FOUR-PHASE
SYSTEMS INC.

SYSTEMS
ENGINEERING
HANDBOOK

SYSTEMS ENGINEERING HANDBOOK

TABLE OF CONTENTS

A1 System Reference

- BOOT Words.. A1-1
- Dedicated Video Memory Locations.................... A1-2
- Standard I/O Priority Assignments.................. A1-3
- Device Addresses & Interrupt Levels................ A1-4
- Recovering Blown Disc Packs......................... A1-5
- EIA RS232-C Inteface Pin Assigments................ A1-6
- Modem Strapping Options............................. A1-7
- Datascope D601B..................................... A1-9
- SYNTECH Modem Options............................... A1-10
- Communication Code Conversion Table................ A1-11
- SNA Network Generation Procedures.................. A1-15
  - Introduction.................................. A1-15
  - ACF/VTAM - ACF/NCP Operands.................... A1-17
  - ACF/VTAM - SYS1.VTAMLST MODETAB................ A1-21
  - CICS/VS Gen Considerations..................... A1-22
  - Job Entry System Gen Considerations............ A1-23
  - Sample SNA Gen Configurations.................. A1-24

A2 Code Tables

- ASCII IV/70 Display Characters...................... A2-1
- Powers Of 2 And 8................................... A2-2
- ASCII Code Set For Line Printers.................... A2-3
- ASCII Code Set For Character Printers............... A2-4
- EBCDIC Code Set..................................... A2-5
- Card Reader Code Conversion......................... A2-6
- ASCII - EBCDIC Conversion Table..................... A2-7
- EBCDIC - ASCII Conversion Table..................... A2-8
- COBOL and RPG Signed Numeric Fields................. A2-9
- 7201 Keyboard, 2260 Typewriter Style................ A2-10
- 7204 Keyboard, 2260 Data Entry Style................ A2-11
- 7202/7242 Keyboard, DATA IV Keypunch Style...........A2-12
- 7203/7243 Keyboard-DATA IV/70 Typewriter Style................................. A2-13
- 7225 Keyboard-NTP/100,150,250 (3270) Keypunch Style............................ A2-14
- 7226 Keyboard-NTP/100, 150, 250 (3270) EBCDIC Typewriter Style................................. A2-15
- 7227 Keyboard-NTP/100, 150, 250 (3270) ASCII Typewriter Style................................. A2-16
- 7228 Keyboard-NTP/100, 150, 250, (3270) Data Entry Style...................................... A2-17
- Codes Generated by 7200/20/40/01/02/42/04/07......... A2-18
- Codes Generated by 7225/75/26/46/27/40/51............ A2-19

A3 JCL Parameters

- DOS Programs...A3-1
- IDOS Programs..A3-7
- DKOS Programs..A3-14

AB	COBOL	Four-Phase COBOL Differences.........................	AB-1
		If The Compiler Halts or Loops.......................	AB-2
		If The Object Program Halts or Loops.................	AB-2
		COBOL Reserved Words.................................	AB-4
		COBOL Low Memory Allocation-48.......................	AB-5
		COBOL Low Memory Allocation-81.......................	AB-6
		COBOL Keyboard User Table-Diagram....................	AB-7
		COBOL Keyboard User Table-Detail.....................	AB-8
		JCL To LOADOV And Execute............................	AB-9
		COBOL Program Origination Points.....................	AB-10
		File Control Block (FCB) Format......................	AB-11
		FCB Word Detail (Card Reader)........................	AB-12
		(Printer)............................	AB-13
		(Mag Tape)...........................	AB-14
		(SD File)............................	AB-15
		(DC File)............................	AB-16
		(DISAM File).........................	AB-17
		GENERAL..............................	AB-18
		ISAM File Layout.....................................	AB-21
		DISAM File Structure.................................	AB-22
		DISAM File layout....................................	AB-23
		DISAM Table..	AB-24
		DISAM Table Detail...................................	AB-27
		SD File Structure....................................	AB-28
		Disc File Organization...............................	AB-29
		LIBGEN Questions.....................................	AB-30
AC	DOS	Commonly Used Low Memory Locations In MONITR.........	AC-1
		Clearing A Printer Halt Under DOS....................	AC-2
		Allocation Table.....................................	AC-3
		Disc Directory.......................................	AC-3
		Sector Format Of A Chained File......................	AC-4
		Structure Of A Chained File..........................	AC-5
		$DISC Request Table..................................	AC-6
		$IDISC Request Table.................................	AC-7
		$ITAPE Request Table.................................	AC-8
		$JTAPE Request Table.................................	AC-9
		$TPE Request Table...................................	AC-10
AD	IDOS	IDOS Notes...	AD-1
		IDOS Disk Structure..................................	AD-2
		Communications Region Format.........................	AD-3
		IDOS Directory, Allocation Table, Boot...............	AD-4
		IDOS Directory Format................................	AD-5
		Sector Format Of A Chained File......................	AD-6
		Error Bits In System Subroutines.....................	AD-7
		-> Clearing a Printer Halt...........................	AD-8
		$IDISC Request Table.................................	AD-9
		$JDISC Request Table.................................	AD-10
AE	2260	Summary Of Command Key Functions.....................	AE-1
		2260 Simulator Video Gaps............................	AE-2
		2260 Simulator Options...............................	AE-3
		Placing The 2260 Simulator On The 8250 Diskette......	AE-4
		Placing The 2260 Simulator On Punched Cards..........	AE-5
		2260 REMOTE Simulator-Hardware Configurations........	AE-6
		F5 Diagnostic Key Display-Remote Version............	AE-7
		Steps To Generate A Remote 2260 Simulator............	AE-8
		Initialization Constants.............................	AE-10
		Source Code Alteration Method........................	AE-10
		Absolute Load Module Alteration Method...............	AE-10
		Octal Card Deck "Patch" Cards Ateration Method.......	AE-11

