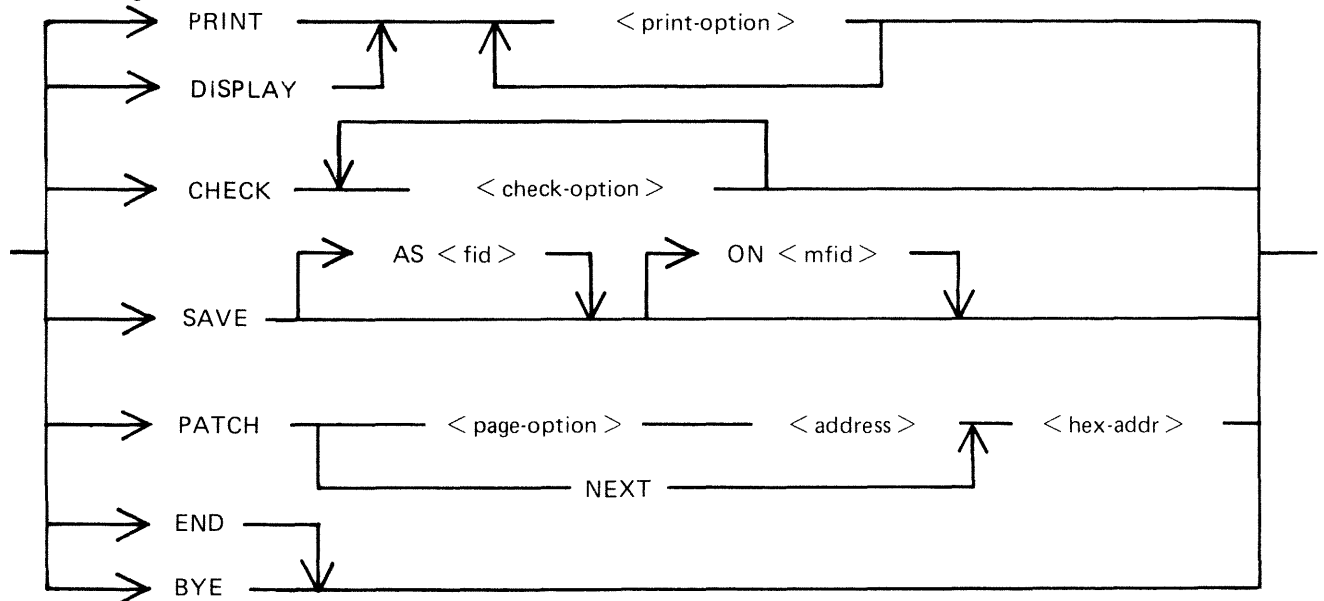
Using the Utility

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 the 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 later.

Railroad diagram:



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

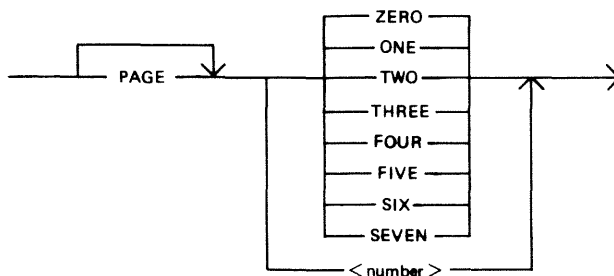
This option enables a copy of the dump (patched if required) to be made on the specified disk with the specified file-name.

PATCH

This option enables invalid areas of memory to be patched in the dump, to enable PMB90 to continue its analysis.

The <address> is four hex digits, and the <hex-addr> may be from 1 to 16 hex digits. The <page option> is as follows:

Railroad diagram:

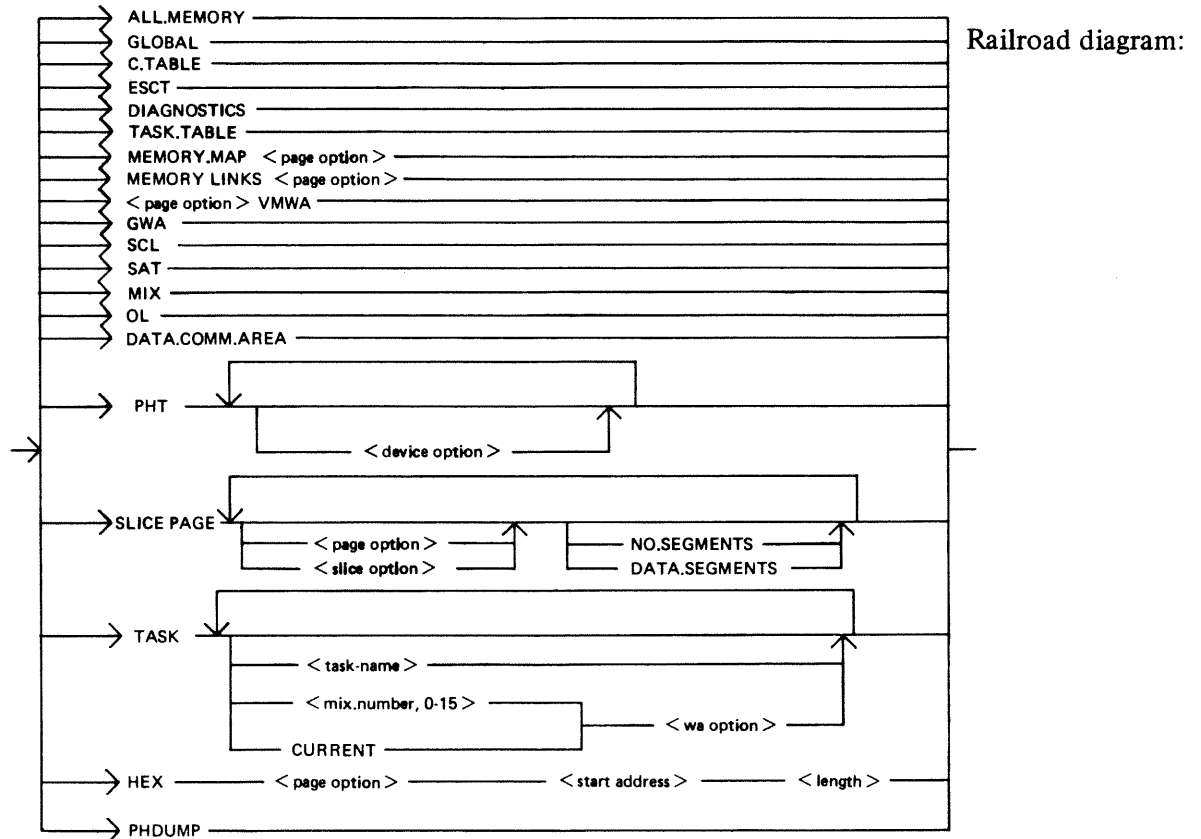


END, BYE

These alternative options cause PMB90 to go to normal EOJ.

<print-option>

The format of <print-option> is as follows:



ALL.MEMORY

This option creates an 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 areas. 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 provides 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
- SS – Self-Scan
- 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 0-45 or one of the following names:

DISKDDR
LPDDR
PANDDR
KBDDR
CASSDDR
SENDDR
CONSOLE
INXS
SCREENSN
SUSN
INITIALIZE
ADCDDR
SDCDDR
OPENCLOSE
DCCH
SPO
CONBUFSN
SCLBUFSN
ICMDDDR
OCOMSN
DIAGSN
REMSPOSN

TASK

This option prints the contents of a Task Control Block (TCB).

The <task-name> may be one of:

NDL
MCS
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.

Example:

To obtain complete memory dump print on the console printer:

```
PMB90 (OCK)
PRINT ALL.MEMORY (OCK)
```

To obtain a dump of the data comm buffers, plus the data comm controller device-dependent routines, plus the MCS and NDL task tables:

```
PMB90 (OCK)
PRINT DATA.COMM.AREA (OCK)
PRINT PHT ADC SDC (OCK)
PRINT TASK MCS NDL (OCK)
```

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.

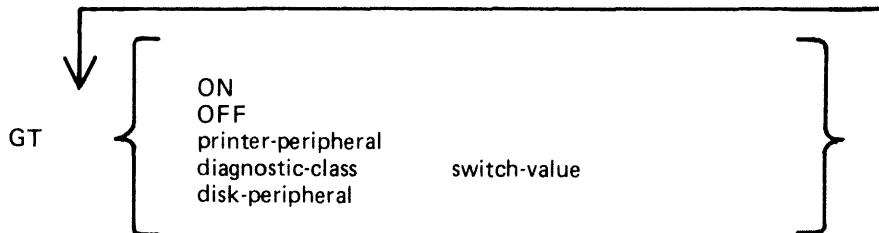
B 90 DEPENDENT INTRINSICS

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 diagram:



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).

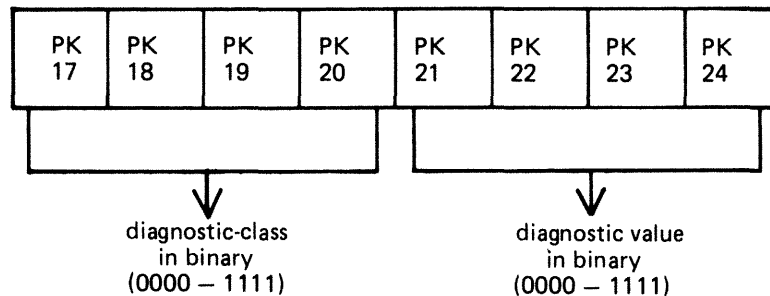
The GT command allows the storage of a switch-value for each diagnostic class. If the trace has been initiated via the GT ON command, then each time a trace point is encountered in the machine code, a diagnostic printout will occur if the switch-value for that class of trace is lower than or equal to 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 is displayed.

Examples:

To find the retry count for DMA and DKA:

```
GT DMA
DMA      0      RETRIES

GT DKA
GT DKA NOT ON SYSTEM
```

To set the switch-values to CC for interpreters, then interrogate all switch-values, then turn on printing to LPA:

```

GT      B      CC
GT      .
      DIAG SWITCHES FF FF FF FF FF FF FF FF FF FF
.....FF FF CC FF FF FF FF
GT LPA
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 1111 xxxxxxxx xxxxxxxx xxxx xxxx xxxx DF xx

```

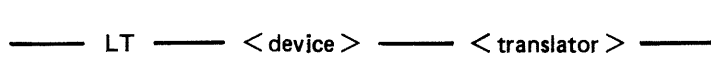
where x indicates any hexadecimal digit.

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 be 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.

LT (Load Translation Table)

This intrinsic will cause the specified translation table, in SYSTRANS, to be used and will store appropriate information in the file SYSCONFIG so that it is remembered across system power off and warm-start. This will only occur if the device has already had a translation table assigned when CONFIGURER was executed.

Railroad diagram:



where <device> may be KB, SP, SS and LP

and <translator> may be the name of the translation table.

Example:

To load and assign the translation table LP.B924930.BRZIL.V6 to LPA, the following syntax could be used:

```
LT LPA LP.B924930.BRZIL.V6
```

LT Error Messages

Error Message	Description
<invalid input> INVALID	The device is in use, or is invalid; also the syntax of LT may be invalid. Check input and ensure the device is online and ready.
<device> NOT ON SYSTEM	The specified device is not present. Ensure that the required device is present and ready before using LT.
LT FAIL – TRANSLATION FILE NOT FOUND	The system file SYSTRANS, which contains the translation tables, is not present on the system disk. Copy SYSTRANS to the system disk.
LT FAIL – TRANSLATION TABLE NOT FOUND	The required translation table is not present in SYSTRANS.
LT FAIL – TRANSLATION TABLE INCOMPATIBLE	The file SYSTRANS on the system disk is incompatible with this system. Copy the correct version of SYSTRANS onto the system disk and try loading the translation table again.
LT FAIL – NO SYSCONFIG ENTRY FOR THIS DEVICE	This device did not have a translation table assigned to it in the file SYSCONFIG. Execute CONFIGURER and assign the table to the device.
LT FAIL – I/O ERROR ON TRANSLATION FILE	A hardware error has occurred while accessing the file SYSTRANS on the system disk. Copy a good version of it to the system disk and execute LT.

ND (New Density)

This intrinsic allows the operator to define the print density on suitable console printers.

Format:

ND peripheral density

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.

Examples:

ND SPA 1
ND SPA B

When the number of characters per inch changes, the operating system will adjust the values of page width and offset (previously set by an FD command or by default) so that any subsequent output is restricted to the part of the platten available with the previous density. The 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 the system configuration (SYSCONFIG) file and are remembered across system shutdown and warmstart.

At warmstart, if any inconsistency is found between the system configuration file information and the capability of the current printer, then the system configuration file information is ignored and default values are 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	

Output Messages

Message	Possible Cause	Suggested Action
input INVALID	(1) The specified density value exceeds the number of options available on the configured console. (2) The printer is in use. (3) The mnemonic is incorrect.	(1) Enter correct input. (2) Wait until program has closed the console file. (3) Correct the input and re-enter.
input NOT ON SYSTEM	The specified peripheral does not exist.	Correct the input and re-enter.

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 users at two distinct levels of sophistication.

1. System 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 Intrinsic. These utilities and intrinsic are either unique to this system or have some feature that is unique to these systems. For a discussion of other utilities and intrinsic, 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) and to take ROM-generated dumps (explained later in this section).

At the upper right of the control panel is a series of switches. There are more switches for the B 900/CP 9570 than there are for the CP 9550.

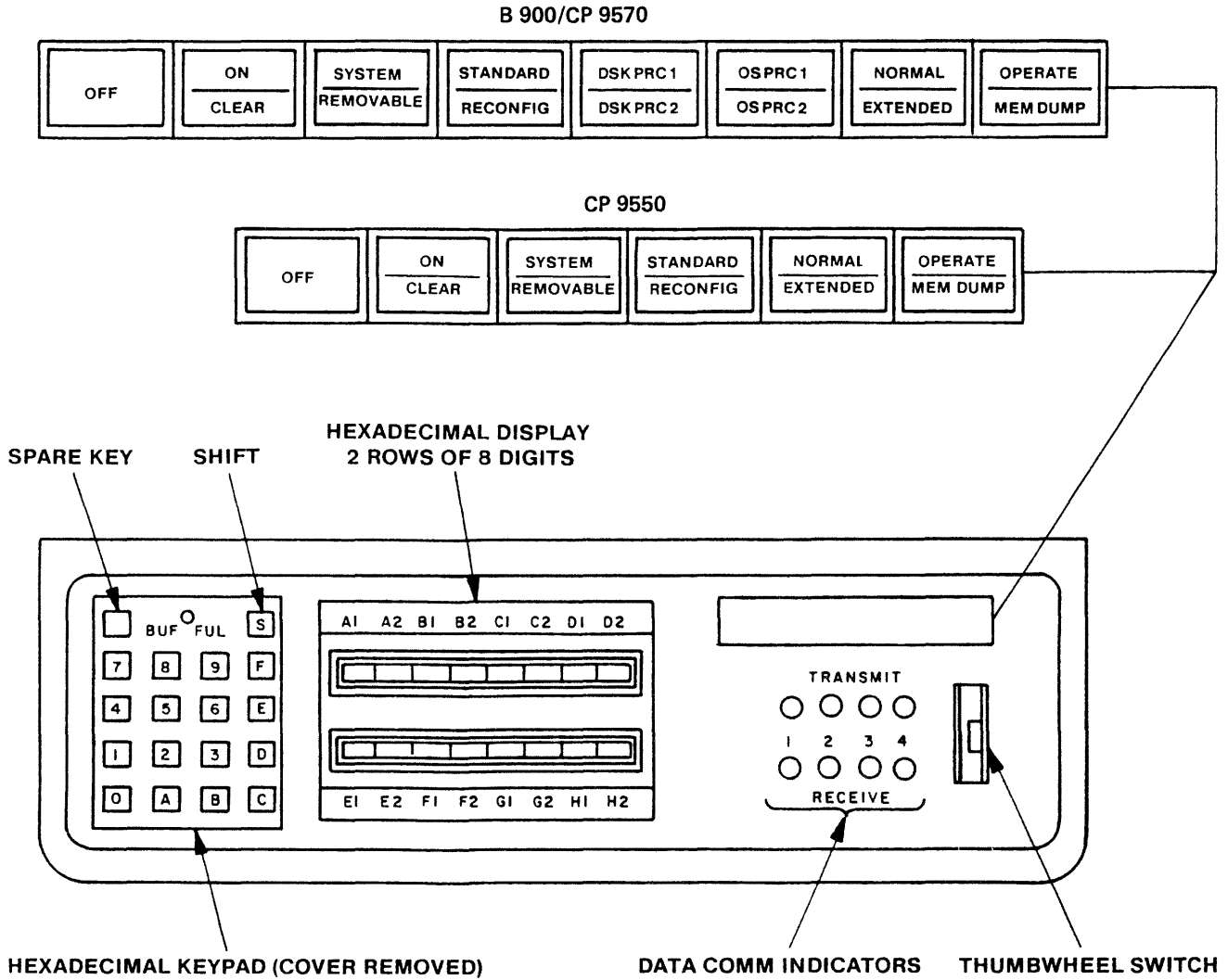
At the left is a set of lighted displays that flashes on and off when the system is operating. The numbers shown are significant. Most indicate normal operation or are for maintenance purposes. The Functional Description subsection discusses these numbers in detail. The time is displayed in the bottom 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.



ED2143c

Figure 9-1. Control Panel

OPERATOR ATTENDED/UNATTENDED

Systems are described as either:

- Operator attended
- or
- Operator unattended

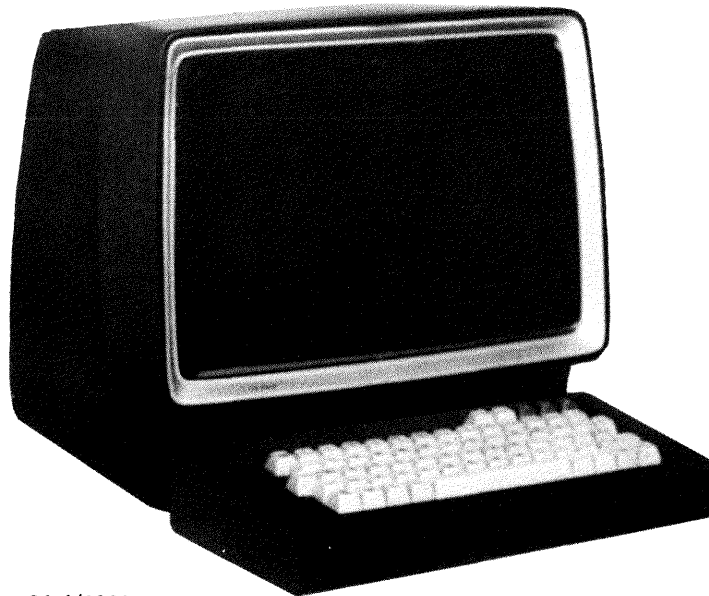
In operator attended mode, an Operator Display Terminal (ODT) is part of the system. This ODT looks very much like a typewriter attached to a television screen. An operator types information on the keyboard to communicate with the processor. The numbers and words typed on the keyboard appear on the screen. 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.

An ODT is shown in figure 9-2.

In operator unattended mode, there is no ODT. 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.

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.

The rest of this section focuses on the operation of systems where an ODT is present.



6A-1/1001

Figure 9-2. Operator Display Terminal (ODT)

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 powering on and off procedure, refer to the B 900/CP 9500 Operators Manual, form 1118270.

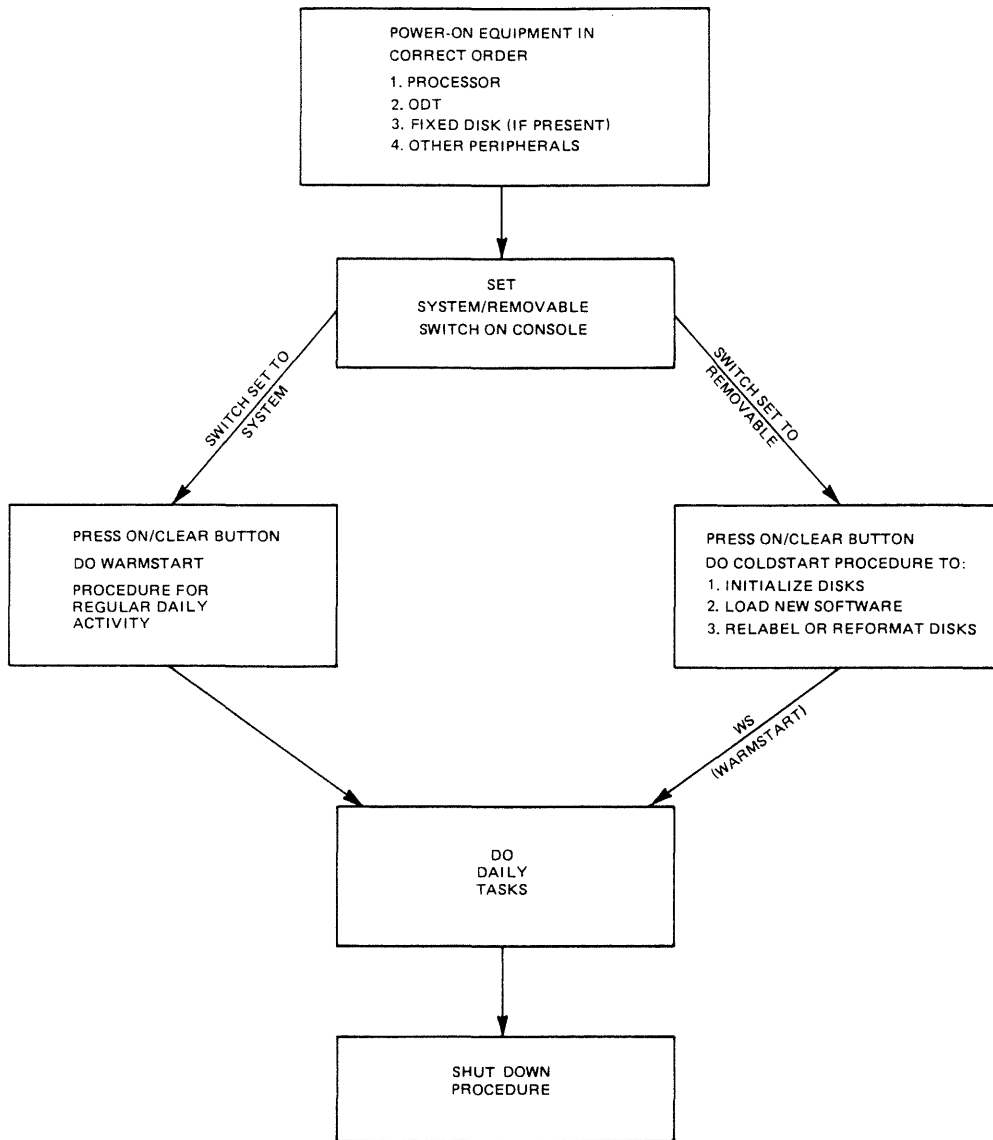


Figure 9-3. Block Diagram of Operating Steps in Daily Running of B 900/CP 9500

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.

First, turn on the processor.

If your system has an ODT, it should be turned on next, followed by the fixed disk unit if your system has one.

After these have been turned on, any removable disk units (including PACK and the PACK controller) and printers can be turned on. Again, the order is:

1. Processor
2. ODT
3. Fixed Disk Unit (when present)
4. Other Peripherals

Equipment is turned off in 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 ODT

In a system that has an ODT, the warmstart procedure is:

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 ODT, 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 03.04.01
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 11/14/82 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 82 82315* 1020
TRANSFER COMPLETED
COMMENCING LOG FILE REALLOCATION
LOGGING IS INITIATED ON 11/14/82 at 10:22:32 (MCP VERSION 03.04.07)
```

* Julian date

When the MCP version message appears, the system is warmstarted.

Warmstart Variations

1. If the system has no ODT (operator unattended): It is not possible to enter a date. 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 unit. 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 over.

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 in the Functional Description subsection which follows.

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 or reformat a disk.

There are other uses for coldstart, including:

1. Initializing maintenance test routine (MTR) disks (the FE command).
2. Adding 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 rather than the system operator.

There are two types of coldstart operation:

1. Operator attended. (ODT is present.)
2. Operator unattended. (There is no ODT, 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 appropriate device (write disabled).
2. Set the SYSTEM/REMOVABLE switch to REMOVABLE and put the other switches in their primary positions (STANDARD, NORMAL, OPERATE, DISK PROCRI, and OSPROC1).
3. Press ON/CLEAR button on processor. Wait a few minutes.

The system displays a message on the ODT screen giving the number of the MCP operating the system and requesting the date.

Example:

```
*BOJ COLDSTART -REV.03.04.20  
ENTER DATE <MM/DD/YY>
```

(*BOJ stands for beginning-of-job.)

4. Enter the date in the style requested. Leading zeros must be included.

Example:

01/04/83

The system responds by giving the status of each disk drive.

Example:

PAYROL READY ON DKA

SYSTEM READY ON DKB

The format is:

```
                READY ON
<disk name>    NOT READY ON      <drive mnemonic> <driveid>
                WRITE INHIBITED ON
```

If some disk units are Not-Ready, the following message also appears:

TYPE "A" WHEN DISKS ARE READY

5. If necessary, ready the needed disk drive. To ready the disk drives needed:

If cartridge, press RUN button after disk is inserted.

If mini, insert write-enabled disk.

If fixed disk, power-on.

If disk pack, press READY button.

Then type in A.

NOTE

If the operator wishes a drive to be NOT READY, it can be left in this state.

Type A and transmit, and the coldstart procedure continues.

The system now gives a message similar to this:

STORING SYSINITBOOT IN MEMORY

SOURCE DISK: B900RL1

B900RL1 IS ON <device-id>

ENTER HELP FOR LIST OF COMMANDS

ENTER FUNCTION

The operator is now ready to enter one of the functions from table 9-1.

Table 9-1. Coldstart Functions

?DS	Discontinue: Terminates the following functions: FE, IN, LD, PT, RF, RP.
FE	Field Engineering: Test function used by Burroughs engineers.
HE	Help: Displays list of Coldstart functions and their mnemonics.
IN	Initialize: Initializes a disk.
LD	Load: Loads files from release disk to system disk.
OL	On-Line: Lists status of drives on system.
PT	Patch: Patches system files by replacing system files on the system disk with their counterparts from a patch disk. PATCH function performs a check to ensure file release level consistency for the files.
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.

?DS
COLDSTART

?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 ODT.

FE
COLDSTART

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.)

HE
COLDSTART

HE (Help)

If the operator has forgotten the format of the command to be entered, he/she can enter HE when the ENTER FUNCTION message appears.

If HE is entered, the following list appears on the ODT:

FE = INITIALIZE MTR DISK
HE = HELP
IN = INITIALIZE A DISK
LD = LOAD FILES FROM RELEASE DISK
OL = LIST STATUS OF DRIVES
PT = PATCH SYSTEM FILES
RF = REFORMAT DISK TO INITIAL STATE
RL = RELABEL DISK
RP = REPLACE SYSTEM FILES
WS = WARMSTART

IN COLDSTART

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 or a 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, its size, and its 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 from a disk.

Procedure

Follow steps 1 through 5 under Coldstart Initiation, then proceed with step 6 as follows.

6. When the message ENTER FUNCTION appears, put the disk to be initialized (often called a scratch disk) in an empty drive. Enter:

IN

at the ODT and transmit.

The system responds with:

TYPE OF DISK <REM OR FIX>

7. Type in REM (removable) and transmit. (The use of FIX in this response is explained later.)

CAUTION

Before any of these procedures is done, all existing files the operator wishes to save should be backed up.

The system responds with the message:

DRIVE <3 CHARACTERS>

8. Enter the three-character drive name of the drive where you placed the disk which is to be initialized.

Example:

DKA

Several minutes pass, and the normal system response is:

NO. OF BAD SECTORS = ##

DEFAULT MODE <Y OR N>

9. It is possible that standard (default) answers are already established and that Y should be entered. If Y is appropriate, enter it. (See page 9-43 for default values.) The system then automatically completes the initialization.

It is more likely that N is the appropriate answer. If N is entered, the system responds with:
ENTER FILES <1-2804>

10. Enter the number of files that is to be the maximum permitted on this disk. The numbers 1 to 2,804 give the range that is possible. 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.

The system responds with:

ENTER OBJECT PACKID

11. Enter the name of the disk being initialized. The name is selected by the operator and can be up to seven letters or numbers long, beginning with a letter.

The system responds with:

ENTER SERIAL <6 DIGITS>

12. Enter a six-digit serial number. Many companies feel no need to have serial numbers on their disks. If this is the case, pick any six-digit number at random and enter it, as there must be some entry made to this request.

When the serial number has been entered, the system responds with:

ENTER OWNER <1-14 CHARACTERS>

13. It is customary either to enter a company name, a department name, or the name of an individual.

When this information has been received, the system proceeds to complete initialization. For a cartridge disk, this total procedure takes about four minutes.

When initialization is complete, the message:

END INITIALIZE

ENTER FUNCTION

appears.

To initialize additional disks, 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 identical except that FIX is entered instead of REM.

LD COLDSTART

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. Use the new B900RL1 disk to do the coldstart.

The initialization of the disks is finished when the message ENTER FUNCTION appears on the screen.

1. When the message ENTER FUNCTION appears, enter

LD
and transmit.

The system responds to LD with a message:

TYPE OF DISK <REM OR FIX>

2. Enter the disk type of the destination disk.

If REM is entered, the system responds with the message:

ENTER OBJECT PACKID <1-7 CHARACTERS>

and the operator responds by entering the disk name. If FIX is entered, the OBJECT PACKID message is omitted.

The system then begins loading the system files. Each file is listed as it is loaded.

It is possible that the information you are loading will be on more than one disk, particularly if using mini disks. If so, they are labeled in sequence: B900RL1, B900RL2, and B900RL3.

When the information on a disk has been loaded, the system issues the following message:

SOURCE DISK: B900RLn

PACK NOT ON LINE

ENTER RELEASE DISK: B900RLn

(where n is the number of the release disk being asked for, the next one to be loaded)

3. If there is more than one release disk, remove the first one, place the next one in the drive, type A, and transmit.

If there is only one disk or if you have entered the last disk in a series, type B and transmit.

After B has been entered, the system responds with one of two messages:

SYSTEM LOAD INCOMPLETE

If the load was unsuccessful and the initialize and load must be started over;

or

END LOAD

ENTER FUNCTION

if the load was successful.

4. When ENTER FUNCTION appears, enter:

WS

to warmstart or, if another coldstart function is desired, enter its two-character ID.

OL
COLDSTART

OL (ON LINE)

This command lists the disks that are on-line. It works the same as the OL command does during regular system operation.

Example:

OL

is entered. The system responds with:

DMB ACCOUNT

DFC THISDSK

DKG PACK123

(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)

PT

COLDSTART

PT (PATCH SYSTEM FILES)

From time to time Burroughs makes changes to system software, and the changes are smaller than releases (which completely replace the system files). These improvements are called patches. They are received in one of two ways:

1. On some removable media such as a one-megabyte mini disk, or
2. In printed form on a Flash (a formal announcement from the corporation).

Patches received in printed form must be placed on the appropriate file on the B900RL1 disk using the FPP command (explained in the Utilities and Intrinsic subsection).

The newly patched file then can be placed on the system disk (replacing the old file of the same name) using the PT command. (RP also can be used.)

Patches received on a removable media can be put directly on the system using the PT command.

The following system files can be patched:

SYSCOBOL
SYSMCP
SYSICP
SYSDSCP
SYSHOSTLOAD
SYSWARMSTART
SYSMPLII
SYS-SUPERUTL
SYSCONFIG
SYSTRANSLATE
SYSBOOTSTRAP

To use this command, insert the disk containing the updated system files in the loader channel (the channel where B900RL1 disks are placed).

Follow the basic coldstart procedure.

When the system responds with:

TYPE OF DISK <REM OR FIX>

the operator selects FIX when the system disk is a fixed disk, or REM if the patch is being made to a disk pack or other removable media.

The system responds with:

ENTER OBJECT PACKID <1-7 CHARACTERS>

The operator enters the name of the disk receiving the updated software and transmits. Patching is complete when the system sends the message:

END PATCH
ENTER FUNCTION

Patching Restrictions

Each system file and each updated file have a version string associated with them. The format of this string is:

RRRRPP

where the RRRR represents the release level and PP represents the patch level. The version string for each file on the patch disk is checked against its corresponding system file version string before any loading of updated files occurs. One of the following actions results:

1. The release level field of the version string in the updated file and system file must be equal. If this is not true, the patch function terminates immediately and the operator is notified:
VERSION MISMATCH
2. If the updated file has a patch level which is less than the patch level of the corresponding system file, patch will terminate and the operator is notified:
PATCH LEVEL ERROR
3. If the patch level of an updated file is equal to the patch level of the corresponding system file, no error results but the updated file is ignored and is not loaded.
4. Any updated file having a patch level greater than the patch level of the corresponding system file is loaded.

Once all updated file version strings have been checked against all system file version strings, Patch copies all updated files to the disk being patched. If any error occurs, the Patch function terminates and whatever updated files have been copied are deleted. This returns the disk being patched to its original state. If all patch files are copied without error, the corresponding old system files are deleted. This completes the Patch function.

RF COLDSTART

RF (REFORMAT)

The Reformat (RF) coldstart command clears a disk (fixed or removable).* This disk must have been initialized previously.

RF is faster than IN.

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 number which was reported the last time the disk was initialized. Consequently, it is recommended that the CHECK.DISK command be run against any disks which are reformatted.

CHECK.DISK will list all bad sectors. These bad sectors then can 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

To use Reformat, follow the standard coldstart procedures. When the system message is:
TYPE OF DISK <REM OR FIX>

1. If REM is entered, the following message is issued:

ENTER OBJECT PACK ID <1-7 CHARS>

The operator enters the name of the disk. Once reformat identifies the disk, its label is accessed and used to print the following information:

NO. OF BAD SECTORS <#>
DATE <date of last Initialize or Reformat>
FILES <maximum number of files presently allowed>
SERIAL <serial number of disk>
PACKID <name of disk>
OWNER <owner of disk>

2. If Fix is to be entered, the fixed disk unit must be turned on and the following messages issued:

NO. OF BAD SECTORS <#> **
PACKID <name of disk> **
FILES <maximum number of files presently allowed>

*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.

** These messages are repeated for each spindle in the assemblage.

The operator is now offered the option of using default mode by the message:

DEFAULT MODE <Y OR N>

If Y is selected (default mode), all further information needed by reformat is derived from the default values. No further information is requested from the operator. Refer to the Default Coldstart Assignment subsection for a list of default options.

If N is selected, Reformat will generate the disk directory and label using the information supplied to the following queries:

ENTER FILES <1-2804>

Enter the number of files to be allowed on the disk.

NOTE

When reformatting fixed disks, the statement ENTER FILES <1-2804> is displayed once for each fixed disk assemblage.

ENTER OBJECT PACKID <1-7 CHARS>

Enter the disk name which can be from one to seven alphanumeric characters.

ENTER SERIAL <6 DIGITS>*

Enter six numerical characters as the disk serial number.

ENTER OWNER <1-14 CHARS>*

Enter the name of the owner of the disk which can be from 1 to 14 alphanumeric characters.

When Reformat terminates, it issues the message:

END REFORMAT

ENTER FUNCTION

*This message is displayed only when reformatting a removable disk.

RL COLDSTART

RL (RELABEL)

This function is used to relabel (retitle) removable disks, fixed disks, and pseudo disks.

To use RD, follow the basic coldstart procedures and enter
RL
when the system message
ENTER FUNCTION
appears.

Relabel then requests the type of disk to be relabeled by issuing the following message:
TYPE OF DISK <REM OR FIX>

Next, Relabel requests the name of the disk to be relabeled by issuing the following message:
ENTER ID OF PACK BEING RELABELED

Relabel will look for the disk with the name specified by the operator. When it has located the disk, the following message is displayed:

<packid> IS ON <disk mnemonic>
Relabel then requests the new name by issuing:
ENTER NEW PACKID <1-7 CHARS>

Relabel updates the specified disk with the new name and then displays the message:
END RELABEL DISK

RP COLDSTART

RP (REPLACE)

The Replace command (RP) can be used to load new software into the system.

It has an advantage over LD in that an RP can be done without removing other files from the system. Its disadvantage is that version string checks are not done.

The following system files can be replaced:

SYSBOOTSTRAP
SYSDSCP
SYSMCP
SYSWARMSTART
SYSHOSTLOAD
SYSICP
SYSMPLII
SYSTRANSLATE
SYS-SUPERUTL
SYSCOBOL
SYSCONFIG

To use RP, follow the basic coldstart procedure and enter RP when the
ENTER FUNCTION

message appears on the ODT. The new B900RL1 disk must be used to coldstart. The system then requests information on the type of disk to be updated:
TYPE OF DISK <REM OR FIX>

If the disk type specified is "removable", Replace requests the name of the disk to be updated:
ENTER OBJECT PACKID <1-7 CHARS>

The Replace function uses the system files (just listed) from the release disk to replace the copies of these files on the specified disk. If any 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 user is notified when the Replace function finishes successfully via the message:
END REPLACE

WS COLDSTART

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

Warmstart is finished when the MCP version appears on the ODT.

Coldstart Operator Unattended Mode

The system automatically enters coldstart unattended mode when there is no ODT on the system. 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 only because all three functions will be executed automatically. The individual functions are the same as those implemented in coldstart attended mode. There are three series of functions which the operator may select:

1. INITIALIZE, LOAD, and WARMSTART
2. PATCH and WARMSTART
3. REPLACE and WARMSTART

The system decides the set of functions to perform using the switch settings on the control panel. If the system detects the SYSTEM/REMOVABLE switch in the "SYSTEM" position, WARMSTART is initiated. If "REMOVABLE" IS DETECTED, COLDSTART will then examine the STANDARD/RECONFIG and NORMAL/EXTENDED switches for positioning, and execute the appropriate set of functions. When the functions are completed, COLDSTART signals the user to set the control panel switches in their primary positions (that is SYSTEM, STANDARD, DSKPRC1, OSPRC1, NORMAL, OPERATE) for WARMSTART to proceed. See figure 9-4 for a description of switch settings and their corresponding functions.

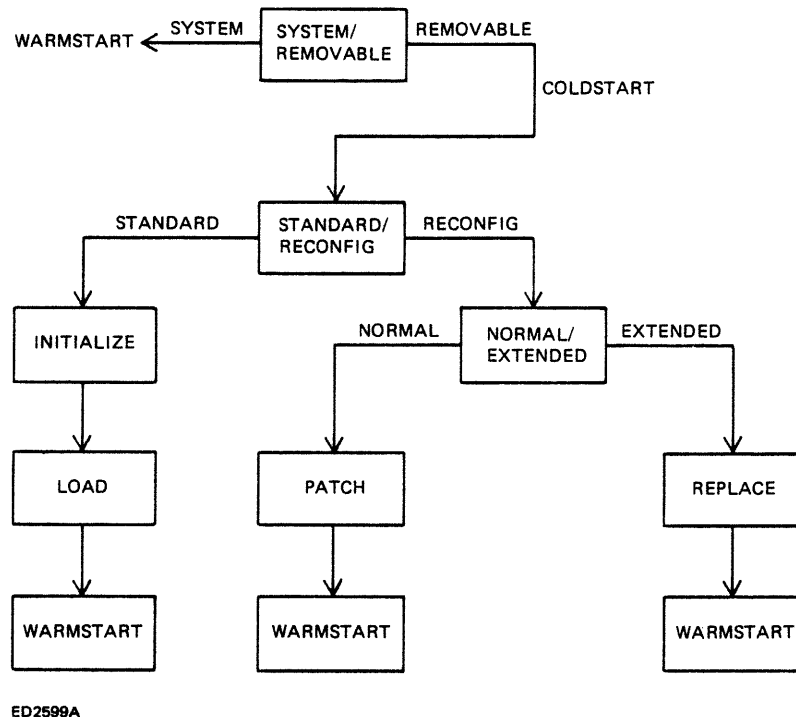


Figure 9-4. Switch Settings for Unattended Operation

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 STANDARD/RECONFIG switch to STANDARD.
3. Press ON/CLEAR.

The INITIALIZE/LOAD functions (usually fixed disk) act upon the fastest disk device available to the system. 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 completed, 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 Initialize function, @20@ during Load function, and @50@ during 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, thus indicating 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.

Patch, Warmstart

In unattended mode the PATCH/WARMSTART series of functions is initiated when the user puts a B900RL1 release disk in the loader channel and sets/presses the following switches:

1. Set the SYSTEM/REMOVABLE switch to REMOVABLE.
2. Set the STANDARD/RECONFIG switch to RECONFIG.
3. Set the NORMAL/EXTENDED switch to NORMAL.
4. Press the ON/CLEAR switch.

Coldstart executes the PATCH/WARMSTART function on the fastest disk device available to the system. As in operator attended mode, the version string for each patch file is checked against the version string of the corresponding file before any patching occurs. If an error occurs, the patch function will terminate, and coldstart will display an appropriate message on the hex display. Files which may have

been loaded prior to the error are automatically removed, thus returning the disk to its original state. When the patch function completes, Coldstart will flash the hex display on and off to indicate Warmstart is ready to take control. The operator should:

1. Remove the B900RL1 release disk.
2. Set the STANDARD/RECONFIG switch to STANDARD.
3. Set the SYSTEM/REMOVABLE switch to SYSTEM.

After step three is completed, warmstart will automatically be initiated.

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 STANDARD/RECONFIG switch to RECONFIG.
3. Set the NORMAL/EXTENDED switch to EXTENDED.
4. Press the ON/CLEAR switch.

As in operator attended mode, the Replace function is used to replace certain system files on a previously created disk. Unlike the patch function, 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 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 STANDARD/RECONFIG switch to STANDARD.
4. Set SYSTEM/REMOVABLE switch to SYSTEM.

After step four is completed, Warmstart is automatically initiated.

POWER OFF

The system must be powered off in two steps:

Logical Power-off
and
Physical Power-off.

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 & B banks = Memory Address
 D1 = Bus Address
 D2 = Page Address
 E & 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", then 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 Option

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, then the "F" key. This step initiates the memory dump process.
3. The ROM dump is complete when all hexadecimal displays are blank. (Continue to Step 12.)
4. If the ROM dump terminates with a @12@ in bank "D", then 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, and then press the "B" key.
7. Remove the full disk; replace it with a new one.
8. Press and hold the "S" key, and then press the "D" key.
9. Enter the values as noted in step five above. (Leading zeros must be included.)
10. Press and hold the "S" key, and 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, DUMPFIL.UTL, pertains to the dumping procedure. This command can be used to partition a dump file into a number of smaller files or to reconstruct a file that has been previously partitioned.

The use of these utilities is explained in detail in the Utilities and Intrinsic subsection later in this book.

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 then can be analyzed using the ANALYZER command explained in the Utilities and Intrinsic portion of 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 standalone 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 standalone utility whose primary function is to determine the system hardware and software configuraton 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 and 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 labeled B900RL1.

The operator therefore is responsible for selecting whether the system will Coldstart or Warmstart when it is put in startup mode by appropriately setting the SYSTEM/REMOVABLE switch.

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 9-2.

Table 9-2. 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 9550-2 systems with disk pack, a disk other than disk pack must be on-line and have a valid 3.4 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 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 in 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 standalone utility Coldstart, the disk must have a packid of B900RL1 and contain the system files SYSCOLDSTART and SYSDSCP. If the B9387 DPDC is on-line, then the B900RL1 disk must also contain the file SYSHOSTLOAD.

To load the standalone utility Warmstart, the required system files are SYSWARMSTART and SYSDSCP. If 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, then 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 9-2 shows the format of the HEX display during TRACK ZERO BOOTSTRAP and SYSBOOTSTRAP execution. The primary status of TRACKZERO BOOTSTRAP and SYSBOOTSTRAP execution can be determined by the value appearing in bank "A". Banks "B" through "H" contain additional information on TRACK ZERO BOOTSTRAP and SYSBOOTSTRAP status. For detail on the identifiers found in banks "B" through "H" of table 9-2, refer to the information which follows.

TRACK 0 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 thru 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 thru H Bank = Disk IOC status

A Bank Display	Description
@23@	Invalid Disk Drive ID: 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 thru 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 thru 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 thru 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 thru H Bank = Disk IOC status
@27@	Key Search Failure: 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 thru 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 thru H Bank = Disk IOC status
@2E@	No SYSBOOTSTRAP can be found. Note: A 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.

SYSBOOTSTRAP ERROR DISPLAYS

A Bank Display	Description
@30@	<p>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.</p>
@31@	<p>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.</p>
@32@	<p>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</p>
@33@	<p>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 D Bank = Drive ID read E Bank = 00 – Load in progress 01 – Load complete else – Load failure; reason unknown</p>
@34@	<p>SYSHOSTLOAD Loading: If C & E Thru H 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 & E Thru H Not Blank B Bank = Most Digit = 0, Least Digit = Channel C Bank = ERROR</p> <p style="margin-left: 40px;"> 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.</p>

A Bank Display	Description
@34@ Cont.	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 & D Banks = M1 Register E & F Banks = MAX Register G & H Banks = M2 Register If Micro Access C & D Banks = UMAR Register E & F Banks = UMARX Register
@3F@	SYSBOOTSTRAP Terminating: E Bank = Blank – normal termination 00 – normal termination 01 – Cannot find OS processor 02 – Cannot find a SYSCOLDSTART/SYSWARM-START disk 03 – Not able to set proc. 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 the section BANK A HEX DISPLAYS FOR STARTUP PROGRAMS under diskid, table 9-2.)

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 is 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 Warmstart use default assignments, then Warmstart will use the default configuration. (See default configuration below.)
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 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	80K	40K
4	80K	40K
3	64K	32K
2	32K	16K

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 128KB of memory will be assigned both interpreters. Task processors with 128KB 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 in the Utilities and Intrinsic section of this document.

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 in the Utilities and Intrinsic section of this document.)

Warmstart has two ways of reporting its status to the operator. When an Operator Display Terminal (ODT) is present on the system, Warmstart attempts to direct messages to the ODT device, and in both operator attended and unattended operation, Warmstart uses the hexadecimal display to indicate the status of its execution. During Warmstart, Bank A is a primary indicator of Warmstart's progress through its code, while banks B through H provide additional information on the status of Warmstart.

Table 9-3 outlines the contents and meanings of values appearing in the display banks during Warmstart.

Table 9-3. 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 ODT
@EO@	WARMSTART COMPLETE - PASS CONTROL TO OS

Table 9-3. General Description of Warmstart Messages (Cont.)

A Bank Display	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 usccessfully: B Bank - Number of logical operations performed. C Bank - Operation in process E - F Bank - Current offset G - H Bank - Next offset	
@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 = ODT, 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 = ODT, 02 = Processor, 03 = Memory F Bank - If disk: Most-digit = logical drive number or If Proc or memory: Most-digit = Bus, Least-digit = Page G,H Bank - Device Dependent Parameters	

Table 9-3. General Description of Warmstart Messages (Cont.)

A Bank Display	Description
@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 = ODT, 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 = ODT, 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 D Bank - Operation type: 00 = Command, 01 = Read, 02 = Write, 03 = Search E Bank - Device Type: 00 = Disk, 01 = ODT, 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 = ODT, 02 = Processor, 03 = Memory F Bank - If disk: Most-digit = logical drive number or If Proc 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

Table 9-3. General Description of Warmstart Messages (Cont.)

A Bank Display	Description
@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
@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 = ODT, 02 = Processor, 03 = Memory F Bank - If disk: Most-digit = logical drive number or If Proc or memory: Most-digit = Bus, least-digit = Page G Bank - General Status (See description in @C9@.) H Bank - Device Error Status (See description in @C9@.)

Table 9-3. General Description of Warmstart Messages (Cont.)

A Bank Display	Description
@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 = ODT, 02 = Processor, 03 = Memory F Bank - If disk: Most-digit = logical drive number or If Proc 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 standalone 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 an Operator Display Terminal (ODT), thus allowing Coldstart to display information to, and request information from the operator via the ODT device. If a system does not have an ODT device, then Coldstart will operate in an Operator Unattended Mode, thus causing Coldstart to become user-transparent. No information is requested from or displayed to the operator except via the hexadecimal display.

B 900/CP 9500 system software is distributed on release disk(s) labeled B900RL1, B900RL2, and so forth. 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 maintain 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 apply these patches to the system disk. The Field Patch Program is described in detail in this section under Utilities and Intrinsic.

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 flag 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 by the FE function is for identification of the MTR disk's drive. The following message appears:

ENTER DRIVE <3 characters>

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, mnemonic ("DFX") cannot be used as the MTR disk. Only the mnemonics "DKX" and "DMX", cartridge and mini respectively, are valid.

Bad sectors within the six tracks reserved for usage by MTR will cause failure of the FE initialize and the message "BAD MTR TRACK" to be displayed. Other tracks, which later are to contain MTR software, are allowed up to 50 bad sectors, total. The number is reported to the user:

NO OF BAD SECTORS = ##

The user is now offered the option of using "DEFAULT MODE". The following message is issued:

DEFAULT MODE <Y OR N>

If "Y" is selected (that is, default mode), all further information needed by FE initialize is derived from the default values. No further information is requested from the operator. Refer to the section "COLDSTART DEFAULT ASSIGNMENT" later in this document.

If "N" is selected, FE initialize will generate the disk directory and label using the information supplied to the following queries:

ENTER FILES <1-2804>

Enter the number of files to be allowed on the disk. After the number of files is entered, FE initialize will request the packid for the disk about to be initialized.

ENTER OBJECT PACKID <1-7 CHARS>

Enter the diskid which can be from one to seven alphanumeric characters. After the diskid is entered, FE initialize will request a serial number via the message:

ENTER SERIAL <6 DIGITS>

Enter from one to six numeric characters as the disk serial number. After the serial number is entered, FE initialize will request the name of the owner of the disk which can be from one to fourteen alphanumeric characters via the message:

ENTER OWNER <1-14 CHARS>

When FE initialize ends successfully, the following message is displayed.

END INITIALIZE MTR DISK

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 in this document 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 which is 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.

Coldstart Disk Selection

All disks have been assigned priority according to their speed. The fastest disk device is the fixed disk, then disk pack, then 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:
DEFAULT MODE <Y OR N>

is displayed on an ODT. If "Y" 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 9-4.

Table 9-4. 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”

(*) If 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.

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 ODT during Coldstart follows:

Coldstart Messages

CODE MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
@41@ File not found	This error occurs internally to the SYSCOLDSTART program.	Request technical assistance.
@42@ Duplicate file- <filename> not loaded	Occurs when loading files to disk. If a file was loaded from a B900RL# disk that has the same file-id as one on a subsequent B900RL# disk, this error occurs.	Warmstart the system and copy the correct file.
@43@ <drive mnemonic> disk controller error	The disk processor has reported to COLDSTART that a controller error has occurred. This is a hardware error in the disk detected by the disk processor.	Request technical assistance.

CODE MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
@44@ Invalid system error number	The number reported to the system error procedure is not a valid COLDSTART error number.	Request technical assistance.
@45@ <drive mnemonic> device error	A device error was reported for the specified disk. This is a hardware detected error.	Request technical assistance.
@46@ Write inhibited	The specified disk is write disabled.	Write enable the disk. Refer to CP 9500 Operator's Manual for Instructions, form 1118270.
@47@ Name list full	There are no available entries in the file name list. Therefore, no files can be loaded.	Reformat disk increasing the number of files. (May require use of disk media with greater storage capacity.)
@48@ No user disk	All available disk space on a cartridge has been utilized.	Remove any unwanted file(s) and/or squash the disk.
@49@ No disk space	The available disk space on all of the fixed disks has been utilized.	Remove any unwanted file(s) or squash the disk.
@4A@ Not ready	SYSCOLDSTART is trying to access a not ready disk.	Make the disk ready.
@4B@ Not single area file	The file to be patched is not a single area file.	Copy the file specifying "singlearea."
@4C@ Pack not online	A pack-id of a not ready disk has been entered.	Check input and re-enter or insert requested pack in drive.
@4D@ Incorrect file type	When loading or patching system files, a check is done on their file type. The proper type must be present for the load and patch functions to successfully complete.	B900RL# disk may be corrupted. Try a backup copy of the disk or request technical assistance.
@4E@ Invalid drive	A drive which is not on the system has been specified.	Execute OL to see what peripherals are on-line. Check input and re-enter.
@4F@ Invalid label on drive	When performing an OL function, this message occurs, indicating that the label on the specified drive is not a CMS disk label.	None.
@50@ Number of bad sectors = nn	When initializing or reformatting, this message reports the number of bad sectors on the disk.	None, normal operation of initialize.
@51@ Extended pack files not handled	Files that extend over multiple packs cannot be loaded by COLDSTART.	Copy file to one pack.

CODE MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
@52@ Invalid pack	An attempt was made to initialize a pack named "B900RL#" or when entering drive for "B900RL1", the drive does not contain 'B900RL1" disk.	Insert proper disk.
@53@ No ready fixed disk on system	In response to the prompt "TYPE OF DISK?," "FIX" is entered and no ready fixed disks on the system exist.	Ready fixed disk .
@54@ Too many bad sectors	A disk containing more than 50 bad sectors is not initialized. COLD-START stops initializing when this event occurs.	Initialize another disk.
@55@ Bad track zero	CMS defines that track zero must be error free. The disk is not initialized when an error on this track is discovered.	Initialize another disk.
@56@ Enter release disk	When a B900RL1 disk is needed, this message occurs. It can occur at the beginning of COLDSTART if the SYSINITBOOT file is needed, or during the load, patch or replace functions.	Insert the B900RL1 disk.
@57@ No disk space for directory	During the initialize or reformat function, if the disk has bad sectors there may not be enough contiguous good sectors for the directory size chosen.	Reduce the directory size or use another disk.
@58@ Disk is not in initial state	When performing the load function the object disk must be in an initialized state.	Initialize or reformat the disk.
@59@ Duplicate packid	An attempt was made during the initialize or reformat functions to label the disk with the same name as one already on-line.	Power off the "duplicate" pack or specify another name.
@5A@ No bootstrap file	At the beginning of COLDSTART the SYSINITBOOT file is loaded from the B 900RL1 disk into processor memory. If this file is not found, this message will occur.	Warmstart the system with a disk that has the SYSINITBOOT file and copy it to B900RL1.
@5B@ B900 System load incomplete	After the load function has performed the file transfers to the system disk, a check is done to see if the necessary files have been loaded. If all of the required files are not on the system disk, this message occurs.	Check the B900RL1 disk to see what files are missing. Copy those files to B900RL1.

CODE MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
@5C@ Bad MTR track	If one of the tracks to be used for MTR purposes, during the FE initialize function, has been determined to be bad, this message will occur.	Try FE initialize on another disk.
@5D@ Ready	This is not an error message. It indicates the status of the drives during the OL function.	None.
@5E@ Unexpected interrupt	Interrupts are disabled at all times during the execution of COLDSTART. This message will occur if an interrupt is received.	Request technical assistance.
@5F@ Decimal number > 65535	This message occurs when COLDSTART detects a decimal number greater than 65535.	Request technical assistance.
@60@ ODT not ready	This error code displayed if the ODT is not ready.	Power on the ODT.
@61@ Filesize error	When computing area sizes for files, if the file area is larger than (65535 minus allocation unit) sectors.	Request technical assistance.
@62@ DP bad prep status	A bad status error was reported (that is, a status is not available for each active device).	Request technical assistance.
@63@ DP-bad status	An invalid command status result was returned by the disk processor.	Request technical assistance.
@64@ DP-unsolicited status	A command complete status was returned by the disk processor for an invalid drive.	Request technical assistance.
@65@ DP-bad ready status	An unexpected READY status message is returned for a ready drive.	DS function and repeat. If message reappears, request technical assistance.
@66@ Fixed disk not ready	The fixed disk was selected as the system disk and it is not ready.	Power on the fixed disk.
@67@ Not extendable	The system entry in the PPIT does not contain a logical unit of @FF@.	Reformat fixed disk.
@68@ PPIT mismatch	Before loading to fixed disks, the fixed disks are checked for identical PPIT's.	Power off all fixed disks except the one being loaded to or re-initialize with all fixed disks powered on.
@69@ Duplicate logical unit	The same logical unit number has been assigned to two fixed disks.	Request technical assistance.

CODE MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
@6A@ Divide by zero	A zero divisor was sent to the divide routine.	Request technical assistance.
@6B@ Memory parity error	A hard read error occurred.	Request technical assistance.
@6C@ No system pack	Load has detected an allocation unit of zero while attempting to load a multi-area file.	Re-initialize or reformat disk.
@6D@ Version mismatch	When patching, a check is done on the version string source file.	Use a more current B900RL1 disk or use replace function.
@6F@ Invalid disk	Device ID returned by the valid ID.	Request technical assistance.
@71@ Invalid bootstrap file	The Bootstrap file contains an invalid check string.	Use another copy of SYSBOOTSTRAP.
@73@ Function DS'ED	This message will occur when ?DS is entered to discontinue the present function.	None.
@74@ Patch level error	When executing the patch function, the patch file must be a later release than the source file.	Try a more current B900RL1 disk or use the replace function.

During Coldstart Bank "D" will contain a code reflecting the Coldstart function currently operating. The Coldstart functions are outlined on the next pages. This information should be used in conjunction with the previous information.

BANK D DISPLAY DESCRIPTION

BANK D DISPLAY

MEANING

@10@	The initialization function of COLDSTART is in operation. Banks F, G, and H contain an incrementing track or sector location for pack devices. For disk devices, Bank E contains an IVR pass number and Banks G and H contain the decimal cylinder number of the cylinder being IVR'ed.
@11@	The write-address logic of the initialize function is executing. Banks F, G, and H contain an incrementing address being written to disk.
@12@	The accuracy-test logic of the initialize function is executing.
@13@	The build-Track-0 logic of the initialize function is executing.
@14@	The build-non-file-directory logic of the initialize function is executing.
@15@	The build-non-file-name-list logic of the initialize function is executing.

BANK D DISPLAY

MEANING

@16@	The build-disk-file-name-header-list logic of the initialize function is executing.
@17@	The build-SYMEM logic of the initialize function is executing.
@18@	The build-PPIT logic of the initialize function is executing.
@20@	The load function of COLDSTART is in operation.
@30@	The reformat function of COLDSTART is in operation.
@33@	The build-Track-0 logic of the reformat function is executing.
@34@	The build-non-file-directory logic of the reformat function is executing.
@35@	The build-disk-file-name-list logic of the reformat function is executing.
@36@	The build-disk-file-header-list logic of the reformat function is executing.
@37@	The build-SYMEM logic of the reformat function is executing.
@38@	The build-PPIT logic of the reformat function is executing.
@40@	The OL function of COLDSTART is in operation.
@50@	The WARMSTART function of COLDSTART is in operation.
@60@	The HELP function of COLDSTART is in operation.
@70@	The FE initialize function of COLDSTART is in operation.
@80@	The patch function of COLDSTART is in operation.
@90@	The replace function of COLDSTART is in operation.

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 above 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 first action involved in dumping to disk is the location of the dump disk and the location on that disk of 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	---	---

The following table 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 the ROM Dump Display Reference Table which appears at the end of the 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 above.
38	Dump routine failed while checking disk label prior to dumping to disk.	Same action as above.
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 above.
C8	An write error was detected when dumping to disk.	Same action as above.
FF	A Fatal error. A hardware detected error has occurred which gives an unexpected error status.	System clear is necessary.
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 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 text above.
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.

ERROR CODE	POSSIBLE CAUSES	SUGGESTED ACTION
02	A disk read error has occurred.	Same action as above.
03	A disk write error has occurred.	Same action as above.

ROM Dump Display Reference Table

Tables 9-5 and 9-6 contain ROM Dump display information.

Table 9-5. 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 follows:

Table 9-6. 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 disk units to be logically concentrated into one large available area. Files then can 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 he/she so desires.

Many users run several different application programs during the course of a day. Often, these programs require the disks to be labeled 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 either does not have to relabel the disk every time he/she wants to run an application, or to modify the programs to change the disk name they presently require.

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 Intrinsic 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 relabeled 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 ODT 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 ways to visualize this system is as terminals which are some distance from the system itself.

In a school system, for example, the computer processor, ODT, 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 ODT, 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 communication 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.

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 also some disadvantages. The pluses are that tapes:

1. Tend to be less expensive storage media than disk.
2. Can make it easier to move information from one computer system to an unrelated system.
3. Use less external storage space than disks.

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 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.

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 warmstart 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. Then files on that pack can 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 that, when warmstarting the system when pack is the system disk required, that some disk unit (fixed, cartridge, or mini) 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 0 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 fifty (50) bad sectors, which 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 follows:

Table 9-7. 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, 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 B9387 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 B9387 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: TOO MANY BAD SECTORS.

NOTE:

In the 3.04 system software release the message: TOO MANY BAD SECTORS may not always be correct. However, it is an indication that there is a problem with the IVR process. If this message appears during pass @00@ of an IVR, then it is not an indication that the disk pack media is bad. It is an indication that the system is experiencing hardware failures.

2. If a verify operation fails for any reason other than a read parity error or a disk address error, the IVR aborts with the message: CANNOT VERIFY CYLINDER.
3. If an initialize data field operation fails for any reason other than a read parity error or a disk error, then the IVR aborts with the message: CANNOT INIT CYLINDER DATA FIELDS.

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 (5) 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 sector 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: $445 * \text{CYLINDER} + 90 * \text{HEAD} + \text{SECTOR} = \text{ADDRESS}$
205 DISK PACK: $445 * \text{CYLINDER} + 90 * \text{HEAD} + \text{SECTOR} = \text{ADDRESS}$

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 bands "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 labeled as: OS PRC1/OS PRC2 and DSK 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 such 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, DSK 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 PRC2 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 in the section on Warmstart.

MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
<FILENAME>NOT FOUND	One of the following: a.<packid> was of valid length and contained valid characters, but identified. wrong pack. b.<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 BIL 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)*

The AP intrinsic is used to create a pseudo-pack on fixed disk. For more details on pseudo-packs refer to the Special Considerations subsection.

The pseudo-pack cannot be created on any removable media.

The syntax for this command is:

```
AP <pseudo-packid>/
      |
      |-----<fixed disk name>-----|
      |
      |-----ON-----|
```

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 PSEUDO1/
```

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 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.

*Pseudo-Pack and Pseudo-Disk are interchangeable terms in this context.

Once a pseudo-pack has been created, it cannot be removed unless the fixed disk is initialized. However, the pseudo-pack can be relabeled.

OUTPUT MESSAGES

MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
AP<PACKID>/ COMPLETE	The command was completed successfully.	None - Normal operation.
AP<PACKID>/ INVALID	The pack-id specified is not legal.	Re-initiate with a valid packid.
AP<PACKID>/ INVALID PPIT FULL	The PPIT is full.	Use RL intrinsic to re-label an existing pseudo-pack that is no longer required.
AP<PACKID>/ DUPLICATE	A duplicate pack condition has been detected.	Re-initiate with a different pack-id.
AP<PACKID>/ NOT ON SYSTEM	The peripheral specified is not ready.	RYthe specified device and re-initiate.
AP<PACKID>/ INVALID	The peripheral specified is not a fixed disk or is not on the system.	Re-initiate specifying a valid peripheral.

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

RD < device mnemonic >

RD < device mnemonic >

BINARY.COPY

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.

AD < mix no > < device mnemonic >
(source)

AD < mix no > < device mnemonic >
(target)

SYSTEM MESSAGES

< device mnemonic > < packid > /RESERVED

< device mnemonic > < packid > /RESERVED

Note: The write-disabled disk does
not list its packid.

ENTER TARGET PACKID

< mix no > /BINARY.COPY ACPT

< mix no > /BINARY.COPY WAITING SOURCE/SYSMEM

< device mnemonic >

DEVICE ASSIGNMENT REQUIRED

< mix no > /BINARY.COPY WAITING TARGET/SYSMEM

< device mnemonic >

DEVICE ASSIGNMENT REQUIRED.

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 ODT 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 ODT 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:

```
AD <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: 2 3/6 MINI DRIVES  
OR  
1 1 MB MINI DRIVE  
SYSTEM DISK: 2 205/206 DISK PACK DRIVES
```

With this configuration, BINARY.COPY is unable to execute off the system disk if the disk packs are to be duplicated. Thus, to use BINARY.COPY, the MCP must be run off the loader disk drive. This requires the following files be present on the mini disk:

```

SYSBOOTSTRAP
SYSDSCP
SYSHOSTLOAD
SYSWARMSTART
SYSMCP
SYSICP
SYSMPLII
SYSLANGUAGE
SYSTRANSLATE
SYSCONFIG
BINARY.COPY
SYS-SUPERUTL

```

The mini 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
Null input is not valid	Blanks were entered for the target pack-id.	Check input and reenter.
Imbedded blanks are not valid	A pack-id containing imbedded blanks was entered.	Check input and reenter.
Invalid character in pack-id: <pack-id > <pointer>	A pack-id containing a character other than "A" thru "Z", "0" thru "9", "." or "-" was entered. <Pointer> is an asterisk appearing beneath the first invalid character.	Check input and reenter.
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.

MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
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 and 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 copied as binary zeroes. This message appears on line printer report only.	Run KA utility against target disk. Check list of sectors in error against KA. If sectors are within file areas, remove files from target disk and recopy from backup disk using COPY utility.
<Number> input disk 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 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.

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 create the SYSCONFIG file, update it, or list its contents. 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 ODT 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.

Figure 9-5 summarizes the structure of SYSCONFIG.

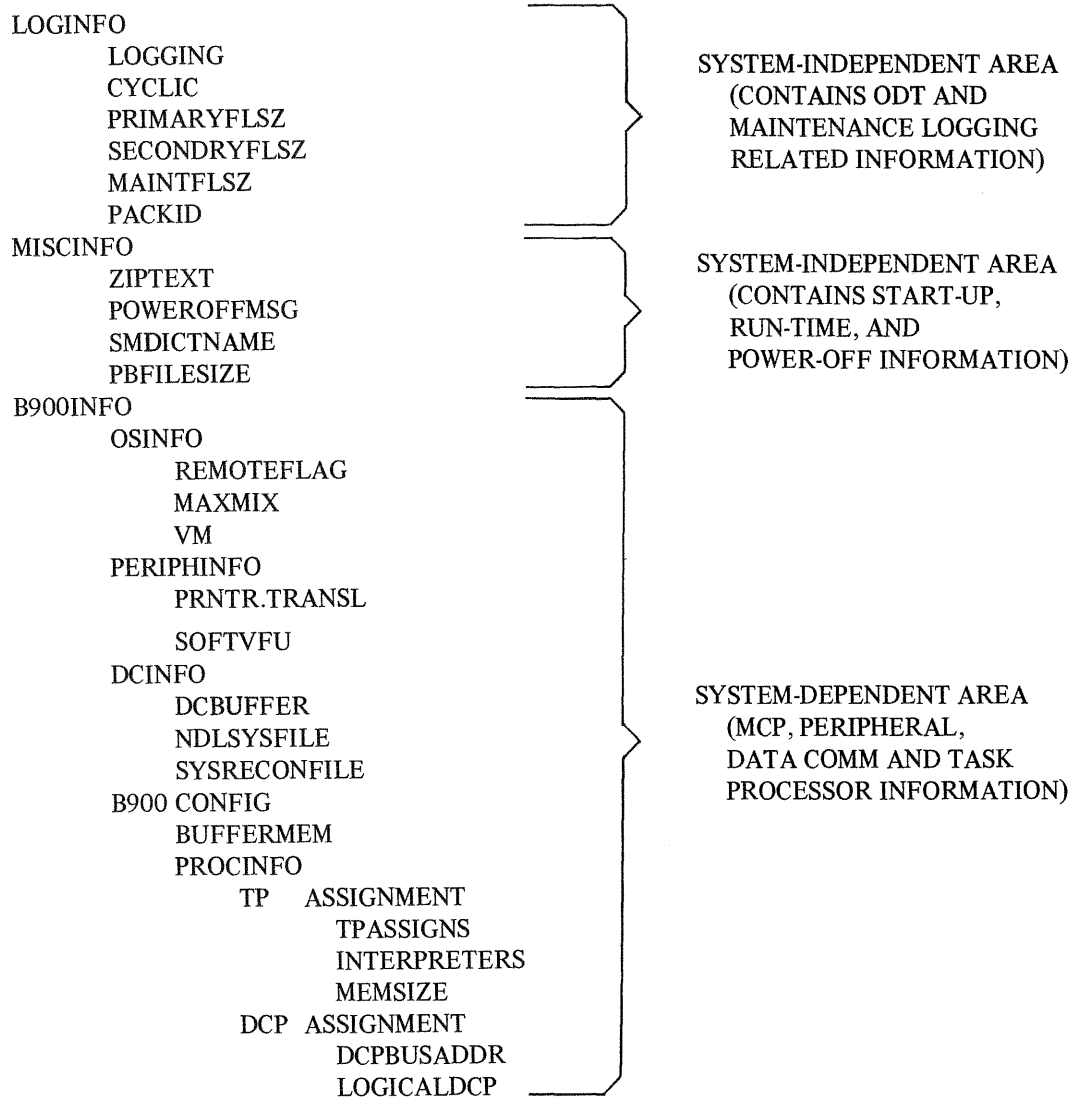


Figure 9-5. Summarized Structure of SYSCONFIG

Initiation

The initiating syntax for CONFIGURER is as follows:

CONFIGURER

OPERATOR ENTERS

CONFIGURER

SYSTEM MESSAGES

<mix no>/CONFIGURER BOJ PR IS X
(X is the job priority)

<mix no>/CONFIGURER DISP:B 900 CONFIGURER UTILITY

<mix no>/CONFIGURER DISP: CONFIGURER OPTIONS ARE:
MAKE, FIX, LIST, HELP, END

<mix no>/ CONFIGURER DISP:SELECT CONFIGURER OPTION

<mix no>/CONFIGURER ACPT

AX <mix no> <option>

OPTION SELECTION

AX <mix no> LIST

AX <mix no> MAKE

AX <mix no> FIX

To select HELP or END enter:

AX <mix no> HELP or

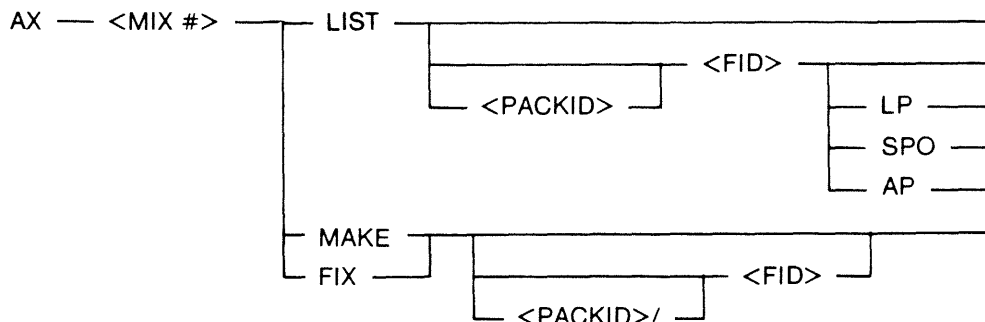
AX <mix no> END

Entering "HELP" causes configurer to display additional information for the selection of MAKE, FIX, or LIST modes. The request for the selection of a "CONFIGURER OPTION" is then repeated.

Entering "END" sends CONFIGURER to EOJ.

When selecting LIST, MAKE, or FIX, a filename may be specified. If no filename is specified, then the default is "SYSCONFIG". If no packid is specified, the system disk packid (seven ASCII zeroes or SYSPACK) is used as the default. Therefore, if no pack-id or filename is specified, then the default is "0000000/SYSCONFIG".

Any user response that can be derived by following this railroad diagram from left to right is expected:



List Mode

Selecting configurer's List mode causes CONFIGURER to find a SYSCONFIG file that has the specified (or default) file name, and list its contents on the ODT, the line printer, or whatever printer becomes available.

An example of LIST mode's line printer output is shown in figure 9-6. This example shows the values of the SYSCONFIG file as they are provided with the 3.04 B 900/CP 9500 CMS software release.

```

                B 900 CONFIGURER Utility
LIST OF          B 900RL1/SYSCONFIG
ON              12/11/81
AT             15:45:59
SYSCONFIG AREA LOGINFO
FIELD          MEANING/CONTENTS
LOGGING        YES
CYCLIC         YES
PRIMARYFLSZ    256
SECONDRYFLSZ   512
MAINTFLSZ      256
PACKID         0000000
SYSCONFIG AREA MISCINFO
FIELD          MEANING/CONTENTS
ZIPTXT        < EMPTY FIELD >
POWEROFFMSG    SYSTEM DISK POWERED OFF
SMDICTNAME     SYSLANGUAGE
PBFILESIZE     9999
SYSCONFIG AREA B900INFO
B900INFO AREA OSINFO
FIELD          MEANING/CONTENTS
REMOTEFLAG     YES
MAXMIX         26
VM             A
B900INFO AREA PERIPHINFO
FIELD          MEANING/CONTENTS
PRNTR.TRANSL   NO TRANSLATION
SOFTVFU        A
B900INFO AREA DCINFO
FIELD          MEANING/CONTENTS
DCBUFFER       40
NDLSYSFILE     0000000/NDLSYS
SYSRECONFILE   SYSRECON
B900INFO AREA B900CONFIG
FIELD          MEANING/CONTENTS
BUFFERMEM      80
B900CONFIG     TP ASSIGNMENTS
                INTERPS  MEM
                4  COBOL  00000256
                MPLII
                1  COBOL  00000128
                MPLII
B900CONFIG     DCP ASSIGNMENTS
                BUS LOGICAL
                8  0
```

Figure 9-6. Sample SYSCONFIG File

Make Mode

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 displays messages that lead the user through the areas and fields of the SYSCONFIG file by successively naming areas and fields until a complete SYSCONFIG file is generated.
3. After all fields have been processed, the workfile is closed with Remove. It is given the filename which the user specified (or allowed to default) in the message that selected "Make" mode. The only way to exit make mode is to DS configurer using DS <mix no>/CONFIGURER. Otherwise, all sections of the configuration are presented.
4. The user is returned to the 'SELECT CONFIGURER OPTION' level, to select another option.

SYSTEM MESSAGES

<mix no>/CONFIGURER DISP: CURRENT AREA IS: <field/area name>

<mix no>/CONFIGURER DISP: ENTER <field name> <value choice>

The operator enters:

AX, <mix no> <area name>

The system then displays

<mix no>/CONFIGURER DISP ENTER <field name> <choice of values>

The operator enters:

AX <mix no> <value choice>

Fix Mode

Selecting "FIX" mode causes the following to happen:

1. CONFIGURER finds 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 old SYSCONFIG file are copied into the work file.
3. If a SYSCONFIG file does not have a system dependent area, FIX mode will create one. As in MAKE mode, the user is requested to supply responses for all fields primed until a complete system dependent area is generated.
4. If a SYSCONFIG file already has a system dependent area, FIX mode will allow the user to select specific fields in any of the three major areas (LOGINFO, MISCINFO, B900INFO) in a SYSCONFIG file for updating. The operator can select one of the three areas to "fix".
5. When FIX mode finishes, it closes the workfile with remove, giving it the specified (or default) file name.