AF 2260 Local

2260 Local Simulator-Hardware Configurations......... AF-1
Steps To Generate A Local 2260 Simulator............. AF-2
2260 Local Simulator-Channel Debugging............... AF-4
Channel Apaptor Diagnostic Status Words.............. AF-6
F5 Diagnostic Key Display-Local Version.............. AF-7

AG 3270

SECTION I System Configuration
Terminal Support..................................... AG-2
Simulator Configuration Words........................ AG-2
A76 Configurator..................................... AG-3
Execution Time Reconfiguration....................... AG-5
MINIGEN.. AG-6
Store and Forward.................................... AG-8
TBLEDT... AG-9

SECTION II NTP/150
SB3270 Interface..................................... AG-10
Entry Codes.. AG-10
Detail Codes... AG-11
NTP/150 Subroutines.................................. AG-13
SBRSET... AG-14
Software Action Codes................................ AG-15

SECTION III DEBUGGING
Escape Key Functions................................. AG-16
Display of Attribute Characters...................... AG-17
Display of Line Control Characters................... AG-18
TRC327 Execution..................................... AG-19
TRC327 Output.. AG-20
Taking Memory Dumps.................................. AG-22
DMP327 Execution..................................... AG-22
DMP327 Output.. AG-23

SECTION IV BISYNC and 3270 PROTOCOL
3270 Control Characters.............................. AG-27
Bisync Data Link Control............................. AG-28
3270 Message Formats................................. AG-29
Local and Remote Comand Codes........................ AG-30
Write and Copy Control Characters.................... AG-30
Buffer Control Orders................................ AG-31
Attribute Bytes...................................... AG-31
Attention ID Byte.................................... AG-32
Remote Status/Sense Bytes............................ AG-33
Local Status and Sense Bytes......................... AG-35

SECTION V COMMUNICATION CONTROLLERS
8436-2 Status and Control............................ AG-36
7073 Status, Control, and IOID's..................... AG-37
7074 Status and Contril (WIDGET)..................... AG-39

SECTION VI MISCELLANEOUS TOPICS
Bypassing Communication Controller Halt.............. AG-40
Adjustment of Dual Intensity......................... AG-40
Marketing Considerations............................. AG-40

AH DATA IV
Version 1

(AH04)

Mastering Directory Sector.......................... AH-1
Standard 4-Word Sector Header........................ AH-2
CONFIG Sector Layout................................. AH-3
CONFIG Sector Notes.................................. AH-4
JOB Directory Sector Layout.......................... AH-5
Batch Directory Sector Layout........................ AH-6
Data Sector Layout................................... AH-8
FORMAT/VALUE Set Directory........................... AH-9
FORMAT Code Sector Layout............................ AH-10
VALUE Set Sector Layout.............................. AH-11
Overlay Sector Layout................................ AH-12
Directory of In-Ram Format Pages..................... AH-13
Disc Space Allocation Calculation.................... AH-14
RAM Layout... AH-15
IN-RAM Job Directory................................. AH-16
IN-RAM Format/Value Set Directory.................... AH-17
User Table Layout.................................... AH-18
User Table Footnotes................................. AH-19
Dumping Ram to Disc or Tape.......................... AH-20
Frequent Questions About CONFIG...................... AH-21

(AH04)
AI DATA IV/70
Version 2/3

Memory Layout.. AI-1
Active Job Directory In RAM (JOBRAM)................. AI-2
Active Format/Value Set Directory In RAM (FVSRAM).... AI-3
Page Directory In RAM (PAGDIR)....................... AI-4
Index Set Directory In RAM (IXRRAM).................. AI-5
User Table Skeleton Definitions...................... AI-6
User Table Flag Words................................ AI-8
Disc Sector Pool and Request Queus................... AI-10
Zone Pointer Table................................... AI-11
Overlay Directory.................................... AI-11
Submonitor Control Blocks............................ AI-12
Zone Layout.. AI-13
Disc Layout.. AI-13
Sector Structure..................................... AI-14
Allocation Sector.................................... AI-16
Maintenance (IOD) Sector............................. AI-17
Master Sector.. AI-18
CONFIG Sector.. AI-19
Keys Sector.. AI-22
Job Directory.. AI-23
Batch Directory...................................... AI-24
Batch Directory Flags................................ AI-25
Batch Index.. AI-26
Data Sectors... AI-27
Format/Value Set Directory........................... AI-28
Format Object Code Sector............................ AI-29
Value Set Sector..................................... AI-29
Index Set Organization............................... AI-30
Index Sets... AI-31
Overlay Object Code.................................. AI-32
Overlays... AI-33
Writing Overlays..................................... AI-34
Writing Transfer Programs............................ AI-34
DEBUG Screen... AI-35
DYNAMO Operation..................................... AI-36
Dump Procedure....................................... AI-37
Performance Monitor.................................. AI-40
SYSGEN Utility Programs.............................. AI-41

(AI continued next page)

AI (continued)

RELEASE AI15 DETAILS

Memory Layout.. AI-43
User Table Skeleton Definitions...................... AI-44
Overlay Directory.................................... AI-47
Overlay Request Queus................................ AI-48
Master Sector.. AI-49
CONFIG Sector.. AI-50
Writing Overlays..................................... AI-54
Writing Transfer Programs............................ AI-55
DEBUG.. AI-56
DYNAMO... AI-57
Dump Procedure....................................... AI-60
Single Words In Ram.................................. AI-61

AJ RPG
(AJ06)
Conventional Definition and Use if Indicators........ AJ-1
Matching Record Algorithm for Ascending Files........ AJ-2

AM LAM
(AM08)
Status Block Information And Meaning................. AM-1
Format Of LAM Common Area............................ AM-2
Error Status... AM-2
Summary of Source Files.............................. AM-3
SYSGEN Configuration Word............................ AM-6
Printer Carriage Controls............................ AM-7
Line Trace Format.................................... AM-8

AN RBS Non-Spool
(AN05)
RBS Operator Console Display......................... AN-1
RBS Keyboard Operations.............................. AN-3
Function Keys.. AN-3
Operator Commands.................................... AN-5

AP DKOS
(BJ03)
Create An Operational Diskette....................... AP-1
Bypassing The Auto-Boot Sequence..................... AP-1
Taking A Checkpoint.................................. AP-2
Reloading After A Checkpoint......................... AP-2
DKOS System Dump..................................... AP-2
Diskette Track Format................................ AP-3
$DSKT Request Table.................................. AP-4

AQ RBS-Spooling
RBS Spooling Operator Console Display................ AQ-1
VCQ Display.. AQ-2
RBS Spooling Keyboard Operations..................... AQ-3
RAM Dumps.. AQ-4
Unattended Mode...................................... AQ-4
Transfer Statements.................................. AQ-5
Parameter Definitions................................ AQ-6
Syntax Errors.. AQ-7
RBS Disc Files-Formats............................... AQ-8
RBS Disc Files-Flag Byte............................. AQ-8

AR RSU
(BJ03)
Remote Software Update............................... AR-1
Screen Display....................................... AR-2
Keyboard Mode.. AR-2
Control Files.. AR-2

AS NP/80 Support
Generate An 8265 Disc Pack Using An 8231 Disc Pack for Releases Prior to A4........................ AS-1
Generate An 8265 Disc Pack Using An 8131 Disc Pack for Release A4 and up........................... AS-2
Generate An 8265 Disc Pack Using An NPDTUX Tape for Releases prior to A4........................ AS-2
Generate An 8265 Disc Pack Uding AN NPDTUX Tape for Release A4 and up........................... AS-3
Items To Be Aware.................................... AS-4
Upgrading A 8265 Disc Pack To A More Current Release (Using An 8231 Disc Pack)......................... AS-4
Upgrading An 8265 Disc Pack To A More Current Release (Using An NPDTUX Tape..................... AS-4
Backing Up An 8265 Disc Pack Onto NPDTUX Tapes....... AS-5

AT MLAM
HASP Message Block Format.......... AT-1
Record Control Byte.......... AT-2
Sub Record Control Byte.......... AT-2
String Control Byte.......... AT-3
END-OF-JOB Sequence.......... AT-3
MLAM Line Control Block.......... AT-4
MLAM Status Word.......... AT-4
MLAM Status Table.......... AT-5

AV Starter
STRSAM.......... AV-1
Adding EDIT Rule Tables.......... AV-4
Starter Utility Program-STRUTL.......... AV-5

AY ForeWord
System Memory Organization.......... AY-1
User Table Definitions.......... AY-2
System Disc Organization.......... AY-5
Text Area Organization.......... AY-6
Allocation Sector.......... AY-6
Text Area Directories.......... AY-6
Data Sectors.......... AY-7
Creating Password Text Areas.......... AY-8
System Text Area - WRDFIL.......... AY-8
Flag Byte Description.......... AY-9
System Naming Conventions.......... AY-10
Configuration File.......... AY-11
Terminal Parameters.......... AY-12
Communication Patameters.......... AY-13
TBLIDC Utility.......... AY-14
Processor JCL.......... AY-18
Taking/Printing a Dump.......... AY-20

AY06
System Memory Organization - AY06.......... AY-22
Physical Memory Requirements - AY06.......... AY-23
User Table Definitions - AY06.......... AY-24
Allocation Sector - AY06.......... AY-31
Data Sectors - AY06.......... AY-32
Flag Byte Description - AY06.......... AY-35
File Naming Conventions - AY06.......... AY-36
CHKTXT - AY06.......... AY-37
Taking a Dump - AY06.......... AY-39
ForeWord Formatted Dump - AY06.......... AY-40

BA VISION

(BA03/04)