When updating a complete SYSCONFIG file, the user is allowed to select specific fields. This is done by presenting the user with a choice, one level at a time, of all areas/fields that are subsidiary to the area which the user currently is updating.

SYSTEM MESSAGES

<mix no>/CONFIGURER DISK <area-name> OPTIONS ARE: <area names>

<mix no>/CONFIGURER DISP SELECT <area name> OPTION

The operator then enters

AX <mix no> <area name>

Selecting an area name causes FIX mode to enter that area, and repeat the message, this time listing the areas/fields subsidiary to the area the user selected.

LOGINFO Area

The LOGINFO area of a SYSCONFIG file is system independent. It contains information on the logging of ODT messages.

When specifying a value for a field in the LOGINFO area, please observe the following syntax:

LOGGING	Enter "YES" ("Y") or "NO" ("N"). The LOGGING field indicates whether ODT messages are to be recorded in log files. (Default = "YES").
CYCLIC	Enter "YES" ("Y"), or "NO" ("N"). The CYCLIC field indicates whether cyclic or non-cyclic logging is to be performed. (Default = "YES").
PRIMARYFLSZ	Enter the file size (from 32 to 16,383 sectors) of the primary log files. The primary log files are SYS-LOG-01 through SYS-LOG-04. (Default = 32).
SECONDARYFLSZ	Enter the file size (from 32 to 262,140 sectors) of the secondary log file. This field is required when non-cyclic logging is to be performed. The secondary log files are SYS-HLDJJJ00 where JJJ is the Julian date and SYS-LOG-HOLD (Default = 32).
MAINTFLSZ	Enter the file size (from 32 to 262,140 sectors) of the maintenance log file. This field is required for both non-cyclic and cyclic logging.
PACKID	Enter the PACK-ID of the disk where the secondary and maintenance log files will reside. "0000000" denotes the system disk and is the default. (Default = "0000000").

MISCINFO Area

The MISCINFO area of a SYSCONFIG file is also system independent. The MISCINFO area consists of four fields:

ZIPTEXT
POWEROFFMSG
SMDICTNAME
PBFILESIZE

When specifying a value for a field in the MISCINFO area, please observe the following syntax:

ZIPTXT	Enter from 0 to 255 valid ASCII alphanumeric characters. The message in ZIPTXT must constitute a valid SCL command. The ZIPTXT field (if not null) contains the message that the CMS super utility is to zip as soon as it starts executing (that is, the name of a program to be executed). (Default = <empty>).
POWEROFFMSG	Enter from 0 to 80 valid ASCII alphanumeric characters. The message supplied in the POWEROFFMSG field is to be displayed on the ODT at the time that the system disk is powered-off. (It is therefore the last message that the system displays.) (Default = < empty >).
SMDICTNAME	Enter a valid CMS < file-id>. System messages to the SPO are generated by using a system message dictionary file. The SMDICTNAME field specifies the name of the dictionary file that is to be used. (Default = "SYSLANGUAGE").
PBFILESIZE	Enter a decimal integer between 1 and 1048560. All printer backup files generated on a CMS system have the same size limit (in bytes); this limit is specified in the PBFILESIZE field. (Default = 0).

B900INFO

The B 900/CP 9500 system dependent area ("B900INFO") must be present in any SYSCONFIG file used to warmstart a B 900/CP 9500.

This area consists of four subsidiary areas:

1. Operating system information ("OSINFO").
2. Information about peripherals ("PERPHINFO")
3. Data Comm subsystem information ("DCINFO").
4. Warmstart information ("B900CONFIG").

These areas are described in the following paragraphs.

OSINFO

This area contains information on the MCP and consists of three fields:

REMOTEFLAG
MAXMIX
VM

When specifying a value for a field in the OSINFO area, please observe the following syntax:

- REMOTEF**LAG Enter one of the following: "YES", "Y", "NO", "N".
The "REMOTEFLAG" field indicates whether the system is to allow a message control system (MCS) program to act as the ODT. (This allows a data comm device to be remote OCT.) (Default = "NO").
- MAXMIX** Enter a decimal integer between 1 and 26. The "MAXMIX" field sets a limit on how many jobs can be in the mix at one time.
- VM** Enter a null entry. This field is not currently used by the B 900/CP 9500.

PERIPHINFO

The second area subsidiary to "B900INFO" is named "PERIPHINFO". It contains information concerning peripherals (other than Data Comm and disk devices). This area consists of two fields:

PRNTR.TRANSL SOFTVFU

When specifying a value for a field in the PERIPHINFO area, please observe the following syntax:

Table 9-8. CONFIGURER Printer Translation

PRNTR.TRANSL The PRNTR.TRANSL field names the translation table to be used for the printer. (Default = NO TRANSLATION.) Enter one of the following table names:

VERSION 1	VERSION 2 (Cont.)	VERSION 3
AUSTRALIA	KENYA	ALGERIA
CANADA	MALAYSIA	BELGIUM
KOREA	MALTA	FRANCE
NEW ZEALAND	NETHERLANDS	FRENCH GUIANA
USA	NIGERIA	FRENCH WEST INDIES
	NORTHERN RHODESIA	MALAGASY REPUBLIC
	RHODESIA	MOROCCO
	SINGAPORE	TUNISIA
	SOUTH AFRICA	WEST AFRICA
	TAIWAN	ZAIRE
	THAILAND	
	TRINIDAD	VERSION 4
	UNITED KINGDOM	ITALY
	USA	
		VERSION 5
		AUSTRIA
		CZECHOSLOVAKIA
		GERMANY
		SWITZERLAND

VERSION 6

BRAZIL
PORTUGAL
PORTUGUESE
WEST AFRICA
PORTUGUESE
EAST AFRICA

VERSION 7

ARGENTINA
CHILE
COLUMBIA
COSTA RICA
DOMINICAN REPUBLIC
ECUADOR
HONDURAS
MEXICO
NICARAGUA
PANAMA
PERU
PHILIPPINES
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

The above tables are to be used for the B 9249-2/3/4 line printers.

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 128char" be specified.

SOFTVFU This field is not currently used by the B 900/CP 9500.

DCINFO

The "B900INFO" area's third subsidiary area contains information that concerns the Data Comm sub-system. Its name is "DCINFO".

This area consists of three fields:

DCBUFFER
NDLSYSFILE
SYSRECONFILE

When specifying a value for a field in the DCINFO area, please observe the following syntax:

DCBUFFER Enter a decimal number.
The "DCBUFFER" field specifies how many kilobytes
(1 KB = 1024 bytes) of the system's buffer memory
must be dedicated to data comm. (Default = 0).

NOTE

This specifies how much of the system buffer memory is to be reserved for Data Comm. The total amount of buffer memory is determined by BUFFERMEM (in the B900CONFIG area). When DCBUFFER or BUFFERMEM is specified, CONFIGURER checks if both values have been entered, that the amount of DCBUFFER to be reserved is less than the total amount of BUFFERMEM by at least 2 KB. It checks that (BUFFERMEM - DCBUFFER) > 2. If this is not true, CONFIGURER requests new values for both fields after displaying the following message on the ODT:

★★ invalid BUFFERMEM specs ★★

NDLSYSFILE Enter a valid CMS file name (that is, <packid>/<file-id>). Do not enter more than 20 characters. "NDLSYSFILE" contains the file name of the NDLSYS file that is to be used in setting up the CMS data comm subsystem. (Default = "0000000/NDLSYS").

SYSRECONFILE Enter a valid CMS <file-id>. Do not enter more than 12 characters. "SYSRECONFILE" specifies the file-id of the system reconfiguration file. Any data comm configuration changes made during operation are recorded in the reconfiguration file. (NDLSYS is a read-only file.) (Default = "SYSRECON").

B900CONFIG

The system dependent area ends with an area that contains information needed to Warmstart the physical system. This area specifies which physical resources are needed and how they are to be used. "B900CONFIG" begins with a field:

BUFFERMEM
enter a decimal number.
BUFFERMEM specifies how many kilobytes
(1 KB = 1024 bytes) should be allocated for system
buffer memory use.

This is followed by the warmstart table, which specifies how the system task processors are allocated.

The configurer utility handles the warmstart table as an area named "PROCINFO" containing the following areas and fields:

PROCINFO
 TP ASSIGNMENT
 TPASSIGNS
 INTERPRETERS
 MEMSIZE
 DCP ASSIGNMENT
 DCPBUSADDR
 LOGICALDCP

When specifying a value for a field in the PROCINFO area, please observe the following syntax:

PROCINFO	<p>For each task processor or set of identical task processors, enter "TP" and specify the necessary values for the fields in the TP assignment area.</p> <p>For each data comm processor, enter "DCP" and specify the necessary values for the fields in the DCP assignment area.</p> <p>To cause CONFIGURER to make a warmstart table for the default configuration, enter "DEFAULT" without specifying "TP" or "DCP".</p> <p>To stop specifying processor assignments, enter "PROCEND".</p>
TPASSIGNS	<p>Enter a decimal integer from 1 to 6. Depending on the system, there can be up to six task processors on the system. Note that if there is no requirement for the DCP (data communications processor on the system), the DCP can be used as a task processor.</p> <p>All task processors that are to have identical attributes can be assigned in one pass through the "TP ASSIGNMENT" area. The TPASSIGNS field specifies how many TPS are being specified; interpreters names the interpreters that each of these TPS is to support; and memsize specifies the minimum amount (in kilobytes) of local memory that each of these TPS must own. (One page contains 64 KB.)</p>
INTERPRETERS	<p>Enter one of the following: "MPLII"; "COBOL" (that is, COBOL and/or RPG); "ALL" (that is, same as specifying MPLII and COBOL); "ENDINTERP". (Additional interpreters values are requested until "ENDINTERP" is specified, except for the "ALL" specification.)</p>
MEMSIZE	<p>Specify a decimal number. It must be between 64 and 1024.</p>

NOTE

The upper limit for the MEMSIZE parameter corresponds to 16 pages of local memory. This reflects the address capacity of the hardware registers. Currently, a task processor is not allowed to own more than four pages, which sets a practical limit of "256" for the MEMSIZE parameter. Task processors supporting more than one interpreter must have a minimum MEMSIZE of 128 KB.

The DCP assignment area specifies the bus address of each Data Comm processor and the logical DCP number associated with that bus address. When the CMS Data Comm subsystem is loaded, the logical DCP number determines which DCP firmware file is loaded into a given Data Comm processor.

For each DCP, a fresh pass through the DCP assignment area must be made.
For each DCPBUSADDR, enter a decimal integer between 1 and 15.

OS and Data Comm Buffer Memory Validation

In addition to the basic validity checks applied to any other single field, the utility will verify both the operating system and Data Comm buffer memory specifications against each other to assure their compatibility. This is done in either of the following:

1. MAKE MODE: At the time the OS buffer specification is received;
2. FIX MODE: When either the Data Comm Buffer or OS Buffer specification is being updated.

Ensure that the difference between the (OS buffer - DC buffer) is greater than or equal to the OS buffer minimum, where the OS buffer minimum is equal to 2000 bytes. Failing this verification will result in CONFIGURER issuing prompts for the re-specification of both buffer sizes.

The following table lists the maximum amount of buffer memory that can be specified in the SYSCONFIG file based on the amount of memory available on the OS processor.

TOTAL OS MEMORY	MEMORY USAGE
128 KB	32 KB AVAILABLE FOR BUFFER MEMORY
192 KB	64 KB AVAILABLE FOR BUFFER MEMORY
256 KB	128 KB AVAILABLE FOR BUFFER MEMORY

Processor Configuration

The initial selection list for processor assignment allows the following:

1. The initiation of a "TASK PROCESSOR" (TP) assignment sequence;
2. The initiation of a "DATA COMM PROCESSOR" (DCP) assignment sequence;
3. The specification that the default assignments are to be made at Warmstart time;
4. The specification that the user processor assignments are complete (that is, PROCEND).

If the default processor assignments scheme is sufficient, merely entering default will complete the processor assignment. This will cause the Warmstart defaults to be used.

If discrete user assignments are desired, there are two forms of processor assignment types defined: "TP" and "DCP". Each is implemented in a multi-response form. Each of the response parts will have a respective prompt by CONFIGURER indicating its attributes prior to the issuing of the accept for the user input.

The TP and DCP assignments may be used in any order, and the required number of times, such that the desired configuration is completely specified. This will be the logical system allocated at Warmstart time, given that the physical system can support the configuration.

NOTE

If the logical TP and DCP assignments do not match that of the physical system configuration, the system will at Warmstart time come up under its default configuration. Warmstart default is not the same as the SYSCONFIG default. Refer to the Warmstart section for the exact parameters.

NOTE

Do a LIST before ending the configuration procedure to verify that changes have been made correctly.

NOTE

If the TP assignments are changed using FIX mode, DCP assignments must be re-entered.

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 then can be sent on several mini disks.

DUMPFILe.UTL also will reconstruct the file to 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 _____ |

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 ACPT:
```

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 diskid and fileid of the system dumpfile to be partitioned and the diskid 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 ACPT:
<mix no>/DUMPFILe.UTL DISP: ENTER FILE-ID OF SYSTEM DUMPFILe
<mix no>/DUMPFILe.UTL ACPT:
<mix no>/DUMPFILe.UTL DISP: ENTER DISK-ID OF OUTPUT DISK
<mix no>/DUMPFILe.UTL ACPT:
```

NOTE

The first DISK-ID refers to the disk where the dumpfile is originally located. The user must respond with valid diskid's and fileid's when requested. DUMPFILe.UTL will display a message for each partition created and a completion message when partitioning is complete.

For example:

```
<mix no>/DUMPFILe.UTL DISP: <OUTPUT DISKID>/DMFILEPART00 CREATED
<mix no>/DUMPFILe.UTL DISP: <OUTPUT DISKID>/DMFILEPART01 CREATED
<mix no>/DUMPFILe.UTL DISP: <OUTPUT DISKID>/DMFILEPART02 CREATED
<mix no>/DUMPFILe.UTL DISP: <OUTPUT DISKID>/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 DUMPFIL.UTL will prompt the user to enter the diskid and fileid for the re-creation of a system dumpfile followed by the diskid of the disk that contains the system dumpfile parts. For example:

```
< mix no >/DUMPFIL.UTL DISP: ENTER DISK-ID OF DESTINATION
< mix no >/DUMPFIL.UTL ACPT
< mix no >/DUMPFIL.UTL DISP: ENTER FILE-ID OF DESTINATION
< mix no >/DUMPFIL.UTL ACPT
< mix no >/DUMPFIL.UTL DISP: ENTER DISK-ID OF INPUT DISK
< mix no >/DUMPFIL.UTL ACPT
```

The user must respond with the appropriate diskid's and fileid's as they are requested. DUMPFIL.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 >/DUMPFIL.UTL DISP: <INPUT DISKID>/DMFILEPART00 COPIED
< mix no >/DUMPFIL.UTL DISP: <INPUT DISKID>/DMFILEPART01 COPIED
< mix no >/DUMPFIL.UTL DISP: <INPUT DISKID>/DMFILEPART02 COPIED
< mix no >/DUMPFIL.UTL DISP: <INPUT DISKID>/DMFILEPART03 COPIED
< mix no >/DUMPFIL.UTL DISP: FILE RECONSTRUCTION COMPLETE
< mix no >/DUMPFIL.UTL EOJ
```

OUTPUT MESSAGES

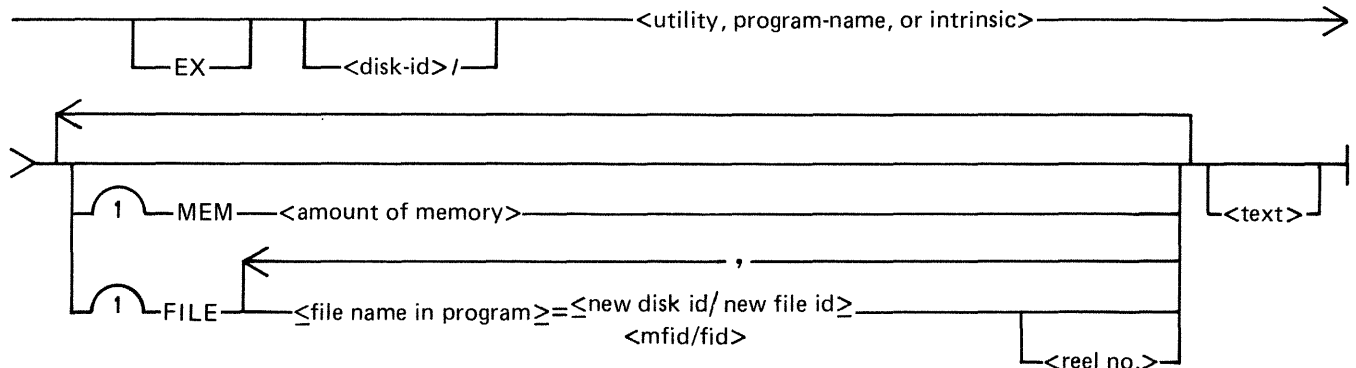
MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
Output disk space exhausted-- enter new output disk. Power off output disk using PO command. Then insert new output disk.	When partitioning a large dump- file more than one mini disk may be required to accommo- date all output files created.	Power off current output disk using PO intrinsic and insert a new output disk.
Next file in sequence is re- quired. 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 A P 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 packid.	Check the packid entered and re-enter correctly.
A valid file-id is 1-12 characters in length	More than twelve 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 packid or fileid is expected.	Determine input expected and enter it.

MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
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 packid or fileid entered.	Check all characters entered in either the packid or fileid and re-enter as necessary.
Unable to open output file	This message is displayed during file partitioning or reconstruction if DUMPFIL.UTL is unable to open the first output file.	Re-execute DUMPFIL.UTL.
Unable to open first input file	Dumpfile.utl is unable to open the first input file (DMFILEPART00).	If complete system dumpfile is still available, then run DUMPFIL.UTL to re-partition the file. Ensure that all dumpfile parts were created at the same time.
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 the first dumpfile part was.	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 DUMPFIL.UTL to reconstruct or partition dumpfile.
Irrecoverable error detected in write	A fatal error is detected when writing a file.	Re-execute DUMPFIL.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 ODT. EX can be used when the operator wishes to specify the amount of memory a program will need. Then the MCP can 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 which 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 disk (B900RL1) or a B 900/CP 9500 tape load/dump disk (B9LDMP) 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 B900RL1 disk, the Patch or Replace function of Coldstart 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:

FILE NAME	B900RL1	B9LDMP
SYSMCP	X	
SYSBOOTSTRAP	X	X
SYSINITBOOT	X	
SYSCOLDSTART	X	
SYSDSCP	X	X
SYSMPLII	X	
SYSICP	X	
SYSWARMSTART	X	
SYSHOSTLOAD	X	X
SYSTRANSLATE	X	
SYSLDMP		X

2. Every file present on the B900RL1 disk must be at the patch level specified in the supplied patch.
3. The patch must be entered correctly.

NOTE

The COBOL interpreter (SYSCOBOL) need not be on the release disk, but if present, it must be at the correct patch level in accordance with condition 2.

The syntax for initiating this program is:
FPP

The following message will be displayed on the ODT:
<mix #>/FPP DISP: ENTER PACKID: "B900RL1 OR B9LDMP"

The user responds via an "AX" command with the appropriate disk name. The authorization code includes the check code, old version code, and new version string. It is provided with the patch.

Next, the authorization code will be requested from the user:
<mix #>/FPP DISP: ENTER AUTHORIZATION CODE

The authorization code is entered in the following format:

AX <mix #> <check code> <old version string> <new version string>

where: <check code>	two groups of four hexadecimal digits (“nnnn nnnn”).
<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 version string that is to be given to each file on the release disk if all files have been patched successfully.

Every required file on a release disk must have the same version string as the <old version string> supplied in the authorization code. The presence of any version string that does not match the one in the authorization code causes FPP to terminate after displaying the message:
<mix #>/FPP DISP: INVALID RELEASE DISK

Entering the Patch

If FPP finds the release disk is valid, it goes on to accept the patch from the user.

To request the first line of the patch, FPP displays the following message on the SPO:
<mix #>/FPP DISP: ENTER PATCH (FILE SEG ADDR PATCH)

This message is not repeated for further patch lines; FPP merely continues to do ACCEPTs until the user responds:

AX <mix #> END

Format of Patch Line

Each line of the patch is entered in the following format:

AX <mix #> <file> <segment> <address> <patch>

where: <file> a two-character file mnemonic, which represents one of the system files:

File	Mnemonic
SYSMCP	OS
SYSBOOTSTRAP	BT
SYSINITBOOT	BZ
SYSCOBOL	CK
SYSCOLDSTART	CS
SYSDSCP	DP
SYSMPLII	MK
SYSICP	TK
YSWARMSTART	WS
SYSHOSTLOAD	HL
SYSLDMP	LD
SYSTRANSLATE	TR

<segment> a segment number composed of two hexadecimal digits. (It must equal zero for files other than than SYSMCP.)

<address> four hexadecimal digits, composing an address that is within the range of the segment.

<patch> an even number of hexadecimal digits (no more than forty).

Blanks may be added or omitted to aid in readability.

If an incorrect patch line is entered, it will be rejected by FPP.

Once FPP has determined the patch is correct, it updates the version string and enters the patch into the files.

OUTPUT MESSAGES

MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
Invalid entry	A packid other than B900RL1 or B9LDMP was entered.	Check input and re-enter.
Authorization code error	FPP has detected an incorrect check code.	Obtain correct code from Burroughs representative.
Illegal input character	An invalid character was detected in the input.	Check patch and re-enter.

MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
Input too long	Too many characters were entered in the authorization code.	Check patch and re-enter.
Input incorrect length	An incorrect number of characters was entered in the authorization code.	Check patch and re-enter.
Invalid release disk	One or more of the required system files does not have the correct version string or one of the required system files on the release disk may be corrupt.	Verify that the authorization code has been entered correctly and verify that all required system files are valid. The operator may use the COMPARE utility against a known good release disk or use a backup release disk.
Invalid patch re-enter patch	A patch line may have been entered incorrectly.	Verify input and re-enter patch.
Cannot find prime-resident segment	Release disk has become corrupt.	Verify that the release disk is valid by running the COMPARE utility against a known good disk or use a backup release disk.
Invalid segment number (must be 00 for non OS)	A non-zero segment number has been detected in a line of patch. All patch lines other than OS patch lines must contain zeroes as the segment number.	Check patch and re-enter.