Memory Layout.. BA-1
Memory Layout for Extended Memory Systems............ BA-2
Active Job Directory in Ram (JOBRAM)................ BA-3
Active Format/Value Set Directory in RAM (FVSRAM).... BA-4
Page Directory in RAM (PAGDIR)....................... BA-5
Index Set Directory In RAM (IXRRAM).................. BA-6
User Table Skeleton Definitions...................... BA-7
User Table Data Structures........................... BA-9
Zone Pointer Table................................... BA-12
Zone Layout.. BA-13
Sector Structure..................................... BA-14
Allocation Sector.................................... BA-15
Master Sector.. BA-16
CONFIG Sector.. BA-17
Job Directory.. BA-20
Batch Directory...................................... BA-21
Batch Directory Flags................................ BA-22
Data Sectors... BA-23
Maintenance Sector (OID)............................. BA-24
Format/Value Set Directory........................... BA-25
Format Object Code Sector............................ BA-27
Value Set Sector..................................... BA-28
Index Set Orginization............................... BA-29
Index Sets... BA-30
Index Sector... BA-30
File Access Data Structures (on Disc)................ BA-31
    File Access Tabel................................ BA-34
    Sequential Disc.................................. BA-35
    Index Sets....................................... BA-35
    Printers... BA-35
    Forms Control Block.............................. BA-36
    Card Reader...................................... BA-36
    Magnetic Tape.................................... BA-36
    Magnetic Tape Control Block...................... BA-37
Disc Sector Pool and Request Queue................... BA-38
Overlay Object Code.................................. BA-40
Overlays... BA-41
Level 8 Processing................................... BA-42
Dump Procedure....................................... BA-43
DYNAMO Operation..................................... BA-44
Performance Monitor.................................. BA-47
3270 Attributes as Displayed on Video................ BA-49
Single Words in RAM.................................. BA-50

<u>RELEASE BA04 DETAILS</u>

Memory Layout BA-53
User Table Skeleton Definitions...................... BA-54
CONFIG Sector.. BA-57
Data Sectors... BA-61
Overlays... BA-62
Dump Procedure....................................... BA-63
DEBUG.. BA-64
DYNAMO... BA-65
Single Words in RAM.................................. BA-67

BD PROGRAMMER WORKSTATION (PWS)
- Configuring Communications.......................... BD-1
- Create or Modify Operator Ids File.................. BD-4
- Defining PWS File Types............................. BD-5
- Directory List Utility.............................. BD-6
- Log Report Utility.................................. BD-7
- PWS Log Message Organization........................ BD-9
- Communications Line Trace........................... BD-10
- Calculating Memory Requirements..................... BD-11
- Formatted Dump Printing Facility.................... BD-12
- PWS Memory Load Layout.............................. BD-13
- Task Control Block.................................. BD-15
- Word Expansions..................................... BD-17
- File Control Block.................................. BD-20
- Directory Control Block............................. BD-22
- Communications Control Block Table.................. BD-23
- MLAM Line Control Block............................. BD-25
- LAM Line Control Block.............................. BD-26
- Printer Control Block............................... BD-27
- Card Reader Control Block........................... BD-28
- Virtual Printer Control Block....................... BD-28
- Directory Sector Format BD-29
- Header Sector Format................................ BD-30
- Data Sector Format.................................. BD-31
- PWS Chained File Format............................. BD-32
- PWS Data Sector Format.............................. BD-33
- Component List...................................... BD-34

BE COBOL 74
- Syntax Summary...................................... BE-1
    - Reserved Words.................................. BE-1
    - Notes... BE-3
- COBOL Language Formats.............................. BE-4
    - Identification Division......................... BE-4
    - Environment Division............................ BE-4
    - Data Division.................................. BE-6
    - Procedure Division.............................. BE-7
- Changes - AB to BE.................................. BE-13
- Compiler TEMP Files................................. BE-19
- Error File Format................................... BE-20
- PASCAL Error Messages............................... BE-20
- Compiler Table Sizes................................ BE-21

BF LAM/8437
- LAM Internal Errors ($LOGIC)........................ BF-1
- LAM Status Indicators............................... Bf-1
- Request Table Format................................ BF-3
- LAM SYSGN... BF-5
- LAM 8437 Dump Analyzer.............................. BF-6
- LAM Calling Sequences............................... BF-6
- Application of Fixes................................ BF-7

BG MLAM/8437
- HASP Workstation (Model 20) Protocol Notes.......... BG-1
- External Interface Codes............................ BG-1
- Request Table Format................................ BG-2
- MLAM Translation Tables............................. BG-4

BK 3270/SDLC
- CFG327.. BK-1
- Debug Facilities.................................... BK-3
    - Escape Key Functions............................ BK-3
    - TRC327.. BK-4
- SDLC an SNA Protocol................................ BK-8
    - SNA Terminology................................. BK-8
    - SDLC Line Transmissions......................... BK-9
- DTCOMM Utility...................................... BK-13
- Cross Reference Table to AG 3270 Simulator.......... BK-14

BN	MFE	Fixed Locations Within MFE	BN-1
		Program Load Table	BN-1
		Partition Control Block	BN-2
		MFE Dump Considerations	BN-5
		Task Control Block	BN-6
		System Maintenance Services	BN-7
		Interrupt Control Block	BN-7
BP	CU3270/8437	A32ERR Error Codes	BP-1
		C32xxx Routine Error Codes	BP-1
		CU3270 8437 Dump Analyzer	BP-1
		Requests from IVxx/Completions from 8437	BP-2
		Requests from 8437/Completions from IVxx	BP-3
		CU3270 Timers	BP-4
		3271 SDLC	BP-5
BR	COMM SERVICES	$XFER	BR-1
		ELOG	BR-1
BS	COMM UTILITIES	DTCOMM, DTCOMF, COMDMP	BS-1
		TRACE, COMTRA	BS-2
BQ	NP/80	NP/80 Disk Drive Characteristics	BQ-1
		NP/80 Disk Sector Usage	BQ-1
		Volume Header Sector	BQ-2
		NP/80 Console Switch Settings	BQ-3
		NP/80 Memory Layout	BQ-4
		IVxx Cutword, NP/80 Hang Register Contents	BQ-5
		NPOS Class Codes	BQ-5
		ALMFMD Section names	BQ-6
		NP/80 Coded Halts	BQ-7
BV	3270 SIMULATOR	Configurator	BV-1
		MINGEN	BV-2
		JCL for Simulator Utilities	BV-2
		Debug Routines	BV-3