GT CF (Get System Configuration)

This command allows the operator to display the system configuration. The following information is displayed on the ODT screen:

Bus address of each processor
Type of processor
Amount of memory on each processor in Kbytes
Total amount of buffer memory
Amount of data comm buffer memory

The syntax for this command is:

GT CF _____|

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	256
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 rewarmed. 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:

GT MD _____|

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:

GT MX _____|

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 are 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
```

PO (Power Off)

This command allows the operator to logically Power Off the system or user disks. The syntax is:
PO-<DISK MNEMONIC> _____|

If the system disk is a fixed disk, the command "PO DF=" should be used. When the message "SYSTEM DISK POWERED OFF" or "<disk mnemonic> POWERED OFF" is displayed, the operator should wait a few seconds before physically powering off the drive(s).

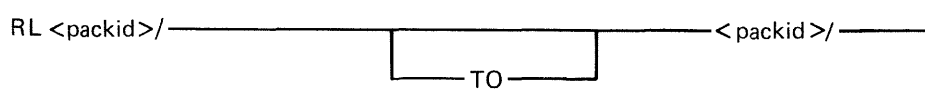
The operator can logically Power Off and rewarmstart system by entering the command "PO DLD". If any tasks are in the mix when this command is issued, they will automatically be DS-ed.

Whenever the system disk is logically powered off, any user disks on the system also will be logically powered off.

If an unused system dumpfile resides on the system disk, it will be removed as part of the system disk PO procedure.

RL (Relabel)

This intrinsic allows the operator to relabel the packid 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:

`RL <packid>/<packid>/ COMPLETE`
 (RL also can be done as a Coldstart option.)

Two conditions are required for this command to be accepted:

1. If a pseudo-packid is being relabeled, then all physical units that have packids listed in the PPIT must be physically ready.
2. A null mix must exist. On a system with a remote operator, the null mix requirement is satisfied if the only tasks in the mix are the MCS that is managing the remote system operation, and a program named "SPIM".

When the packid to be changed identifies a physical pack, (that is, not a pseudo-pack), this command changes the packid in the disk label. Therefore, the new packid will be on-line.

When the packid requested to be changed is a pseudo-pack, this command causes the entry for that **PACKID** in the PPIT to be changed to the new packid.

WARNING

RL does not scan the disk file header list and will not notice nor attempt to correct disk file headers of dual pack file components or Key File Parameter Blocks (KFPB's).

Therefore, if a file is located on two disks and one of those disks is relabeled, the disk file header will still locate the rest of the file under the original disk name. The keyfile also will point to the data file as being located on the original disk name.

OUTPUT MESSAGES

ERROR MESSAGE	POSSIBLE CAUSES	SUGGESTED ACTION
RL INVALID	An attempt has been made to relabel a disk without a null mix.	Re-enter when system has a null mix.
"RL <packid>/TO <packid>/COMPLETE"	The command was completed successfully.	None, normal operation.
"FAULTY SCL INPUT" NOT ON SYSTEM	The pack-id specified to be changed is not present.	Re-initiate RL with both valid packids.
RL PACKID/DUPLICATE PACK	A duplicate pack condition is detected.	If possible, PO the pack that is not being relabelled.
"FAULTY SCL INPUT" INVALID	The physical unit from the PPIT is not ready.	RY the specified device and reinitiate RL.
"FAULTY SCL INPUT" INVALID	The pack-id specified is not legal.	Re-initiate RL with valid packid.

RLD (Release Level Display)

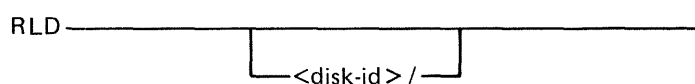
The Release Level Display (RLD) checks whether any of the following system files is present on a specific disk, and reports the release level of each of the following files:

A Sample Output follows:

```
<mix no>/RLD DISP: <disk-id>/SYSBOOTSTRAP-BT-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSCOBOL-CK-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSCOLDSTART-CS-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSDSCP-DP-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSHOSTLOAD-HL-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSICP-TK-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSINITBOOT-BZ-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSMCP-OS-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSMPLII-MK-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSTRANSLATE-TR-RELEASE LEVEL 03.04.08
<mix no>/RLD DISP: <disk-id>/SYSWARMSTART-WS-RELEASE LEVEL 03.04.08
```

The information appears on the ODT 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 to the first release disk (packid "B900RL1"). The packid for the system disk is "0000000" or "SYS-PACK".

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.

The files are sought in alphabetical order. 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 packid	A packid 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.

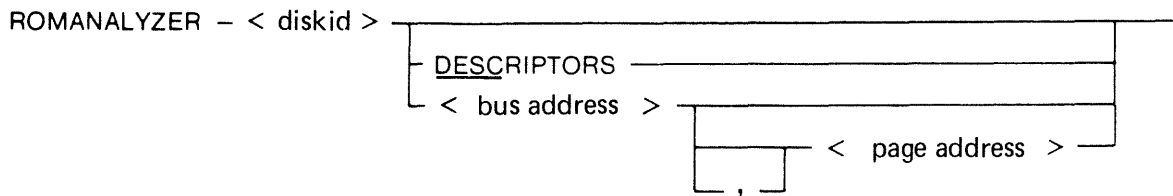
ROMANALYZER (ROM Dump File Analyzer)

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. < diskid > = diskid of the removable disk containing the ROM dump file. The packid 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 bus address to whose pages the ROMANALYZER program is to restrict itself (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# > / ROMANALYZER DISP: < BUS > < EMPTY >: , < PAGE >
```

```
< mix# > / ROMANALYZER ACPT
```

appear on the screen.

When the ACPT message appears on the ODT, enter one of the following:

```
AX < mix # > < bus address >
```

or

```
AX < mix # > < bus address >, < page address >
```

ROMANALYZER continues to request additional bus/page data until a null input (AX < mix # >) 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 ODT 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 @FFF8@).
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 <packid>

Romanalyzer will print the descriptor and the hex dump for all pages of the ROM dump file.

ROMANALYZER <packid> DESC

Romanalyzer will print only the descriptors for the pages generated on a particular disk.

ROMANALYZER <packid> <bus address>

Romanalyzer will print the descriptors and hex dumps for all pages of a particular bus address.

ROMANALYZER <packid> <bus address> <page>

Romanalyzer will print the descriptor and hex dump for a particular page.

ERROR	POSSIBLE CAUSES	SUGGESTED ACTION
**ERROR-NO INITIATING MESSAGE	A packid was not specified in initiating message.	Re-initiate ROMANALYZER with a valid packid.
**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.

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.

ERROR-EOF READ SYMEM, 1 *	**ERROR-LEADING SECTOR NOT ZERO'D OUT ***
ERROR-ERR READ SYMEM, 1 *	**ERROR-ERR IN FILE READ ***
ERROR-ERR SYMEM AT.ADDR + 1 *	**ERROR-EOF IN FILE READ ***
ERROR-EOF SYMEM AT.ADDR + 1 *	**ERROR-GET.RECORD READ ***
ERROR-ERR SYMEM INITIAL SEC *	**EOF-GET.RECORD READ ***
ERROR-EOF SYMEM INITIAL SECTOR *	DESC FOUND IN GET.RECORD
ERROR-BAD LEADING SECTOR *	**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:

```
ROMCONVERT _____|
```

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 DISK(S) <single digit>  
<mix no>/ROMCONVERT ACPT
```

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 packid of the source disk. The following messages will appear on the SPO:

```
<mix no>/ROMCONVERT DISP:  
    ENTER SOURCE DISKID  
<mix no>/ROMCONVERT ACPT
```

The user responds, via the "AX" intrinsic, with a packid (conforming to CMS requirements for a packid).

```
AX <mix no> <pack-id>
```

Next ROMCONVERT will request the packid and fileid for the creation of the converted dump file by displaying the following messages.

```
<mix no>/ROMCONVERT DISP  
    ENTER DESTINATION DISKID  
<mix no>/ROMCONVERT ACPT  
    (RESPOND WITH "AX" <mix no> <PACKID>")  
<mix no>/ROMCONVERT DISP  
    ENTER DESTINATION FILEID  
<mix no>/ROMCONVERT ACPT  
    (Respond with "AX" <mix no> <FILEID>")
```

The fileid 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 ODT 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 DISKID
```

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 #> <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.

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 packid specified.	Check syntax of packid and re-enter.
Entered input too long. A valid diskid is 1-7 characters in length. A valid fileid is 1-12 characters in length.	More than 7 characters entered for packid or 12 characters for a fileid.	Check input and re-enter.
Invalid character in input. A valid diskid or fileid may be constructed from the letters A-Z, the numbers 0-9, and the special characters hyphen (-) and point (.)	Invalid packid entered.	Check syntax of packid and re-enter.
Unable to open SYSMEM.	No disk with the specified packid is present.	Place correct disk on line and re-enter.

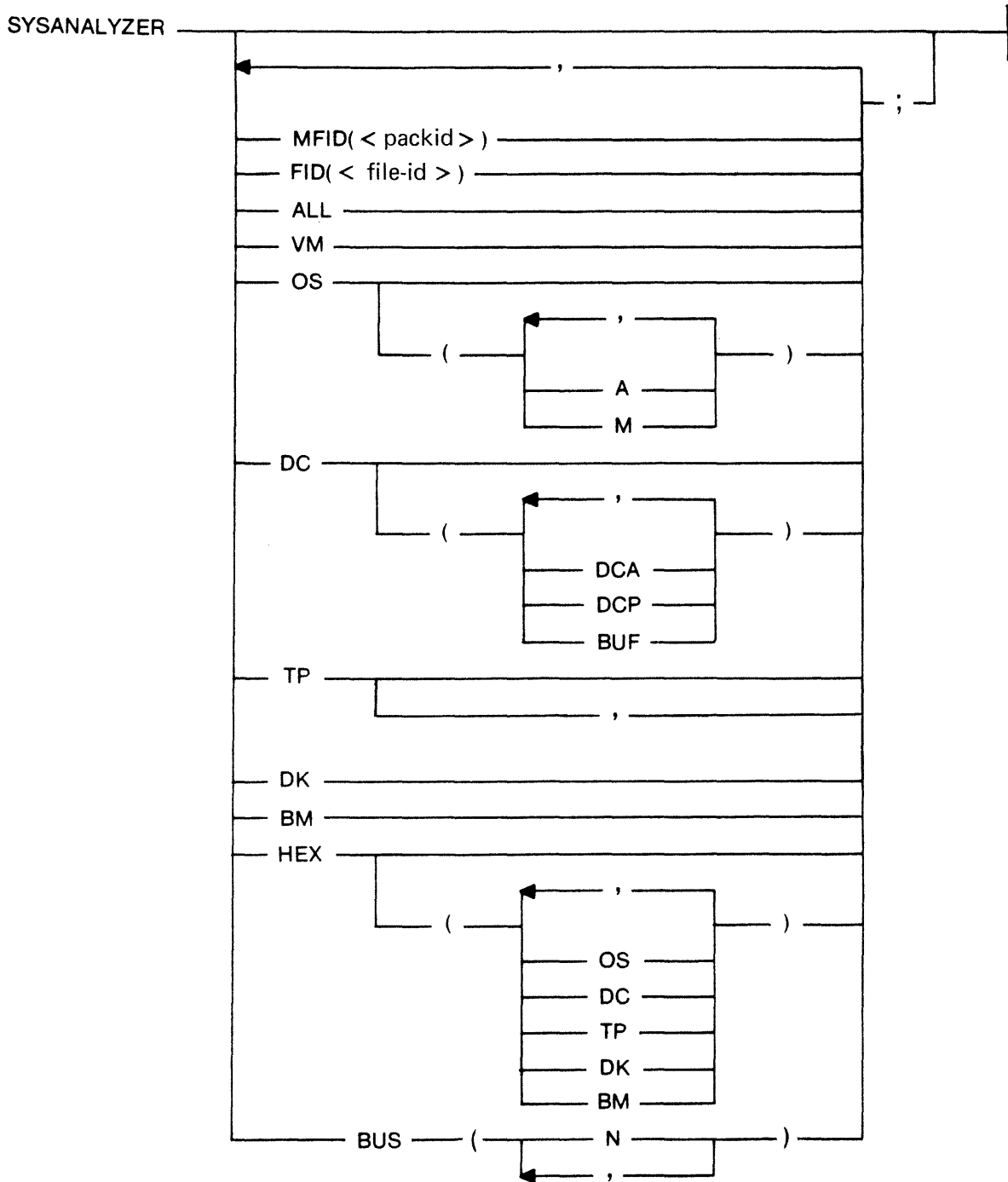
The following messages denote irrecoverable error conditions encountered during processing of the ROM dump file:

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

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 railroad diagram below. The options may be entered in any order. The analysis still will 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 labeled **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 DUMPFIL IS NOT LABELED 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;**

The two initiating messages above each 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 <packid> 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.

OPTION	MEANING
DC	<p>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 DCP's, use the "BUS" option. Suboptions of DC:</p> <p>(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.</p>
TP	<p>Perform an analysis of each task processor. To restrict analysis to individual task processors, use the "BUS" option, described below.</p>
DK	Perform analysis of disk processor.
BM	Perform analysis of buffer management.
VM	Perform analysis of virtual memory.
HEX	<p>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;</p> <p>(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.</p>
BUS (n)	<p>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.</p>

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 UTILITIES

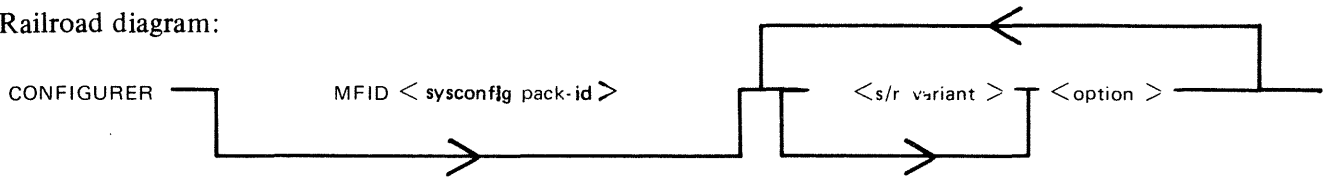
CONFIGURER

This utility creates and modifies the file "SYSCONFIG". Much of the information stored in this file is used at CLEARSTART time. Therefore, these options do not take effect until the next CLEARSTART. 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:

Railroad diagram:



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.

<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.
COUNT	Gives the number of CLEARSTARTs performed since the creation of SYSCONFIG file.
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. The values given by the user as replies to these prompts are used as initial values for the system time. RO DATE and/or RO TIME cause the system time to be initialized with the values contained in the SYSCONFIG file. These values are the actual system date and time at the last time these options were reset.

DCPDUMP
<num> Controls whether the memory of the DCP specified by <num> is to be dumped to a file at DCSS initiation time.
 <num> must be in the range 0 to 4.
 The dump will be contained in the file DUMPDCP<num>, where <num>, as above, is the number of the DCP.
 SO DCPDUMP <num> causes a dump to be created the next time Data Comm is started.
 RO DCPDUMP <num> inhibits a dump. When this option is reset, no DCP dump information is displayed in a CONFIGURER DISPLAY or PRINT.

DCSIZE <num> Specifies the amount of memory allocated for the DCSS message area in K bytes.
 <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 is reserved for future use.

HELP Causes the syntax diagram for CONFIGURER to be displayed on the SPO.
 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.

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 >

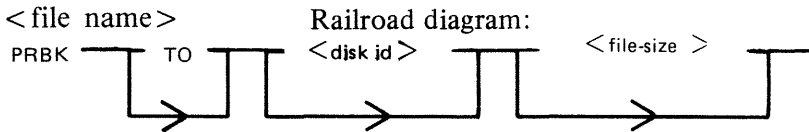
Sets the maximum number of areas per disk file. The maximum value allowed for < num > is 16.

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.

MESSAGE FILE
< file name >

Enables the user to specify the desired message file.



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.

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.

REMOTECHAR
< char >

Specifies the control character for REMOTE SPO activities.

< char > denotes the desired character.

SO and RO are not valid for this option.

REMOTEFLLAG

Controls whether remote SPO activities will be allowed or inhibited.

SO REMOTEFLLAG sets the REMOTEFLLAG at DCSS initialization, allowing REMOTE SPO activity.

RO REMOTEFLLAG 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.

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.
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. 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.
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 printed. 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 of SYSCONFIG contains the compile date of the version of CONFIGURER which created that SYSCONFIG.

Defaults

The values of the SYSCONFIG file included in the 3.04 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	3,04.03
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	000000
BACKUP FILE SIZE	10000 @2710@
NUMBER OF SHARED FILES	32
MAX USERS	63

All other DCSS information is reset.
 ZIP information is reset.
 The backup option, PRBK, is reset.

Update Messages

DCCH TRACE FLAG SET [RESET]
 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 FLAG SET [RESET] (NUMBER OF LOG FILES : xx, LOG FILE SIZE : yy)
 DUMP DCPx SET [RESET]
 BACKUP OPTION SET [RESET] (BACKUP DESIGNATE : "xxx", BACKUP SIZE "yyy"
 RECORDS)
 BACKUP DESIGNATE DISK "xxx" CHANGED TO "yyy"
 BACKUP FILE SIZE CHANGED FROM xxx RECORDS TO yy
 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 >

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 two 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.

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)

<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.

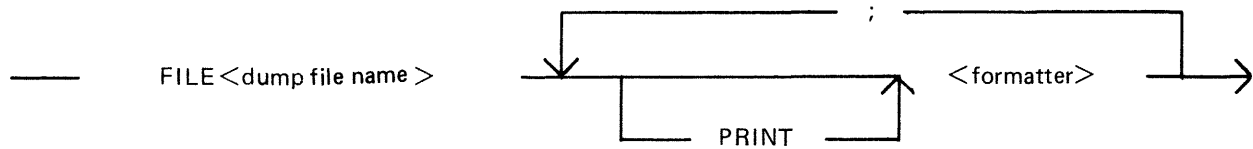
If the output takes more than a page then "... " appears at the bottom left hand corner of the screen, and the user must send a blank input to get the next page. For a terminal the cursor is positioned to send one space character.

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:

Railroad diagram:



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

BYE
CLOSE PRINTER
DCP-1
DCP-3-4
DISPLAY
END
FILE
FORMS
GET
HELP
PRINT
WHAT

Formatters

CONFIGURATION
DUMP
LIEGE.HDR
LINE
LINE.HISTORY
LINE.INFO
LINE.LINKAGE
LINE.TABLE
LIST.AVAIL
LIST.MSG
LIST.QUEUE
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.

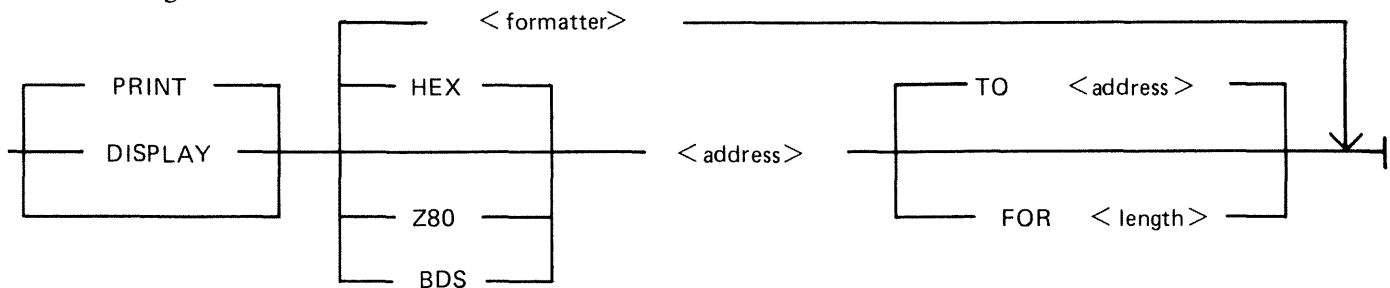
DISPLAY

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:

Railroad diagram:



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. A few “...” are present at the end of the line if the message is longer than 61 characters.

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 a “DISK ERROR” message will be displayed. The meaning of this message is “SYSMEM LOCK”.

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. After the process, both disks are identical, except that they keep their own label. Note that the output disk label may be changed via the initiating message.

It is clear that a copy of a disk to itself (after a transit on a tape) will not be affected by the presence of bad sectors on the disk.

Each sector in error on the input disk is skipped while the corresponding sector on the output disk is filled with the error pattern “BADBAD...”.

Each sector in error on the output disk is skipped, and the corresponding sector from the input disk is not copied. See paragraph entitled “Unlabelled Open”.

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...”.

See paragraph entitled “Unlabelled Open”.

MTPDSK

Every record on the input tape is moved into a sector on the output disk. The first 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. An RD command is then necessary, followed by an AD command (see next paragraph).

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 RD command is then necessary, followed by an AD command.

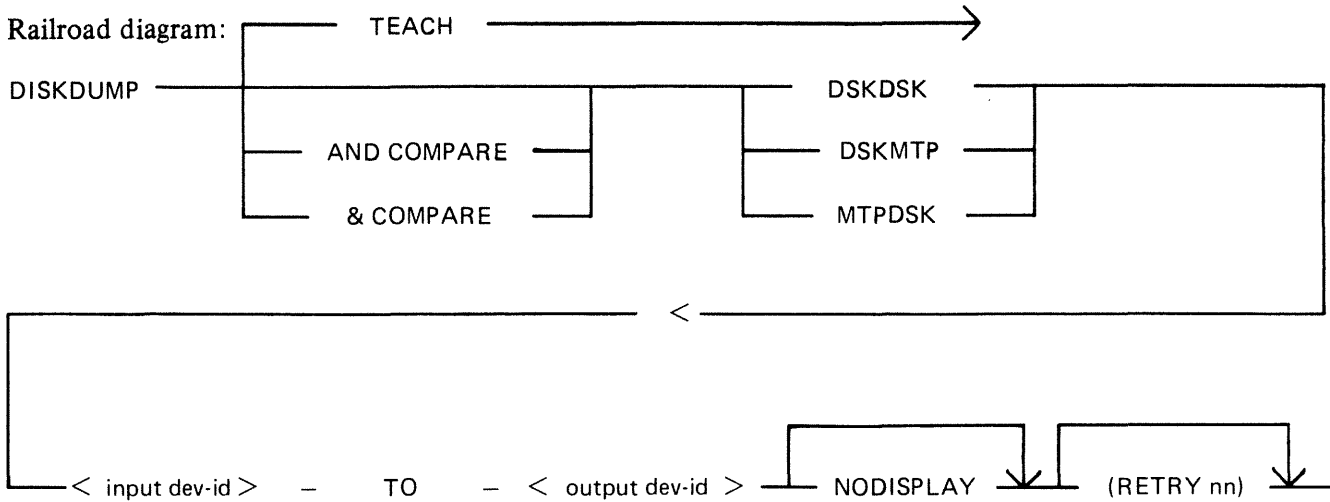
The following messages are displayed in this case:

< 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.

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

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 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 #>).

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.

DP.ANALYZER

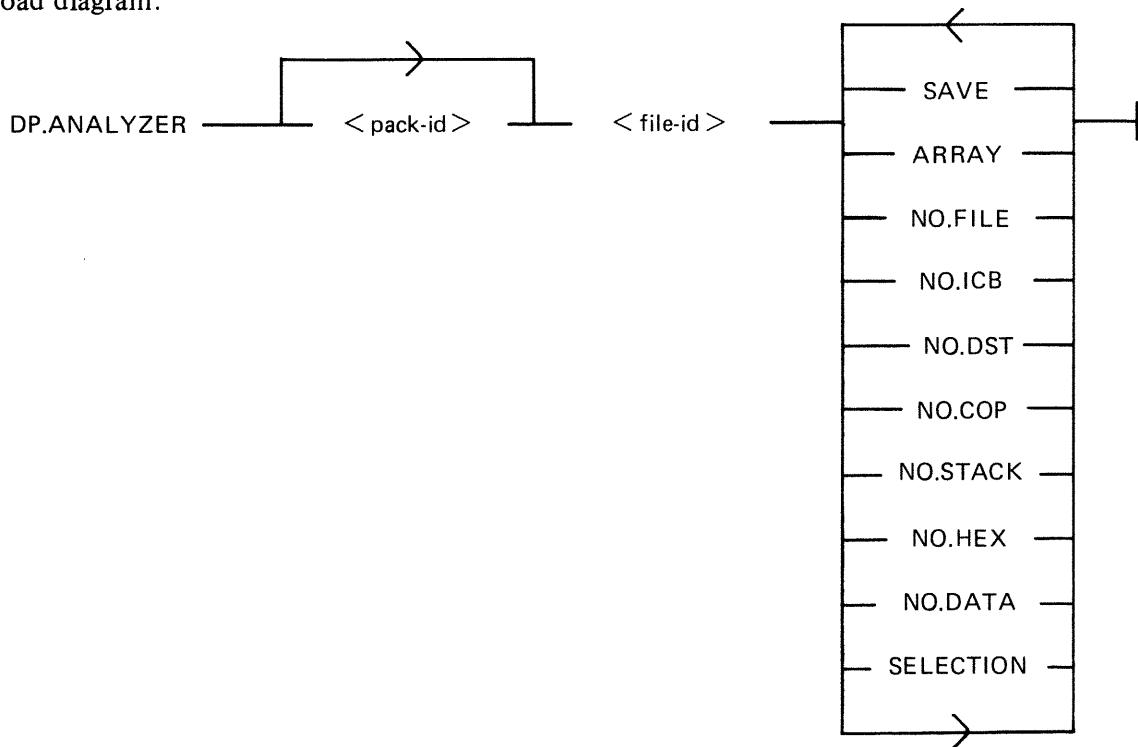
Execution of a program may lead to a DS/DP condition. When a program is DP'ed, a dump file is created on the same disk as the program file (either by the operator or by the COBOL interpreter when a "NO USE PROCEDURE" condition is encountered after an error). The name of the dump file is DMFILnn, where nn is the mix number.

The dump file can be analyzed, in relation to the program file itself, to give a listing containing:

- program parameters
- run parameters
- interface control block
- communicate parameter area
- control stack analysis
- program segment table
- data segment table
- internal file name block
- file information (FPB, FIB, buffers)
- data analysis
- data segments
- current code segment
- locked slice

The listing can be obtained by invoking DP.ANALYZER with the following syntax:

Railroad diagram:



<pack-id> is by default the system disk.

The options, entered in any order, have the following effect:

SAVE	This causes DP.ANALYZER to retain the input dump file rather than remove it after analysis (which is the default).
ARRAY	This is applicable only to COBOL/RPG dump files. It causes all elements of all arrays to be printed. The default is to print a maximum of twenty elements of each array.
NO.FILE	This suppresses printing of any file information.
NO.ICB	This suppresses printing of any ICB information.
NO.DST	This suppresses the DST and PST analysis.
NO.COP	This suppresses the COP Table analysis (only for COBOL programs).
NO.STACK	This suppresses the Data Stack analysis (only for MPL II programs).
NO.HEX	This suppresses printing of the current Code segment and of the Locked Slice.
NO.DATA	This suppresses the printing of the Data segments.
SELECTION	<p>This causes a series of ACCEPT/DISPLAY's to be executed on the SPO, which allow the operator to select sections of the DUMP listing to be printed. Whatever the selection is, the ICB, CPA, PST/DST and IFNB appear at the beginning of the DUMP listing.</p> <p>The options may be specified in any order and also several times. They have precedence over the options in the initiating message. Their syntax is as follows:</p> <p>FIB < FIB.NB > Selects the FIB specified by FIB.NB, which is the decimal number indicated in the INTERNAL FILE NAME BLOCK section of the listing.</p> <p>FIB < Internal File Name > Selects the FIB specified by <Internal File Name>, which is found in the INTERNAL FILE NAME BLOCK section of the listing.</p> <p>SEGMENT < Segment number > Selects the Data Segment indicated by Segment number which is found in the Data Segment Table.</p> <p>Note that when the options mentioned above are invoked, the Dump file is not removed.</p> <p>ALL < init message > All the options that can be entered in the initiating message can be specified after the ALL option and have the same effect.</p> <p>HELP The syntax of the SELECTION option is displayed on the screen.</p> <p>TEACH The syntax of the SELECTION option is displayed on the screen.</p> <p>BYE Sends DP.ANALYZER program to EOJ.</p> <p>END Sends DP.ANALYZER program to EOJ.</p>

Examples

1. DP.ANALYZER DMFIL03 SAVE

DP.ANALYZER expects to find a dump file called DMFIL03 on the system disk, prints a formatted dump listing and retains the dump file.

2. DP.ANALYZER USER/DMFIL02 NO.FILE

DP.ANALYZER expects to find a dump file called DMFIL02 on a disk labelled USER, prints a formatted listing omitting all file information and removes the file USER/DMFIL02 at the end of the job.

3. DP.ANALYZER DMFIL04 ARRAY SAVE

DP.ANALYZER expects to find a dump file called DMFIL04 on the system disk, prints a formatted listing including all the elements of all arrays and retains DMFIL04 at the end of the job.

4. DP.ANALYZER DMFIL05 SELECTION

AX < Mix No > FIB 19	} responses to the prompts
AX < Mix No > SEGMENT 11	
AX < Mix No > FIB SYSMEM	

<mix>/DP.ANALYZER ACPT

DP.ANALYZER expects to find a dump file called DMFIL05 on the system disk, prints a listing consisting of the sections ICB, CPA, PST/DST, IFNB, FIB 19, Data Segment 11, FIB of SYSMEM. DMFIL05 is retained.

NOTE

Sometimes, because of the BIL interpreter, the contents of the end of an MPLII program working stack is unpredictable. In this case, the printing is limited to the analyzable portion.

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
FORTTRAN48.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 alphasized 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
FORTTRAN48	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
FORTTRAN48	130	48-character FORTTRAN
FORTTRAN48.NONSTD	130	48-character FORTTRAN.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

The normal End Of Job message is:

“<train name> = <train number> LOADED ON LP<x>”

Two error messages can be displayed:

“INVALID LT REQUEST”

This indicates that the chain type is invalid, or that the specified chain does not match the printer type.

“LP<x> NOT AVAILABLE FOR LT”

This indicates that the line printer <x> is not on-line.

“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

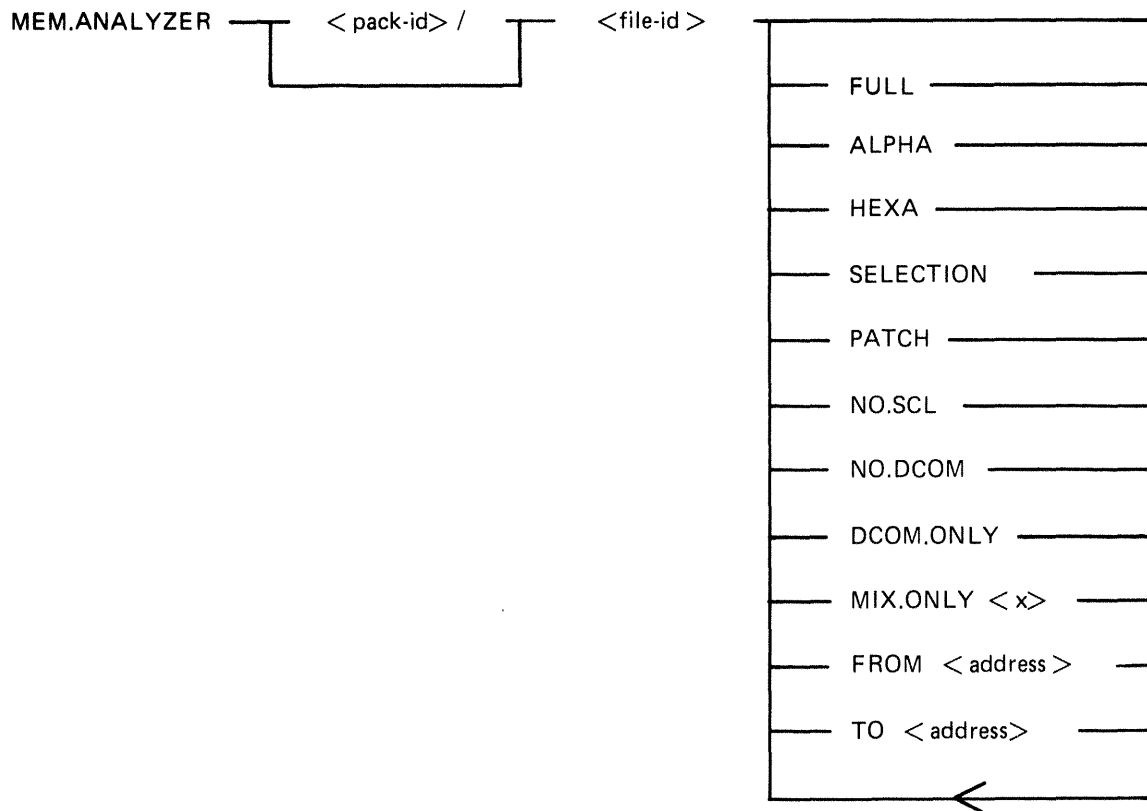
A memory dump file can be obtained by executing the appropriate MEM.DUMP cassette in stand-alone mode (for operating instructions, refer to MEMORY.DUMP in Stand-Alone Utilities later in this section. This causes the contents of the memory to be written into a file named DMFIL00.

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.

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 listing can be obtained by invoking MEM.ANALYZER with the following syntax:

Railroad diagram:



<pack-id>
<file-id>

indicates the name of the device containing the dump file to be analyzed.
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 in one run different portions of memory.
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 indicated by <integer>, where <integer> indicates the Mix number.
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 (register A must be noted manually for a B 1830).
- 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, or is replaced when a new Memory Dump file is created if its name was left as DMFIL00.

Therefore, it is suggested that after creation of DMFIL00, the name of this file is changed to a unique name, it is analyzed as soon as possible by MEM.ANALYZER and is removed as soon as it is certain that the file is no longer needed.

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

Railroad diagram:



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.

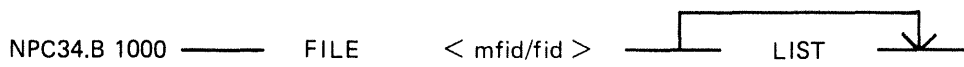
The NPC34.B1000 generates code files to be loaded on DCP-3 or DCP-4.

Use of the NPC34.B1000

The NPC34.B1000 should be run after each successful NDL compilation.

Syntax

Railroad diagram:



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 "33335554444", 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.

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

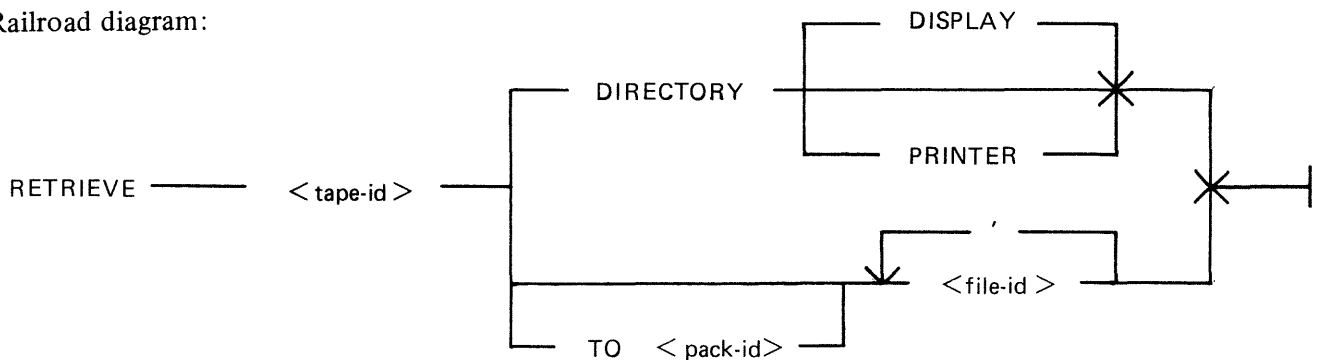
This utility provides the user with a means of copying to disk one or more single files from a tape created by the DISKDUMP program.

Syntax

The following syntax diagram is displayed by entering:

“RETRIEVE”

Railroad diagram:



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).

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
COLDSTART	Loads system software to disk from a COLDSTART magnetic tape.
CLEARSTART	Determines the system configuration, builds tables and places the system under MCP control.
PACK.INIT	Initializes a disk pack in CMS format.
CART.INIT	Initializes a disk cartridge in CMS format.
MEM.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

A utility, CASS.CREATE, has been developed which enables B 1000 series stand-alone cassettes (except B 1825 and B 1830) to be generated under CMS MCP control.

Syntax

```
CASS.CREATE <sau-file-name> TO XXX MTP
```

where <sau-file-name> is one of:

CLEAR.START	for CLEARSTART
COLD.START	for COLDSTART
PACK.INIT	for PACK INITIALIZER
CART.INIT	for CARTRIDGE INITIALIZER
MEMORY.DUMP	for MEMORY DUMP

Operating Instructions

1. Place blank cassette in CTA.
2. Enter "PG CTA".
3. System displays: "CTA PURGED".
4. Enter CASS.CREATE message.
5. System displays:
" <mix>/CASS.CREATE 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

B 1000 System

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.

The following halts are common to all stand-alone utilities except PACKDUMP:

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.

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 3.03.00 release need only initialize a user disk and use "LD LOAD FROM <tape-id> TO <pack-id> ="

Operation

"**** B1800/B1900 CMS COLDSTART ****MARK3.04.00" 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 COLDSTART 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.

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 LIBRARY tape.

“MAG TAPE NOT READY. CORRECT AND START”.

“DISK NOT READY. CORRECT AND START”.

Problem with the indicated device. When corrected, press the START button on the console panel.

“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 D_{xx} COLDSTART ABORTS” (xx = Px or Kx)

“DIRECTORY FULL COLDSTART ABORTS”

“NOT ENOUGH SPACE ON CMS DISK COLDSTART ABORTS”

“NOT A CMS DISK COLDSTART ABORTS”

All these are fatal errors. The disk must be re-initialized 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.

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: @00000000000000100100011@)
 - 3) Press the LOAD button.
3. Set the MODE switch to NORMAL and press the START button.

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.

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 @11111010000000-1000000011@. This enables the MCP to handle only channels 0, 1 and 9. All the other channels are disregarded.

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. 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(PE)
@34@	@97@	MT(NRZ)
@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 INITIALIZERS (CART.INIT/PACK.INIT)

General Information

All disks must be initialized in CMS format before they can be used under MCP control. Each initializer consists of three basic passes:

1. Write addresses for every sector on the disk.
2. Verify all addresses on the disk.
3. Write and verify three data patterns of 2 bytes through the disk: on each cylinder, the utility writes and reads the patterns @6DB6@, @DB6D@, @B6DB@.

For disk pack only, pass 3 consists of three sub-passes:

1. With normal offset.
2. With offset in.
3. With offset out.

In the event of an error, each I/O operation is retried five times and a count is kept of the number of retries (this appears on a KA or LR listing as the field "ERROR COUNT").

If an I/O operation is not successful after five retries, the action taken depends on the type of disk and the operation attempted.

For disk pack an attempt is made to relocate the bad sector (five spare sectors are available per cylinder for this purpose). If the sector cannot be relocated, or five sectors have already been relocated, the track in which the sector lies is completely removed, that is, the bad area size will be either no. of sectors per track if the bad sector is the first one of the track, or (no. of sectors per track + allocation unit) otherwise. Note that the attempts to relocate a bad sector are not displayed.

For disk cartridge, discovery of a bad sector causes the track to be removed.

If any sector in cylinder 0 is discovered to be bad during passes 1 or 2, the initializer terminates.

Error Limits

The disk is regarded as unusable if the number of removed tracks exceeds 25, whatever the device may be.

The disk label and disk directory are built after initialization. The disk directory starts at the first track after track 0 where the total directory (available table + file name list + disk file header) can be located.

Operating Instructions

The operating instructions are similar for both initializers. After the cassette has been loaded, the identification is displayed.

For cartridge:

"CMS DISK CARTRIDGE INITIALIZER MARK 3.04.00"

For pack:

“CMS DISK PACK INITIALIZER MARK 3.04.00”

The operator is then asked to input various parameters about the pack to be initialized. These parameters appear progressively at the top of the screen.

The operator is asked:

“INPUT DEVICE <SPO OR CRD>”

If “SPO” is specified, the initializer operates in an interactive mode. If “CRD” is specified, the input parameters are read from cards, one card for each disk.

Card Format

Column	Description
1	Unit (A thru H)
3-8	Serial number (6 numeric digits)
10-16	Disk identifier (up to 7 alphabetic characters)
21	Disk type (R or blank)
23-25	Maximum number of files (up to 999)
30-37	Date (MM/DD/YY)
40-53	Owner's name (up to 14 characters)

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 enclosed between “? DATA” and “? END” cards (“?” represents any invalid character for an 80 column card).

SPO Input

Parameters are entered interactively. As each field is input, it is checked for validity. If it is invalid, a suitable message is output and the field can be re-entered.

Cartridge

QUESTION: WHICH CARTRIDGE – DK<X>
RESPONSE: DKA, DKB, ... DKH

Pack

QUESTION: WHICH PACK – DP<X>
RESPONSE: DPA, DPB, ... DPH

Both

QUESTION: ENTER 6 DIGIT SERIAL NUMBER
RESPONSE: number in the range 000000 – 999999
QUESTION: ENTER PACK.ID
RESPONSE: up to 7 characters from the set “A” thru “Z”, “O” thru “9”, “.” or “-”
QUESTION: ENTER CARTRIDGE TYPE <R or nothing>
QUESTION: ENTER PACK TYPE <R or nothing>
RESPONSE: R or ETX only

NOTE

R means restricted disk which is not yet implemented.

QUESTION: MAX NB OF FILES (UP TO 2805)

RESPONSE: any decimal number between 1 and 2805

NOTE

The default value is 254 for cartridge and 2805 for pack.

Cartridge:

When 256 files are requested, 8 sectors are allocated for the Available Table. For every additional 200 files declared, one sector is added until the maximum size of 14 sectors is reached.

Pack:

The size of the Available Table is given by the following algorithm:

$$(\text{number of files requested} / 32) + 3.$$

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 File 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).

2803 files will give:

2803 sectors for Headers
255 sectors for File Name List
91 sectors for Available Table $(2803/32)+3$

3149 sectors for the whole directory.

The Available Table is extended to 94 sectors to have a total directory length of 3152 sectors (the next integer multiple of 8 sectors).

It must also be remembered that one sector of disk file header is reserved for each file requested.

QUESTION: ENTER DATE – <MM/DD/YY>

RESPONSE: MM/DD/YY

QUESTION: ENTER OWNER'S NAME (UP TO 14 CH)

RESPONSE: up to 14 characters

Initialization now starts. While it is executing, pressing the XMT button displays information about the processing:

-
1. For pass 1 or 2 the message identifies the pass and the current cylinder number.
 2. For pass 3 the message identifies whether a write or verify operation is in process, the pattern used and the current cylinder.

If the message "DPEC ATTENTION" is displayed, it indicates that an extended result descriptor cannot be cleared from the DPEC. Press START twice on the processor panel.

NOTE

This can happen only with packs.

When initializing a cartridge, after completing the last pass, initialization/verification, the message:

"ENTER FINAL PATTERN (4 DIGITS) OR <ETX>"

is displayed. This enables a user determined pattern to be used. If this extra pass is not desired, enter <ETX>.

At the end of initialization, the following information is displayed:

DRIVE
SERIAL NUMBER
PACK-ID
NUMBER OF REMOVED TRACKS
NUMBER OF COUNT ERRORS (NUMBER OF RETRIES)
NUMBER OF RELOCATED SECTORS (PACK ONLY)

If tracks were removed or sectors relocated, the address of each removed track and relocated sector is given. Finally, the message "DP<X> INITIALIZED" or "DK<X> INITIALIZED" is displayed.

It is then possible to re-initialize another disk (if SPO input was selected) by pressing the START button on the console panel.

Time Needed

An estimation of the time needed to initialize packs follows:

35 minutes for a 205
45 minutes for a 206
135 minutes for a 207
70 minutes for a 225

Error Messages

All the error messages are self explanatory and related to a bad syntax detection.

End Of Job Messages

When the program has run successfully, the parameters entered by the user about the pack to be initialized appear on the screen. If sectors have been relocated, their location is displayed.

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).

MEM.DUMP Cassette

Two cassettes are available, one for B 1830 systems and the other for B 1860 systems. This 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 switch or, if this is not successful, with the HALT button.

The operating instructions to run the cassette are as follows:

1. If running on a B 1825/30, note the value of the A register and press the CLEAR button. On all other systems, do not press the CLEAR button.
2. Mount the appropriate MEM.DUMP cassette.
3. Switch to TAPE mode (with B 1825/30 enter ST).
4. If running on a dual processor, disable the slave processor.
5. Press START.
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. (Tregister contains the release level).
8. Rewind the cassette.
9. Perform CLEARSTART.
10. Print DMFIL00 by executing MEM.ANALYZER.

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 (CR 24 on the B 1830).
3. Set the SINGLE MIC/CONT (SINGLE MICRO/NORMAL) switch to SINGLE MIC (SINGLE MICRO).
4. Press the START button a few times, noting the values, 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.

As the A register points to the next micro-instruction address, this is of great help in debugging.

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 DMFIL00 is directed.

@Dc000d@	Dump is complete.
@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. The dump cannot be resumed. Note : in order to regain space already allocated for the dump file, a SQUASH must be performed.

@Fc003d@ Not enough space on the designated disk. In the L register, enter the parameters of the drive on which the file DMFIL00 is to be created (see below).

@Fc004d@ An error occurred while reading or writing the dump file to disk. Pressing START causes 10 retries to be performed. To direct the dump to another disk, enter in the L register the parameters of the drive on which the file DMFIL00 is to be created (see below).

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 DMFIL00.

@FF0000@	A cassette parity error occurred.
@AAAAAA@	Switch to RUN mode and press START to continue. Information given at the time of the halt is: <ul style="list-style-type: none">- 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.

NOTE

If the system pack was seeking when the interrupt occurred, the dump file creation program can enter an infinite loop. To resume processing, press the HALT then CLEAR then START buttons.

Known Problem

Sometimes the dump cannot be taken, even after CLEAR and START. In this case, the cassette must be reloaded but all status, registers, TAS and Scratchpads are lost.

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. Except for the halts listed as recoverable, a dump must be taken and an FCF must be raised.

The system cannot be restarted, unless specified.

CLEARSTART Halts

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 = @22@ or SYSCONFIG machine type not in range @51@ - @60@
C00008	TRACE not found (internal, press START to resume).
C00009	MCP not found.
C0000A	MCP file type = @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.
C00015	Table error.
Cx00FF	An I/O interrupt has been waited from a channel for 25.6 sec. The value of x is the Channel number. Press START to resume, unless status is not good.

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).
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 STC while in disk Test.Op.
F00021	Invalid STC 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.

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	SYSTEM 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 entry 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	SYSTEM 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	SYSTEM 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.
80FFFF	72 files are already open, corresponding to the maximum allowed.

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.

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.

(continued)

xx	
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

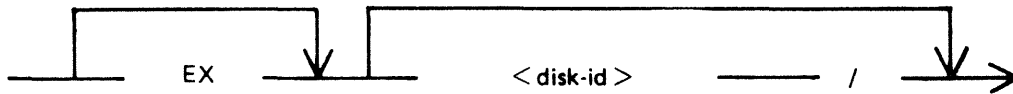
APPENDIX A

COMPLETE INPUT MESSAGE DIAGRAMS

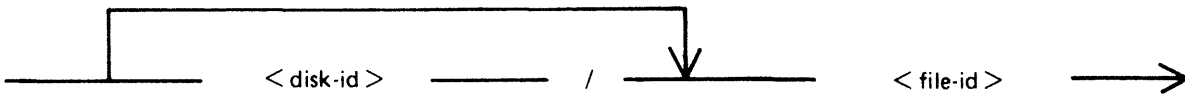
This appendix gives the diagrams for all the CMS-common intrinsics and utilities, including SORT and CO, in alphabetical order. These diagrams give the complete input message formats, for ease of reference.

For details of the meaning of these messages, refer to text.

In the following diagrams, the <ex-option> is defined as:

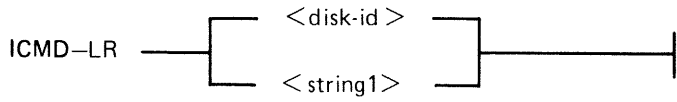


and <file-name> is defined as:

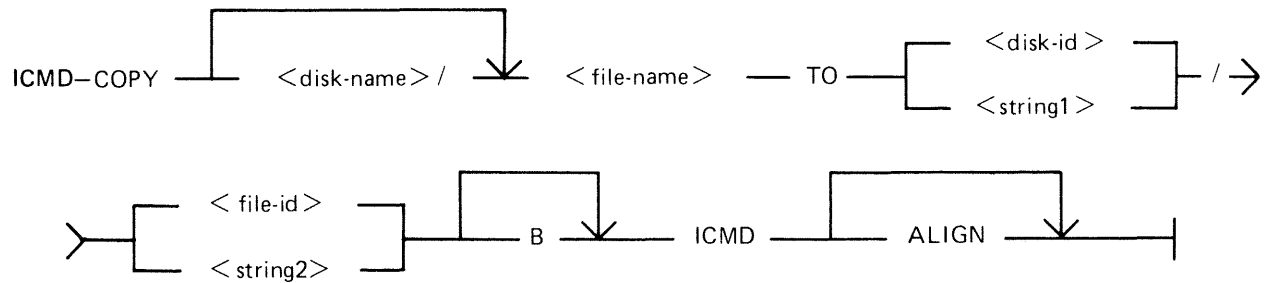


ICMD Utility

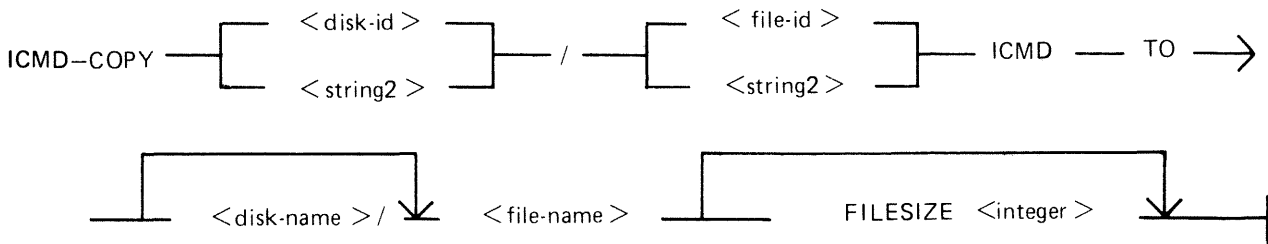
Version 1



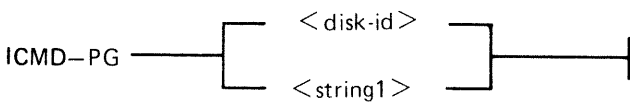
Version 2



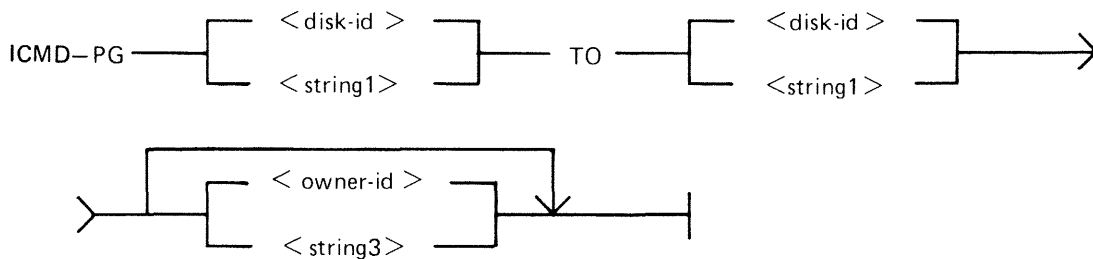
Version 3



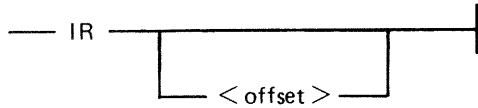
Version 4



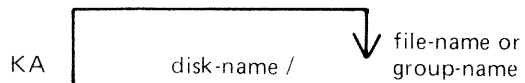
Version 5



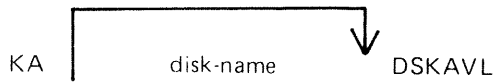
IR Utility



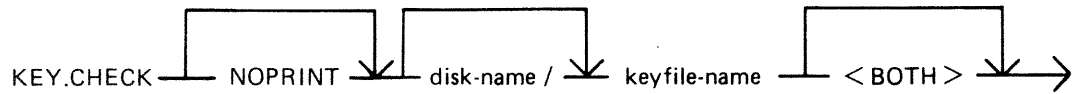
KA Utility



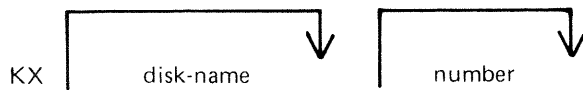
or



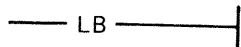
KEY.CHECK Utility



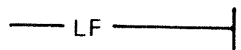
KX Utility



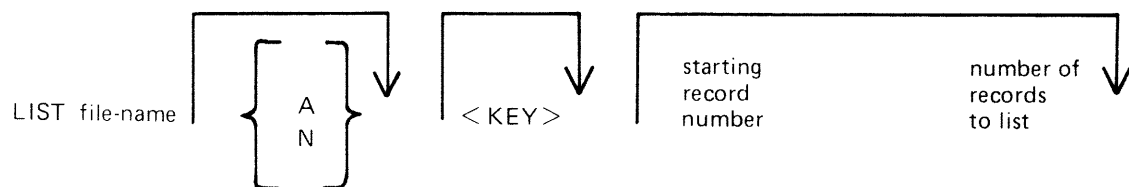
LB Utility



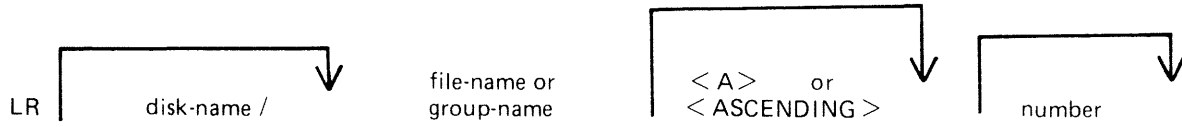
LF Utility



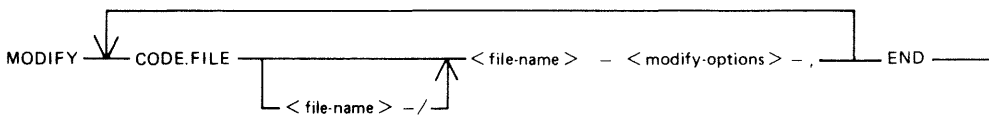
LIST Utility



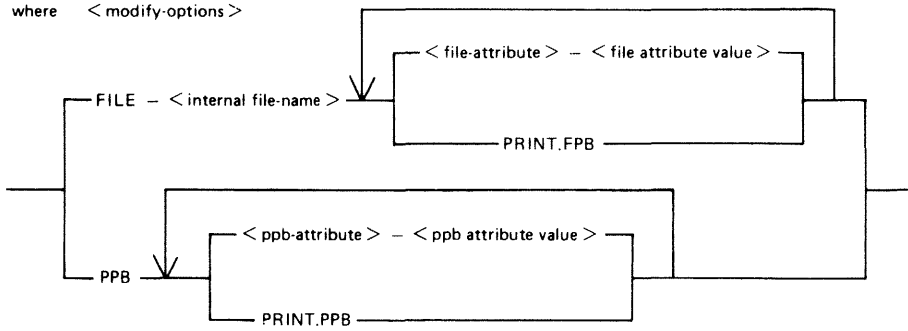
LR Utility



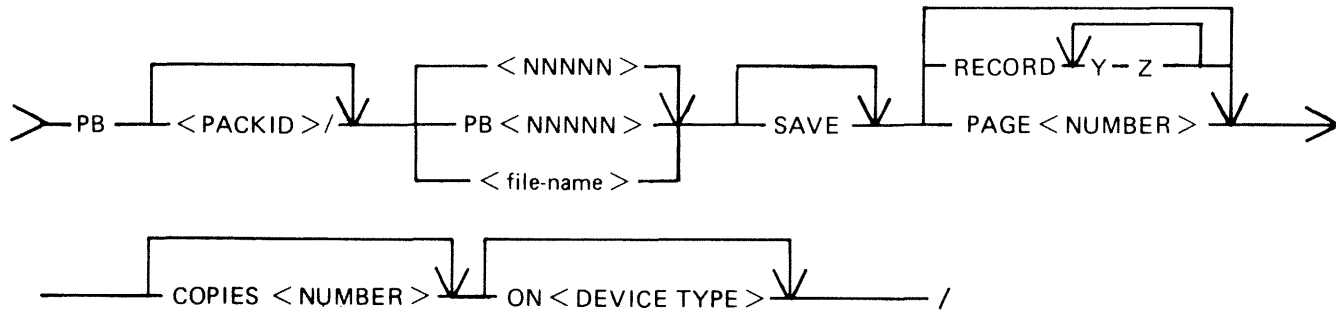
MODIFY Utility



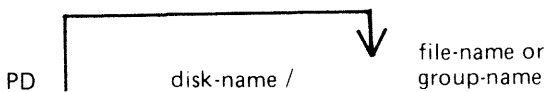
where <modify-options>



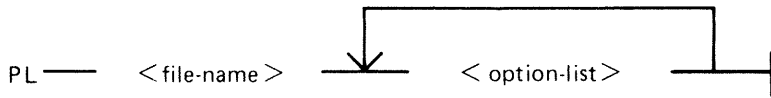
PB Utility



PD Utility



PL Utility



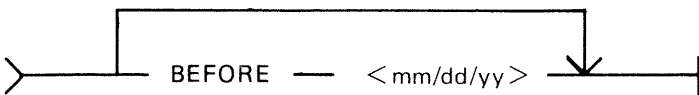
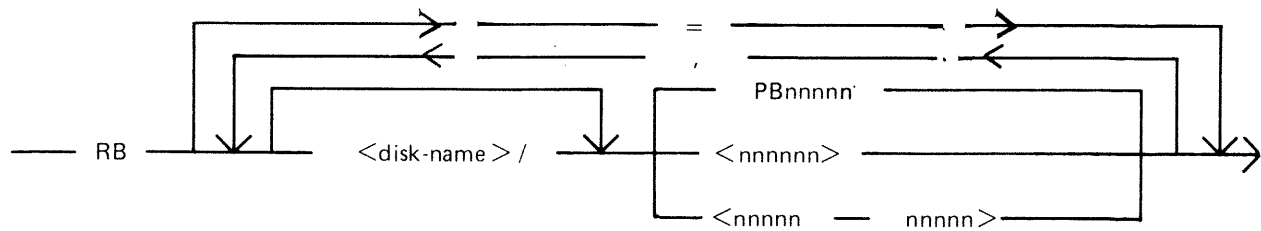
where <option list> = SYSTEM or S

- ERROR or E
- STATS
- DISPLAY
- BACKUP
- FROM <time spec>
- TO <time spec>
- ENTRY <number>
- MIX <number>
- INPUT
- OUTPUT
- <empty>
- DEVICE <device spec>
- PURGE

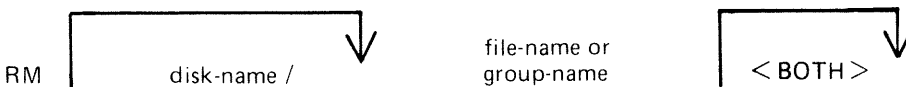
PPID Utility



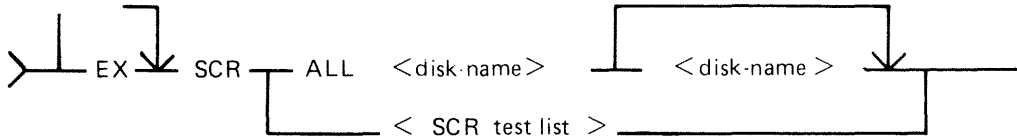
RB Utility



RM Utility

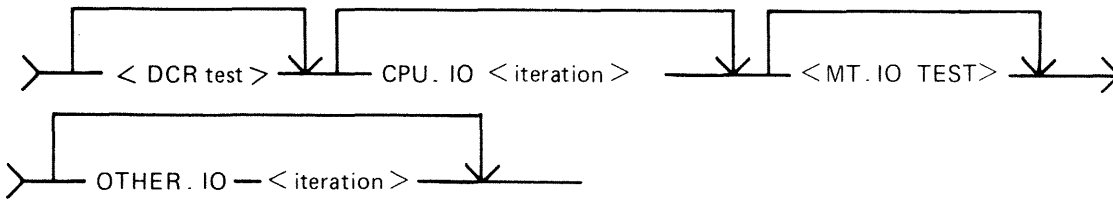


SCR Utility

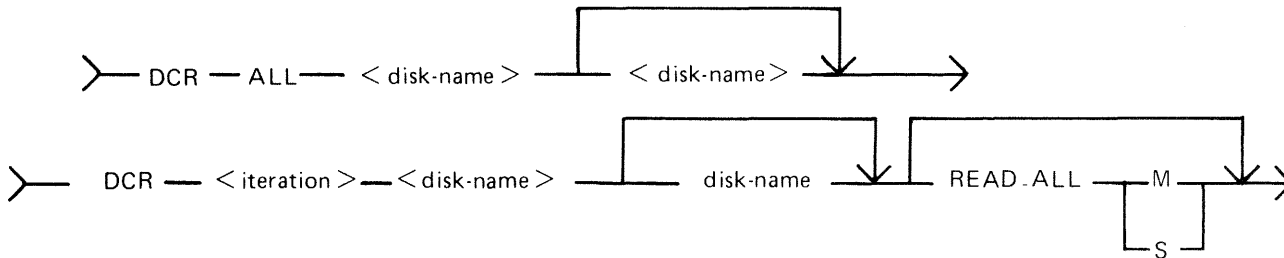


iteration = the number of iterations of the subsection.

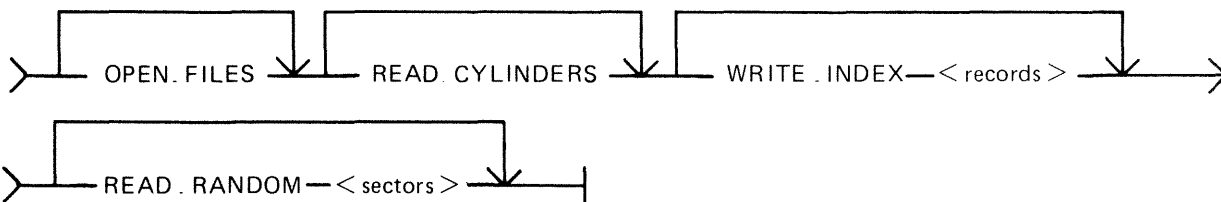
Where SCR test list is:



Where DCR test is:

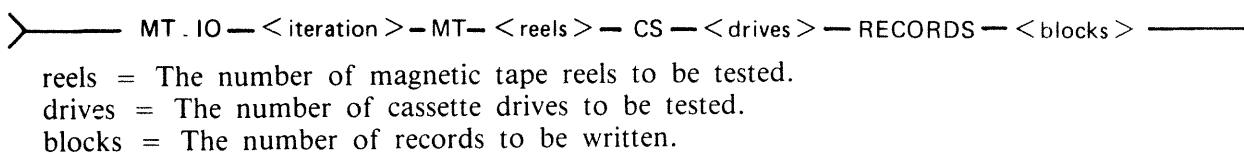


or



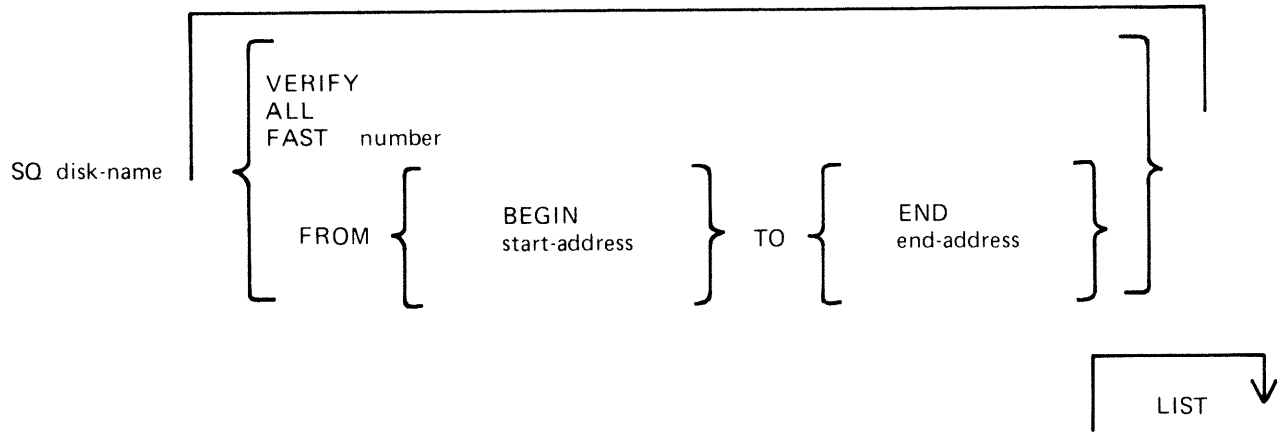
records = The number of (data) records in the index file.
sectors = The number of sectors to be read randomly.

Where MT.IO test is:

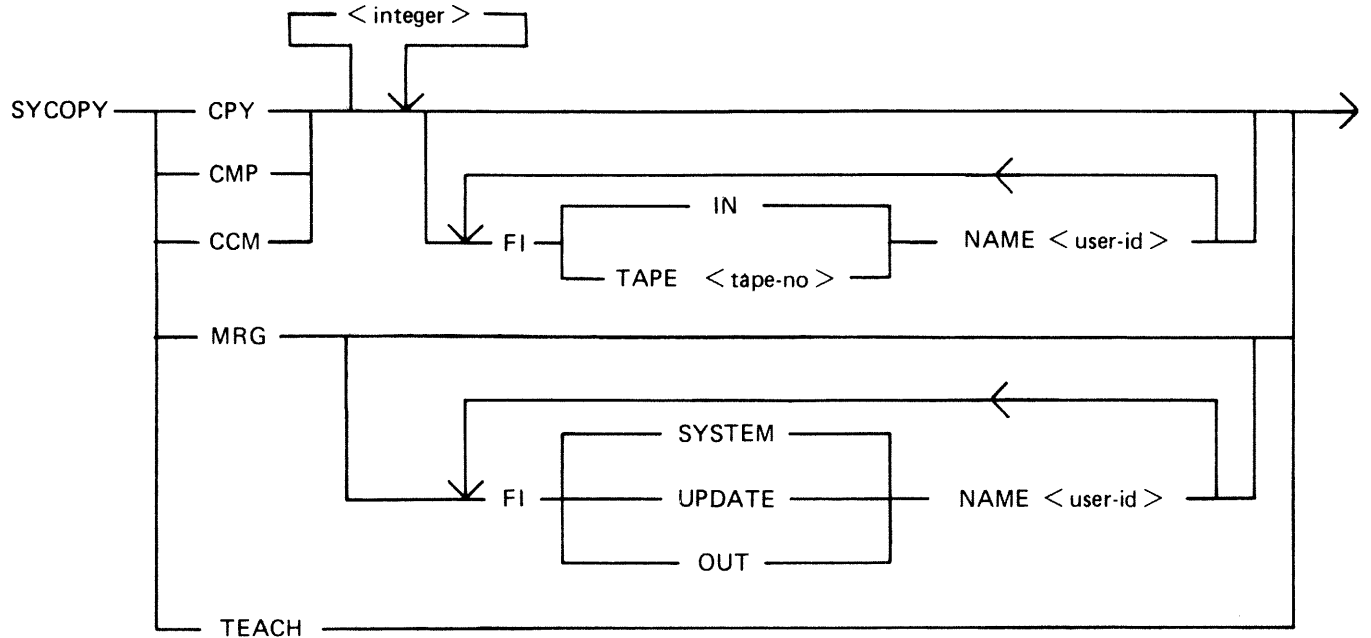


reels = The number of magnetic tape reels to be tested.
drives = The number of cassette drives to be tested.
blocks = The number of records to be written.

SQ Utility



SYCOPY Utility



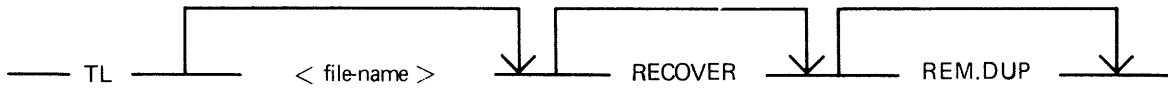
TAPELR Utility

TAPELR library-tape-name

TAPEPD Utility

TAPEPD library-tape-name

TL Utility

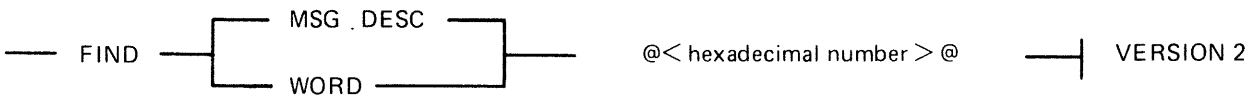
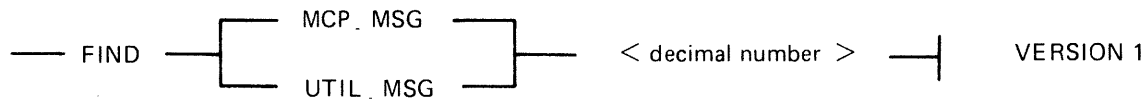


TRANSLATOR Utility

BYE

BYE

FIND

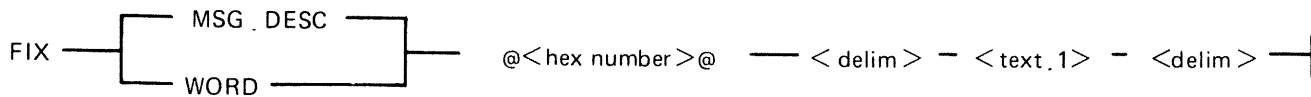


FIX

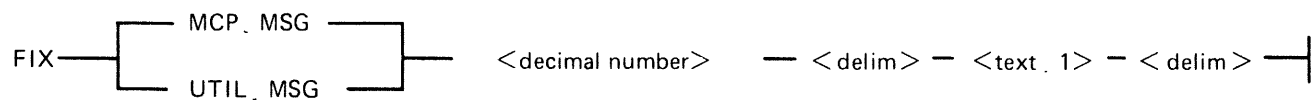
Version 1



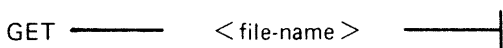
Version 2



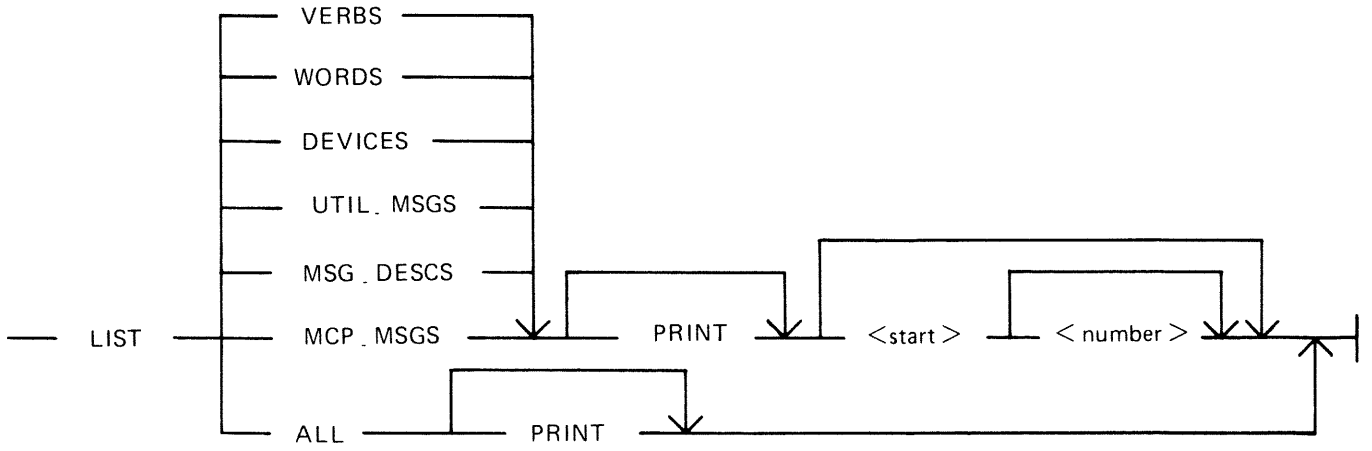
Version 3



GET



LIST



NEXT



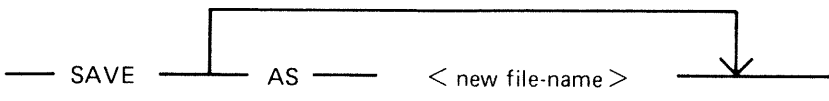
RECOVER



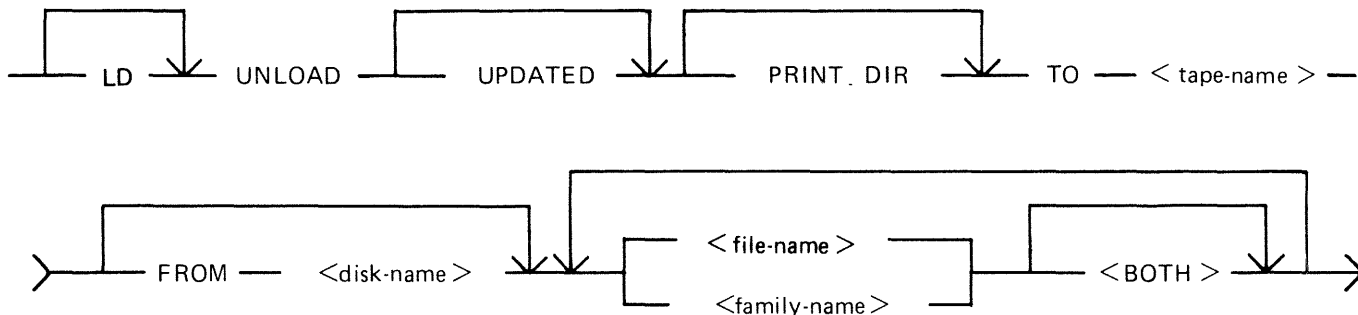
REMOVE



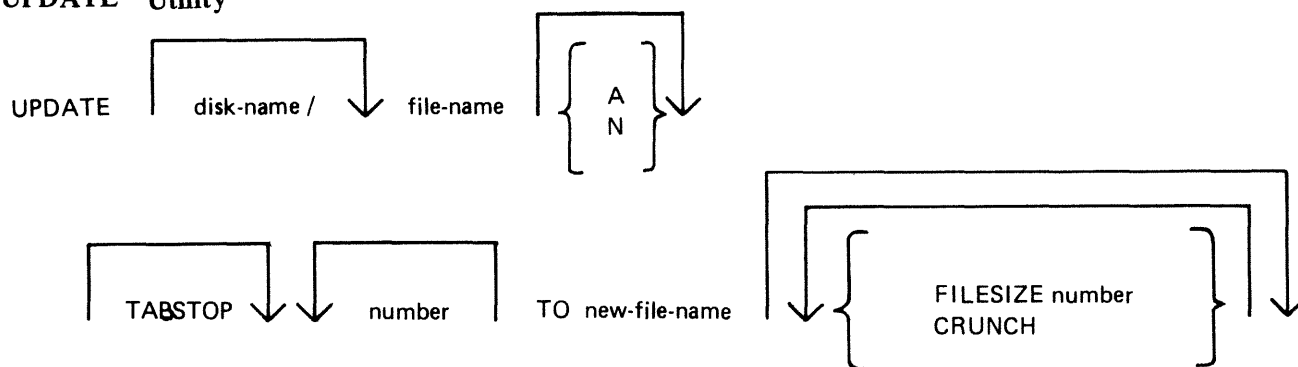
SAVE



UNLOAD Utility



UPDATE Utility



WL Utility

WL

XD Utility

XD disk-name address length

APPENDIX B

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
I	30		<u>CHECK.DISK SOGDOC1</u>
O	7	07/CHECK.DISK <2>	SOGDOC1/SYSMEM DMA INPUT PARITY ERROR WHILE IN REA
		D DISK ADDRESS = 001203	
O	7	ERROR NOTIFIED ON READING DISK SOGDOC1	
O	7	SECTOR 4611/00012030 NOT DENOTED BAD IN DIRECTORY	
O	7	CHECK.DISK ON SOGDOC1 COMPLETED -	
O	7	ONE CONSISTENT ERROR NOTIFIED	
O	7	1 BAD SECTOR<S> NOT DENOTED IN DIRECTORY	

COMPARE

COMPARE TEST1 WITH MYDISK/TEST1

TEST1 WITH MYDISK/TEST1 COMPARED - 1 ERROR

```
DIFFERENCE(S) FOUND FROM BYTES @001F@ AND @001F@

RECORD.      3 OF FILE TEST1          COMPLETE RECORD -
@0000@ 5448 4953 2049 5320 5245 434F 5244 2054 4852 4545 2033 3333 3333 3333 3320"THIS IS RECORD THREE 3333333333 "
@0020@ 4552 5252 4F52 2033 3333 3333 3333 3333 3333 3333 3333 3333 3333 3333 3333"ERROR 3333333333333333333333333333"
@0040@ 3333 3333 3333 3333 3030 3030 3033 3030                                     "3333333300000300"

RECORD.      3 OF FILE MYDISK /TEST1    DIFFERENCE(S) -
@0000@ ..... ..33".....3"
@0020@ 3333 3333 3333 33.. ..... "3333333....."
NO DIFFERENCES IN SECTION(S) NOT PRINTED
```