BOOT WORDS

Model	Description	Boot Word
8230	Disc cartridge	37705121
8001/8003	Card reader	37705101
8240	50Mb disc	37705201
8250	Diskette	37707175
8260/8280/8290	NP/80 discs	37705165
8270	10MB disc	37705121 (drive 0) 37705125 (drive 1) 37705131 (drive 2) 37705135 (drive 3)
8511/8512	9-Track/800 BPI	37705221
8513/8504	9-Track/1600 BPI	37705241
8507	7-Track	37705261
7071/7072	2260 Channel adapter	37704641
8001-5/8003-5	Buffered Card Reader	37703115

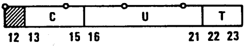
Clearing Memory to a Constant

1. AUTO to MANUAL
2. SYSTEM RESET
3. STEP
4. LOAD constant into RA
5. LOAD 45700000 into TIR
6. NORMAL to REPEAT
7. STEP
8. REPEAT to NORMAL

Booting a 7001 System (with no BOOT)

1. AUTO to MANUAL, RESET then STEP
2. LOAD boot word into X1
3. LOAD boot word into TIR
4. MANUAL to AUTO

I/O Select (CUT) Word



Channel	C Address
0	0000
1	0400
2	1000
3	1400
4	2000
5	2400
6	3000
7	3400

Type	T Address
Data Out	0000
Data In	0001
Control	0002
Status	0003

Unit	U Address	Unit	U Address
00	0000	040	0200
01	0004	041	0204
02	0010	042	0210
03	0014	043	0214
04	0020	044	0220
05	0024	045	0224
06	0030	046	0230
07	0034	047	0234
010	0040	050	0240
011	0044	051	0244
012	0050	052	0250
013	0054	053	0254
014	0060	054	0260
015	0064	055	0264
016	0070	056	0270
017	0074	057	0274
020	0100	060	0300
021	0104	061	0304
022	0110	062	0310
023	0114	063	0314
024	0120	064	0320
025	0124	065	0324
026	0130	066	0330
027	0134	067	0334
030	0140	070	0340
031	0144	071	0344
032	0150	072	0350
033	0154	073	0354
034	0160	074	0360
035	0164	075	0364
036	0170	076	0370
037	0174	077	0374

Dedicated Memory Locations

Table 3—4. Dedicated Memory Locations

Octal Location	Function	Octal Location	Function
00000	Interrupt level 0	00012	Interrupt level 5
00002	Interrupt level 1	00014	Interrupt level 6
00004	Interrupt level 2	00016	Interrupt level 7
00006	Interrupt level 3	00041	Arithmetic Trap, Supervisory Trap
00010	Interrupt level 4		
7001, 48 Character/Line Video Systems†		**7001, 81 Character/Line Video Systems†**	
00060-00657	Video display area A	00140-00732‡	Video display area A
01060-01657	Video display area B	00740-01532‡	Video display area B
02060-02657	Video display area C	02140-02732‡	Video display area C
03060-03657	Video display area D	02740-03532‡	Video display area D
04060-04657	Video display area E	04140-04732‡	Video display area E
05060-05657	Video display area F	04740-05532‡	Video display area F
06060-06657	Video display area G	06140-06732‡	Video display area G
07060-07657	Video display area H	06740-07532‡	Video display area H
4300, 4500, and 7002, 48 Character/Line Video Systems†		**4300, 4500, and 7002, 81 Character/Line Video Systems†**	
00060-00657	Video display area 000	00140-00732‡	Video display area 000
01060-01657	Video display area 001	00740-01532‡	Video display area 001
02060-02657	Video display area 002	02140-02732‡	Video display area 002
03060-03657	Video display area 003	02740-03532‡	Video display area 003
04060-04657	Video display area 004	04140-04732‡	Video display area 004
05060-05657	Video display area 005	04740-05532‡	Video display area 005
06060-06657	Video display area 006	06140-06732‡	Video display area 006
07060-07657	Video display area 007	06740-07532‡	Video display area 007
10060-10657	Video display area 010	10140-10732‡	Video display area 010
11060-11657	Video display area 011	10740-11532‡	Video display area 011
12060-12657	Video display area 012	12140-12732‡	Video display area 012
13060-13657	Video display area 013	12740-13532‡	Video display area 013
14060-14657	Video display area 014	14140-14732‡	Video display area 014
15060-15657	Video display area 015	14740-15532‡	Video display area 015
16060-16657	Video display area 016	16140-16732‡	Video display area 016
17060-17657	Video display area 017	16740-17532‡	Video display area 017
20060-20657	Video display area 020	20140-20732‡	Video display area 020
21060-21657	Video display area 021	20740-21532‡	Video display area 021
22060-22657	Video display area 022	22140-22732‡	Video display area 022
23060-23657	Video display area 023	22740-23532‡	Video display area 023
24060-24657	Video display area 024	24140-24732‡	Video display area 024
25060-25657	Video display area 025	24740-25532‡	Video display area 025
26060-26657	Video display area 026	26140-26732‡	Video display area 026
27060-27657	Video display area 027	26740-27532‡	Video display area 027
30060-30657	Video display area 030	30140-30732‡	Video display area 030
31060-31657	Video display area 031	30740-31532‡	Video display area 031
32060-32657	Video display area 032	32140-32732‡	Video display area 032
33060-33657	Video display area 033	32740-33532‡	Video display area 033
34060-34657	Video display area 034	34140-34732‡	Video display area 034
35060-35657	Video display area 035	34740-35532‡	Video display area 035
36060-36657	Video display area 036	36140-36732‡	Video display area 036
37060-37657	Video display area 037	36740-37532‡	Video display area 037

† Video systems with 40 or 80 characters/line are achieved by programming blanks in the appropriate character positions.