DA

I/O	MIX NO.	MESSAGE	TEXT
I	30	<u>DA MYDISK PRINTER</u>	
O	15	15/DA PRINTER BACKUP FILE SYSTEM/PB00018	OPENED
O	15	DA. MINI DISK MYDISK	
O	15	15/DA ACPT	
I	30	<u>AX 15 DCL</u>	
O	15	DA. ILLEGAL PARAMETER DCL	
O	15	15/DA ACPT	
I	30	<u>AX 15 LABEL</u>	
O	15	15/DA ACPT	
I	30	<u>AX 15 DFH TEST1</u>	
O	15	DA. ILLEGAL PARAMETER TEST1	
O	15	15/DA ACPT	
I	30	<u>AX 15 DFH OF TEST1</u>	
O	15	15/DA ACPT	
I	30	<u>AX 15 DFH NEXT</u>	
O	15	15/DA ACPT	
I	30	<u>AX 15 AVAIL.TABLE AVAIL</u>	
O	15	15/DA ACPT	
I	30	<u>AX 15 NAME.LIST NAMES</u>	
O	15	15/DA ACPT	
I	30	<u>AX 15 READ @0140E@</u>	
O	15	15/DA ACPT	
I	30	<u>AX 15 END</u>	
O	15	DA. TERMINATED	
O	15	15/DA PRINTER BACKUP FILE SYSTEM/PB00018	CLOSED

```

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@    CHS/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      @544553543120202020202020@

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

```


9 COBOL3

@434F424F4C33202020202020@

@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@:	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	
	32/@020@:	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF		
	64/@040@:	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF		
	96/@060@:	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF		
	128/@080@:	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF		
	160/@0A0@:	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF	FFF		

DSKUTL BOOT

I/O	MIX	MESSAGE	TEXT
	NO.		
I	30	<u>DSKUTL BOOT DISK2 <COLDSTART></u>	
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	<u>DSKUTL IN</u>	
O	11	DSKUTL <IN> - ENTER DISK NAME <UP TO 7 LEGAL CHARACTERS>	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 DISK2</u>	
O	11	DSKUTL <IN> - ENTER NO. OF FILES <NUMBER LESS THAN 2805>	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 99</u>	
O	11	DSKUTL <IN> - ENTER SERIAL NO. <6 DIGITS>	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 304002</u>	
O	11	DSKUTL <IN> - ENTER OWNERS NAME <UP TO 14 CHARACTERS. DELIMIT WITH " >	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 "BURROUGHS CORP"</u>	
O	11	DSKUTL <IN> - IS THIS DISK FOR MTR USE? - <Y OR N>	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 Y</u>	
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	<u>AX 11</u>	
O	11	DSKUTL <IN> - ENTER NO. OF TEST PATTERNS TO BE SPECIFIED <NUMBER 0-4>	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 1</u>	
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	<u>AX 11 @FF@</u>	
O	11	READY AND RESERVE REQUIRED DRIVE FOR DISK INITIALISE	
O	11	11/DSKUTL <28> WAITING INITIALISE DK DEVICE ASSIGNMENT REQUIRED	
I	30	<u>RD DMA</u>	
O	30	DMA RESERVED 0 FILES OPEN	
I	30	<u>AD 11 DMA</u>	
O	11	DMA IS CMS DISK DISK1 - ENTER OK TO CONFIRM <IN> REQUIRED	
O	11	11/DSKUTL ACPT	
I	30	<u>AX 11 OK</u>	
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	0 SECTOR(S) ARE NOT AVAILABLE ON DMA DISK DISK2	
O	11	DSKUTL <IN> OF DMA DISK <5632> AS DISK2 ON B90 COMPLETED	
I	30	<u>RY DMA</u>	
O	31	DMA DISK2/ 0 FILES OPEN	

DSKUTL LIST

I/O	MIX NO.	MESSAGE	TEXT
I	30	DSKUTL LIST 0 3	
O	13	PRESENT DISK FOR LISTING PLEASE	
O	13	13/DSKUTL <28> WAITING LIST DK DEVICE ASSIGNMENT REQUIRED	
I	30	RD DMA	
O	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????????		0000000000425552524F5547485320434F52500000000000000000000000000000		
	????????????	??????????	@@???	0000000000000000000000000000000040E5D6D3F2F8F2F2F2F4404040C2C4E220		
	?X? ???)???	?d????3????????		005802200A000029090000200064010000330000000000F1F0F0F0F0F0F0F0		
	????????????	????????????		F0F0F0F00000000100		
	????????????	????????????		00		
	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/@000001@	????????????	????????????		FF		
	????????????	????????????		FF		
	????????????	????????????		FF		
	????????????	????????????		FF		
	????????????	????????????		FF		
	????????????	????????????		FF		
	????????????	????????????		FF		
2/@000002@	8?:?!????^d????1??	?????Q???????		38013A000C21EA1200005E640B1D00943185197D0212110051EFF3CB198A012E		
	?????Q????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 @000000@	32 @000020@	ASSIGNED	*RESERVED			
32 @000020@	10 @00000A@	ASSIGNED	*AVAIL.TABLE			
42 @00002A@	10 @00000A@	ASSIGNED	*FILE DIREC.			
52 @000034@	100 @000064@	ASSIGNED	*FILE HEADRS			
152 @000098@	4 @000004@	ASSIGNED	TEST1			
156 @00009C@	351 @00015F@	ASSIGNED	COBOL1			
507 @0001FB@	98 @000062@	ASSIGNED	COBOL2			
605 @00025D@	223 @0000DF@	ASSIGNED	COBOLXREF			
605 @00025D@	223 @0000DF@	ASSIGNED	COBOLXREF			
828 @00033C@	64 @000040@	ASSIGNED	COBOLINTX			
892 @00037C@	256 @000100@	ASSIGNED	COBOL.NOTES			
1148 @00047C@	256 @000100@	ASSIGNED	COBOL.NOTES			
1404 @00057C@	176 @0000B0@	ASSIGNED	COBOL.NOTES			
1580 @00062C@	110 @00006E@	ASSIGNED	COBOL3			
1690 @00069A@	24 @000018@	ASSIGNED	XA			
1714 @0006B2@	50 @000032@	ASSIGNED	AMEND			
1764 @0006E4@	2488 @0009B3@	AVAILABLE				
4252 @00109C@	331 @00014B@	ASSIGNED	COBOL4			
4583 @0011E7@	1049 @000419@	AVAILABLE				

LD & CHECKADUMP

I/O	MIX NO.	MESSAGE	TEXT
0	31	MTA UNLABELLED	
I	30	<u>SN MTA 00001</u>	
0	31	MTA PURGED 00001	
I	30	<u>DUMP & CHECK PRINT.DIR TO DEMO C=,MPL=,UTIL=</u>	
0	28	<u>EX 000000/CMSCANDE FILE02</u>	CB02 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	<u>RY MTA</u>	
0	31	MTA DEMO/ 002 00002	
I	30	<u>SN MTA 00002</u>	
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
DIRECTORY OF PE TAPE DEMO

TAPE SERIAL NUMBER 00002
DUMPED ON MON 9 AUG 82 AT 13:17:59.3

PAGE 1

FILE NUMBER	FILE NAME	ACTUAL SIZE	MAXIMUM SIZE	RECORD SIZE	RECS/BLOCK	CREATED	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@

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 CN 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/WARNSTART DISK BOOTSTRAP VERSION 03.04.03												
FILE NAME	ACTUAL SIZE	MAXIMUM SIZE	REC. SIZE	RECS/ BLOCK	CREATED	UPDATED	GEN. NO.	FILE TYPE	VERSION NO.	AREAS	AREA LOCATIONS	AREA SIZES
*RESERVED	32	32	180	32	82147	82218	0	SYSTEM @20@		1	0 @000000@	32 @000020@
*AVAIL.TABLE	10	10	180	32	82147	82218	0	SYSTEM @20@		1	32 @000020@	10 @00000A@
*FILE DIREC.	10	10	180	32	82147	82218	0	SYSTEM @20@		1	42 @00002A@	10 @00000A@
*FILE HEADRS	100	100	180	32	82147	82218	0	SYSTEM @20@		1	52 @000034@	100 @000064@
AMEND	50	50	180	1	81058	82218 U	0	C810227 @10@	03.04.00	1	1714 @000682@	50 @000032@
COBOL.NOTES	1542	1542	80	9	82174	82217 U	0	DATA @00@		3	892 @00037C@	256 @000100@
											1148 @00047C@	256 @000100@
											1404 @00057C@	176 @000080@
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 @000032@
COBOL3	110	110	180	1	82096	82217 U	0	C820406 @10@	03.04.05	1	1580 @00062C@	110 @00006E@
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 @00033C@	64 @000040@
COBOLXREF	223	223	180	1	81197	82217 U	0	C810714 @10@	-----	1	605 @00025D@	223 @0000DF@
TEST1	5	5	80	9	82216	82218 U	0	DATA @00@		1	152 @000098@	4 @000004@
XA	24	24	180	1	81058	82218 U	0	C810227 @10@	03.04.00	1	1690 @00069A@	24 @000018@

LR MYDISK/= 100

FRI 6 AUG 82 AT 08:59:39.3 DIRECTORY OF MINI DISK MYDISK SERIAL NO. 000003 OWNER JOHN PAGE 1. INITIALIZED 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												
FILE NAME	ACTUAL SIZE	MAXIMUM SIZE	REC. SIZE	RECS/ BLOCK	CREATED	UPDATED	GEN. NO.	FILE TYPE	VERSION NO.	NO. AREAS	AREA LOCATIONS	AREA 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 @0000BF@
COBOL.NOTES	1542	1542	80	9	82174	82217	U	0 DATA @00@		3	892 @00037C@ 1148 @00047C@ 1404 @00057C@	256 @000100@ 256 @000100@ 176 @0000E0@
COBOL3	110	110	180	1	82096	82217	U	0 C820406 @10@	03.04.05	1	1530 @00062C@	110 @00006E@

AREA ADDRESSES	AREA SIZES	AREA ADDRESSES	AREA SIZES	AREA ADDRESSES	AREA SIZES
AVAILABLE AREAS					
1764 @0006E4@	2488 @0009B9@	4583 @0011E7@	1049 @000419@		
TOTAL AVAILABLE SPACE ON DISK			3537 @000DD1@		

AREA ADDRESSES	AREA SIZES	AREA ADDRESSES	AREA SIZES	AREA ADDRESSES	AREA SIZES
TEMPORARY AREAS					
TOTAL SPACE ON DISK IN TEMPORARY USE			0 @000000@		

MODIFY

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-HLD22505

TIME	DATE	I/O	MIX	ENTRY	MESSAGE	TEXT
HH:MM:SS	MM/DD/YY		NO.	NUMBER		
09:46:33	08/13/82	0	11	467	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:46:34	08/13/82	0	11	468	11/CONFIGURER ACPT	
09:46:36	08/13/82	0	31	469	DMA COBIDX/ 0 FILES OPEN	
09:46:39	08/13/82	I	30	470	AX 11 N	
09:46:40	08/13/82	0	11	471	CURRENT SIZE OF PRIMARY LOG FILES IS 100 SECTORS	
09:46:40	08/13/82	0	11	472	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:46:40	08/13/82	0	11	473	11/CONFIGURER ACPT	
09:46:57	08/13/82	I	30	474	AX 11 N	
09:46:58	08/13/82	0	11	475	CURRENT SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IS 300 SECTORS	
09:47:00	08/13/82	0	11	476	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:47:01	08/13/82	0	11	477	11/CONFIGURER ACPT	
09:47:07	08/13/82	I	30	478	AX 11 Y	
09:47:08	08/13/82	0	11	479	ENTER SIZE OF SECONDARY SYSTEM MESSAGE FILE(S) IN SECTORS	
09:47:10	08/13/82	0	11	480	<MINIMUM 32, MAXIMUM 262140>	
09:47:10	08/13/82	0	11	481	11/CONFIGURER ACPT	
09:47:14	08/13/82	I	30	482	AX 11 32	
09:47:16	08/13/82	0	11	483	CURRENT SIZE OF SECONDARY MAINTENANCE ENTRY FILE(S) IS 32 SECTORS	
09:47:18	08/13/82	0	11	484	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:47:18	08/13/82	0	11	485	11/CONFIGURER ACPT	
09:47:22	08/13/82	I	30	486	AX 11 N	
09:47:23	08/13/82	0	11	487	CURRENT BACKUP DISK FOR LOG FILES IS SYSTEM DISK	
09:47:24	08/13/82	0	11	488	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:47:24	08/13/82	0	11	489	11/CONFIGURER ACPT	
09:47:34	08/13/82	I	30	490	AX 11 N	
09:47:35	08/13/82	0	11	491	NO TEXT WILL BE ZIPPED AT MARKSTART	
09:47:36	08/13/82	0	11	492	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:47:36	08/13/82	0	11	493	11/CONFIGURER ACPT	
09:47:44	08/13/82	I	30	494	AX 11 N	
09:47:45	08/13/82	0	11	495	THERE IS NO POWER OFF MESSAGE	
09:47:46	08/13/82	0	11	496	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:47:47	08/13/82	0	11	497	11/CONFIGURER ACPT	
09:47:52	08/13/82	I	30	498	AX 11 N	
09:47:53	08/13/82	0	11	499	CURRENT ID OF MESSAGE FILE IS SYSLANGUAGE	
09:47:54	08/13/82	0	11	500	DO YOU WANT TO CHANGE IT? - <Y OR N>	
09:47:54	08/13/82	0	11	501	11/CONFIGURER ACPT	
09:47:57	08/13/82	I	30	502	AX 11 N	
09:47:57	08/13/82	I	30	502	AX 11 N	
09:47:58	08/13/82	0	11	503	DO YOU WANT TO CHANGE ANY PRINTER BACKUP OPTIONS? - <Y OR N>	
09:47:59	08/13/82	0	11	504	11/CONFIGURER ACPT	
09:48:04	08/13/82	I	30	505	AX 11 N	
09:48:05	08/13/82	0	11	506	DO YOU WANT TO CHANGE ANY SPO OPTIONS? - <Y OR N>	
09:48:05	08/13/82	0	11	507	11/CONFIGURER ACPT	
09:48:10	08/13/82	I	30	508	AX 11 N	
09:48:10	08/13/82	0	11	509	DO YOU WANT TO CHANGE ANY TRANSLATION TABLES? - <Y OR N>	
09:48:11	08/13/82	0	11	510	11/CONFIGURER ACPT	
09:48:20	08/13/82	I	30	511	AX 11 N	
09:48:30	08/13/82	0	11	512	JOB COMPLETED	
09:48:36	08/13/82	I	30	513	WL	
09:48:57	08/13/82	0	12	514	FILE NAME FILE STATE	
09:48:57	08/13/82	0	12	515	SYS-LOG-01 READY TO TRANSFER	
09:48:57	08/13/82	0	12	516	SYS-LOG-02 ACTIVE	
09:48:58	08/13/82	0	12	517	SYS-LOG-03 NEXT ACTIVE	
09:48:58	08/13/82	0	12	518	SYS-LOG-04 NOT USED	
09:48:58	08/13/82	0	12	519	BACKUP FILE(S) ON FA	
09:48:59	08/13/82	0	12	520	SYS-HLG22300 ACT. 19 RECS.	
09:48:59	08/13/82	0	12	521	MAX. 128 RECS.	

FRI 13 AUG 82 AT 11:39:31

CMS LOG FILE PRINTOUT OF SYS-HLD22505 FROM B90 SYSTEM
STATUS OF FILE IS TRANSFERRED

PAGE 2

TIME HH:MM:SS	DATE MM/DD/YY	I/O	MIX NO.	ENTRY NUMBER	MESSAGE	TEXT
09:49:00	08/13/82	0	12	522	SYS-HLD22300 ACT.	410 RECS.
09:49:00	08/13/82	0	12	523	MAX.	1200 RECS.
09:50:15	08/13/82	I	30	524	RM COBINDX/MAG-RPG-IX2	
09:50:26	08/13/82	0	26	525	COBINDX/MAG-RPG-IX2 REMOVED	
09:50:27	08/13/82	0	26	526	END RM	
09:50:53	08/13/82	0	13	527	13/COPY BOJ PR IS B	0950
09:51:37	08/13/82	I	30	528	MX	
09:51:37	08/13/82	0	30	529	NUM PROGRAM-ID PR STATUS	
09:51:38	08/13/82	0	30	530	10/CMSCANDE C	SUSPENDED WAITING ON I/O
09:51:38	08/13/82	0	30	531	13/COPY B	EXECUTING
09:51:38	08/13/82	0	30	532	27/SPIM C	EXECUTING
09:51:38	08/13/82	0	30	533	28/MCSOBJ C	SWAPPED OUT, WAITING ON I/O
09:51:38	08/13/82	0	30	534	29/NDLSYS C	SHORT WAITED ON I/O
09:51:38	08/13/82	0	30	535	END MX	
09:51:49	08/13/82	I	30	536	DT	
09:51:49	08/13/82	0	30	537	13 AUG 82 82225 FRI	0951
09:52:10	08/13/82	I	30	538	WL	
09:52:19	08/13/82	0	14	539	FILE NAME	FILE STATE
09:52:19	08/13/82	0	14	540	SYS-LOG-01	READY TO TRANSFER
09:52:19	08/13/82	0	14	541	SYS-LOG-02	ACTIVE
09:52:19	08/13/82	0	14	542	SYS-LOG-03	NEXT ACTIVE
09:52:19	08/13/82	0	14	543	SYS-LOG-04	NOT USED
09:52:19	08/13/82	0	14	544	BACKUP FILE(S) ON FA	
09:52:19	08/13/82	0	14	545	SYS-MLG22300 ACT.	19 RECS.
09:52:19	08/13/82	0	14	546	MAX.	128 RECS.
09:52:19	08/13/82	0	14	547	SYS-HLD22300 ACT.	410 RECS.
09:52:19	08/13/82	0	14	548	MAX.	1200 RECS.
09:52:57	08/13/82	0	13	549	MAG-RPG-IX2 TO COBINDX/MAG-RPG-IX2 COPIED	
09:52:59	08/13/82	0	13	550	13/COPY EDJ	0952
09:59:45	08/13/82	I	30	551	PD DMA	
09:59:45	08/13/82	I	30	551	PD DMA	
09:59:46	08/13/82	0	30	552	DMA POWERED OFF	
10:00:08	08/13/82	0	31	553	DMA RPGINDX/	0 FILES OPEN

PL SYS-MLG22500

FRI 13 AUG 82 AT 11:46:32

CMS LOG FILE PRINTOUT OF SYS-MLG22500 FROM B90 SYSTEM
STATUS OF FILE IS TRANSFERRED

PAGE 1

SYSTEM CONFIGURATION :

PROCESSOR NO.	CHANNEL NO.	SUB-CHANNEL NO.	DEVICE NO.	DEVICE	HARDWARE	DEVICE	KIND	I/O (EXCLUDING COUNT RETRIES)
1	4		1	DFA	FIXED DISK (211)			47403
1	3		1	DMA	BURROUGHS STANDARD MINI-DISK			4068
1	5		1	MTA	PE REEL			0
1	5		2	MTB	PE REEL			0
1	5		3	HTC	PE REEL			0
1	5		4	HTD	PE REEL			0
1	7		1	RTA	REAL TIME CLOCK			0
1	6		1	DCA	DATA COMM			0
1	2		1	LPA	LINE PRINTER			0

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:31

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	2	YES		5760 @001680@	4544 @001100@

STATUS :- 6EDE

DESCRIPTOR:- 61E9 BEA6 FF00 0014 0080 1600 0700 C011 0000 0200 0000 0000 0000 00

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:36

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	15	NO		5760 @001680@	4608 @001200@

ERROR TYPE:- PARITY

STATUS :- 6EDE

DESCRIPTOR:- 61E9 BAA6 FF00 0014 0080 1600 0700 0012 0000 0F00 0000 0000 0000 00

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:37

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	4	YES		180 @0000B4@	4609 @001201@

STATUS :- 6EDE

DESCRIPTOR:- 2DFE BEA6 FF00 0014 00B4 0000 0700 0112 0000 0400 0000 0000 0000 00

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:40

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	8	NO		180 @0000B4@	4611 @001203@

ERROR TYPE:- PARITY

STATUS :- 6EDE

DESCRIPTOR:- 2DFE BAA9 300B 0014 00B4 0000 0701 0312 0000 0800 0000 0000 0000 00

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:44

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	1	YES	00000B40	180	4614 00012060
STATUS :- 6EDE DESCRIPTOR:- D300 BE7C 390B 0014 00B4 0000 0700 0612 0000 0100 0000 0000 00											

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:46

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	1	YES	00000B40	180	4626 00012120
STATUS :- 6EDE DESCRIPTOR:- D300 BE7C 390B 0014 00B4 0000 0700 1212 0000 0100 0000 0000 00											

MAINTENANCE ENTRY ON 08/12/82 AT 13:31:48

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
7	01/03/ /01/DMA	INPUT	00	READ	SOGDOC1	123456	1	YES	00000B40	180	4637 000121D0
STATUS :- 6EDE DESCRIPTOR:- 2DFF BEA9 380B 0014 00B4 0000 0701 1D12 0000 0100 0000 0000 00											

FRI 13 AUG 82 AT 11:46:32

CMS LOG FILE PRINTOUT OF SYS-MLG22500 FROM B90 SYSTEM
STATUS OF FILE IS TRANSFERRED

PAGE 4

OVERALL SUMMARY BETWEEN 13:31:31 ON 08/12/82 AND 13:31:48 ON 08/12/82

PROCESSOR/ CHANNEL/ SUB-CHANNEL/ DEVICE NO./ DEVICE	I/O COUNT	NO. INPUT MAINTENANCE ENTRIES	NO. OF INPUT RETRIES	NUMBER OF SUCCESSSES	NUMBER OF FAILURES	NO. OUTPUT MAINTENANCE ENTRIES	NO. OF OUTPUT RETRIES	NUMBER OF SUCCESSSES	NUMBER OF FAILURES
01/04/ /01/DFA	47403	0	0	0	0	0	0	0	0
01/03/ /01/DMA	4068	7	32	5	2	0	0	0	0
01/05/ /01/HTA	0	0	0	0	0	0	0	0	0
01/05/ /02/HTB	0	0	0	0	0	0	0	0	0
01/05/ /03/HTC	0	0	0	0	0	0	0	0	0
01/05/ /04/HTD	0	0	0	0	0	0	0	0	0
01/07/ /01/RTA	0	0	0	0	0	0	0	0	0
01/06/ /01/DCA	0	0	0	0	0	0	0	0	0
01/02/ /01/LPA	0	0	0	0	0	0	0	0	0

SQ

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	FILE	DISK ALLOCATION		AVAILABLE TABLE	REMARKS
	NB	OPEN	LENGTH	FROM	TO	INFORMATION
=====						
			32	@000000e	@00001F0	*RESERVED
			10	@000020e	@000029e	*AVAIL.TABLE
			10	@00002Ae	@000033e	*FILE DIRECTORY
			100	@000034e	@000097e	*FILE HEADERS
TEST1	1		4	@000098e	@00009Be	
COBOL1	1		351	@00009Ce	@0001FAe	
COBOL2	1		98	@0001FB0	@00025Ce	
COBOLXREF	1		223	@00025D0	@00033B0	
COBOLINTX	1		64	@00033C0	@00037B0	
COBOL.NOTES	1		256	@00037C0	@00047B0	
COBOL.NOTES	2		256	@00047C0	@00057B0	
COBOL.NOTES	3		176	@00057C0	@00062B0	
COBOL3	1		110	@00062C0	@0006990	
XA	1		24	@00069A0	@0006B10	
AMEND	1		50	@0006B20	@0006E30	
			2488	@0006E40	@00109B0	AVAILABLE SPACE
COBOL4	1		331	@00109C0	@0011E60	
			1049	@0011E70	@0015FF0	AVAILABLE SPACE

NOTE : NO ERRORS DISCOVERED IN VALIDATION PHASE - USAGE MAP OF THE DISK ENDS

TAPELR

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	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

TAPEPD ARTAPE

NRZI TAPE ARTAPE <00000> DUMPED ON THU 21
... JUN 79 CONTAINS -
MYFILE MYFILEQQ A999
END TAPEPD

APPENDIX C

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 high-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. Intrinsic are actually a part of the MCP and therefore will never be seen on a disk file listing or in the 'mix'. Examples of intrinsic 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 runnable program.

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).

USER DISK

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 D RELATED DOCUMENTATION

The following manuals provide information concerning CMS System Software:

Manual	Form Number
CMS ARCS Reference Manual	2012713
CMS COBOL Reference Manual	2007266
CMS MCP Reference Manual	2007555
CMS RPG Reference Manual	2007274
CMS MPLII Reference Manual	2007563
CMS NDL Reference Manual	1090925
CMS Data Communications Subsystem Reference Manual	1090909