‡ There are 5 unused memory locations at the end of each video line for 81 character/line systems. For example, the characters for the first line of area A occupy locations 00140-00172 while the second line of characters occupies locations 00200-00232.

A312B

Four-Phase Standard I/O Priority Assignments

Channel Number	Unit Number (Octal)	Select Word‡ (Octal)	Device Description
0	—	—	Reserved for real-time clock: an INR instruction is placed in memory location 0 and a 60 Hz clock is tied to the INT 0 line on Interface Card 1.
1†	33	0554	Synchronous Data Set (8435-8436).
1	35	0564	Asynchronous Data Set (8411) or other interactive device.
1	50‡	0640	360/370 Channel Adapter (7071/7072 Series); initial interrupt.
1	51‡	0644	360/370 Channel Adapter (7071/7072 Series); continue interrupt.
1	52‡	0650	360/370 Channel Adapter (7071/7072 Series); end interrupt.
1	53‡	0654	360/370 Channel Adapter (7071/7072 Series); data in/out.
1	54‡	0660	360/370 Channel Adapter (7073 Series); initial interrupt.
1	55‡	0664	360/370 Channel Adapter (7073 Series); continue interrupt.
1	56‡	0670	360/370 Channel Adapter (7073 Series); end interrupt.
1	57‡	0674	360/370 Channel Adapter (7073 Series); data in/out.
2†	20‡	1100	Card Reader, Unbuffered (8001/8003); character ready.
2	22‡	1100	Card Reader, Unbuffered (8001/8003); end of card.
2	24	1120	Disc 0 (8231).
2	25	1124	Disc 1 (8231).
2	26	1130	Disc 2 (8231).
2	27	1134	Disc 3 (8231).
2	34	1160	NP/80 (Sub Unit 0)
2	35	1164	NP/80 (Sub Unit 1)
2	36	1170	NP/80 (Sub Unit 2)
2	37	1174	NP/80 (Sub Unit 3)
2	40	1200	Disc 0 (8241).
2	41	1204	Disc 1 (8241).
2	42	1210	Disc 2 (8241).
2	43	1214	Disc 3 (8241).
2	44	1220	Mag Tape 0 select (8511/8512); data interrupt for selected drive.
2	45	1224	Mag Tape 1 select (8511/8512); status interrupt for selected drive.
2	46	1230	Mag Tape 2 select (8511/8512).
2	47	1234	Mag Tape 3 select (8511/8512).
2	50	1240	Mag Tape 0 select (8513/8504); data interrupt for selected drive.
2	51	1244	Mag Tape 1 select (8513); status interrupt for selected drive.
2	52	1250	Mag Tape 2 select (8513).
2	53	1254	Mag Tape 3 select (8513).
2	54	1260	Mag Tape 0 select (8507); data interrupt for drive 0.
2	55	1264	Mag Tape 0 status interrupt (8507).
3†	0-37	1400-1574	Keyboard Units (7200 Series) 0 through 37_8, data ready, no error.
3	40-77	1600-1774	Keyboard Units 0 through 37_8, data ready, character lost.

A096G

Device Addresses and Interrupt Levels

Relative Address ⑥	Fixed Memory	Level 1	Level 2	Level 3		Level 5	Level 6
				System IV/70	System IV/40		
00	INR Level 0			BRM Kbrd 0 ①		BRM 8122 0 ③	
1				BRM Kbrd 1 ①		BRM 8122 1	
2	IOID Level 1			BRM Kbrd 2 ①		BRM 8122 2	
3				BRM Kbrd 3 ①		BRM 8122 3	
4	IOID Level 2			BRM Kbrd 4 ①		BRM 8122 4	
5				BRM Kbrd 5 ①		BRM 8122 5	
6	IOID Level 3			BRM Kbrd 6 ①		BRM 8122 6	
7				BRM Kbrd 7 ①		BRM 8122 7	
10	BRM Level 4			BRM Kbrd 10 ①		BRM 8122 10	
11				BRM Kbrd 11 ①		BRM 8122 11	
12	IOID Level 5			BRM Kbrd 12 ①		BRM 8122 12	
13				BRM Kbrd 13 ①		BRM 8122 13	
14	IOID Level 6			BRM Kbrd 14 ①		BRM 8122 14	
15				BRM Kbrd 15 ①		BRM 8122 15	
16	BRM Level 7			BRM Kbrd 16 ①		BRM 8122 16	
17				BRM Kbrd 17 ①		BRM 8122 17	
20			IO Card Reader (8001/8003) char ready	BRM Kbrd 20 ①	Not used	BRM 8122 20 ③	
21				BRM Kbrd 21 ①	Not used	BRM 8122 21	
22			IO Card Reader (8001/8003) end of card	BRM Kbrd 22 ①	Not used	BRM 8122 22	
23				BRM Kbrd 23 ①	Not used	BRM 8122 23	IO 8001-5xx/8003-5xx
24			BRM 8231 or 8271 Disc	BRM Kbrd 24 ①	Not used	BRM 8122 24	
25			8231 or 8271 Disc ④	BRM Kbrd 25 ①	Not used	BRM 8122 25	
26			8231 or 8271 Disc ④	BRM Kbrd 26 ①	Not used	BRM 8122 26	
27			8231 Address ④	BRM Kbrd 27 ①	Not used	BRM 8122 27	
30		IO Sync Data Set 8437	BRM future discs	BRM Kbrd 30 ①	Not used	BRM 8122 30	BRM 8135,8145,8149,8154
31		IO Sync Data Set 8437	BRM future discs	BRM Kbrd 31 ①	Not used	BRM 8122 31	BRM 8131
32		BRM Sync Data Set 8437	BRM future discs	BRM Kbrd 32 ①	Not used	BRM 8122 32	
33		BRM Sync Data Set (8435/8436/8437)	BRM future discs	BRM Kbrd 33 ①	Not used	BRM 8122 33	
34			BRM future discs	BRM Kbrd 34 ①	Not used	BRM 8122 34	
35		BRM Async Data Set (8411)	BRM future discs	BRM Kbrd 35 ①	Not used	BRM 8122 35	
36			BRM future discs	BRM Kbrd 36 ①	Not used	BRM 8122 36	BRM KB/Printer/Auto-Dial
37			BRM future discs	BRM Kbrd 37 ①	Not used	BRM 8122 37	BRM 8251 Diskette
40			BRM 8241 Disc	BRM Kbrd 0 ②			
41	BRM TRAP		8241 Address ④	BRM Kbrd 1 ②			
42			8241 Address ④	BRM Kbrd 2 ②			
43			8241 Address ④	BRM Kbrd 3 ②			
44			IO Mag Tape (8511/8512) data int	BRM Kbrd 4 ②			
45			BRM Mag Tape (8511/8512) status int	BRM Kbrd 5 ②			
46			8511/8512 Address ④	BRM Kbrd 6 ②			
47			8511/8512 Address ④	BRM Kbrd 7 ②			
50		BRM Ch Adap (7071/7072) init int	IO Mag Tape (8513/8504) data int	BRM Kbrd 10 ②			
51		BRM Ch Adap (7071/7072) cont int	BRM Mag Tape (8513/8504) status int	BRM Kbrd 11 ②			
52		BRM Ch Adap (7071/7072) end int	8513 Address ④	BRM Kbrd 12 ②			
53		IO Ch Adap (7071/7072) data in/out	8513 Address ④	BRM Kbrd 13 ②			
54		BRM Ch Adap (7073) init int	IO Mag Tape (8507) data int	BRM Kbrd 14 ②			
55		BRM Ch Adap (7073) cont int	BRM Mag Tape (8507) status int	BRM Kbrd 15 ②			
56		BRM Ch Adap (7073) end int		BRM Kbrd 16 ②			
57		IO Ch Adap (7073) data in/out		BRM Kbrd 17 ②			
60	Video Area A48			BRM Kbrd 20 ②	Vcc Hi ⑤		
61	Video Area A48			BRM Kbrd 21 ②	Vcc Low ⑤		
62	Video Area A48			BRM Kbrd 22 ②	Vcc Nom⑤		
63	Video Area A48			BRM Kbrd 23 ②	Vdd Hi ⑤		
64	Video Area A48			BRM Kbrd 24 ②	Vdd Low ⑤		
65	Video Area A48			BRM Kbrd 25 ②	Vdd Nom ⑤		
66	Video Area A48			BRM Kbrd 26 ②	Vgg Hi ⑤		
67	Video Area A48			BRM Kbrd 27 ②	Vgg Low ⑤		
70	Video Area A48			BRM Kbrd 30 ②	Vgg Nom ⑤		
71	Video Area A48			BRM Kbrd 31 ②	Lamp 1 on ⑤		
72	Video Area A48			BRM Kbrd 32 ②	Lamp 1 off ⑤		
73	Video Area A48			BRM Kbrd 33 ②	Lamp 2 on ⑤		
74	Video Area A48			BRM Kbrd 34 ②	Lamp 2 off ⑤		
75	Video Area A48			BRM Kbrd 35 ②	Run lamp on ⑤		
76	Video Area A48			BRM Kbrd 36 ②	Fail lamp on ⑤		
77	Video Area A48			BRM Kbrd 37 ②	Run & Fail lamp off ⑤		

① Data ready, no error.
② Data ready,character lost for System IV/70 Keyboards 0-37 and System IV/40 Keyboards 0-17.
③ The controller for the 8122 Printers can support 020 printers; another controller must be added for each block of 020 or fewer printers.
④ Discs and magnetic tape units may use certain unit addresses without corresponding IOID table addresses being assigned.
⑤ These signals are for diagnostic use only.
⑥ On 64 keyboard systems add octal 100 to relative address for keyboard 040 through 077

Recovering Blown Disc Packs

Bad CRC

One of the most common malfunctions of the disc systems is writing data with bad CRC characters. Since most software does not do a read after write check, the error is not discovered until the sector is read, causing programs to halt or go into an error condition because of the CRC check bit in the status word. It is possible to correct this condition but care must be taken to ensure that the data is actually correct.

> 8230 disc- execute COPY01 with the bad pack on drive 0 and a scratch pack on drive 1. When the message "INPUT PACK IS BAD" occurs (you will want to note the sector address), just clear the halt. If the cause of the error is a bad CRC, then the data in memory (which may be good) will be written to the output pack with a good CRC. Check sector against directory dump and replace any affected files.
>
> 8260 disc- execute COPY60 with the printer on. If CRC errors on the input pack are encountered, the cylinder, track and sector number of each error will be printed. COPY60 will copy the entire pack as best it can, then display/print an unsuccessful completion message. The bad cylinder, track and sector numbers can then be converted to octal sector numbers and checked against a directory dump to determine which files were affected. The documentation for NPFMTX describes the calculations needed.

Rewriting Headers

The processors FMTX and NPFMTX can be used to write and verify headers for every sector on a disc that contains data. As it is possible (though not likely) to destroy data while restoring headers, this is considered a last resort.

CRTDMP

When the octal sector number of a bad sector is known, you can read the sector to the screen using CRTDMP, note the error status, and rewrite the sector to disk. CRTDMP will read the data as best it can, and write what it found back with a good CRC word. Since the rewritten sector may have changed from what it was before, you must check it against a directory dump, and if the sector was in a file, replace the file.

EIA RS232-C INTERFACE PIN ASSIGNMENTS

PIN NUMBER	CKT	CCITT EQUIV.	DESCRIPTION
1	AA	101	Protective ground
2	BA	103	Transmitted Data
3	BB	104	Received Data
4	CA	105	Request to Send (RTS)
5	CB	106	Clear to Send (CTS)
6	CC	107	Data Set Ready (DSR)
7	AB	102	Signal Ground
8	CF	109	Rcv. Line Sig. Det.
9	-	-	Reserved for Data Set Testing
10	-	-	Reserved for Data Set Testing
11	-	-	Unassigned
12	SCF	122	Sec. Rcvd. Line Sig. Det.
13	SCB	121	Sec. Clear to Send
14	SBA	118	Sec. Trans. Data
15	DB	114	Trans. Clock (DCE source)
16	SBB	119	Sec. Rcv. Data
17	DD	115	Sec. Clock (DCE source)
18	-	-	Unassigned
19	SCA	120	Sec. Request to Send
20	CD	108.2	Data Term. Ready (DTR)
21	SG	110	Signal Quality Det.
22	CE	125	Ring Indicator
23	CH/CI	111/112	Data Signal Sel.
24	DA	113	Trans. Clock
25	-	-	Unassigned

RS-232 LEADS

RTS (RS)	- Request to send, issued from Data Terminal. Modem should respond with CTS after a fixed delay period.
CTS (CS)	- Clear to send, issued from Modem in response to RTS after a fixed delay period (delay may be zero). CTS may be constantly high if strapped for constant carrier.
TD	- Transmit Data, issued from Data Terminal. TD will contain one binary bit of transmit data for each clock pulse of TT.
RD	- Receive Data, issued from Modem. RD will contain one binary bit of received Data for each clock pulse of RT.
TT	- Transmit Timing, issued from Modem. TT provides clocking for the Data Terminal. For each pulse of TT the Data terminal must gate a transmit data bit on the TD lead. Four-Phase requires Modem be strapped for internal clock.
RT	- Receive timing, issued from Modem. RT provides clocking for the Data terminal. For each pulse of RT, the Modem will gate a transmit data bit on the RD lead. Requires Modem be strapped for 'internal clock'.
CO	- Carrier On, issued from Modem when carrier is detected from the Remote Modem. Also called 'Carrier Detect -CD'

DTR - Data Terminal Ready, issued from Data Terminal. Not used in leased lines. For switched lines indicates terminal is ready to establish connection and, for auto-answer in response to RI, indicates terminal requests incoming call be answered. Modem should respond with DSR.

DSR - Data Set Ready, issued from Modem to indicate communications connection established. Should always be high for leased lines or manual-answer. For auto-answer, issued in response to DTR after the call has been answered and connection established.

RI - Ring Indicator, issued from Modem when phone is ringing.

DSS - Data Signal Select, also known as 'Rate Select,' issued from the Data Terminal to indicate which of two modem speeds to use on a dual speed modem. Note that most modems with both an internal rate select (RS232) and an external rate select switch (eg. 2400/4800), will consider the internal selection as overriding the manual switch.

MODEM STRAPPING OPTIONS

201 C or equivalent modem on 2 wire line.

Required Options for Four-Phase Operation

	Option
New Sync not used	YA
Transmit Timing Internal	YC
Ring Indication EIA term 22	YG
Grounding option	YK
Auto Answer either will work	YE (could also be YF)

One of the following must be selected:

2 - wire switched network	XD
or	
2 - wire non-switched network	XE

Also the following will depend on installation.

Transmit level (as determined by phone company)
Line impedance (as determined by phone company)
Compromise Equalizer - could be in 00 out
Carrier on sensitivity - depends on line
Use with 828 DAS - normally no (unless the 828 interface is asked for by customer)

For 4 wire use the only change that need be made is instead of options XD or XE one of the following must be specified:

Switched carrier - with 7 ms Delay XA option or
Continuous carrier - with 7 ms Delay XB option.
*** XC option.

Note that carrier or sensitivity will be affected by type of line.

*** XC option - Do not select the XC option, which is Continuous carrier with no delay. This option requires a cabling change to the controller. (RS and CS must be jumper cabled.)

201A/B Modem or Equivalent on 2 wire line.

Required Four-Phase Options

Internal Timing
EIA interface
Half Duplex
Carrier controlled by request to send without new sync.
150 ms CTS delay

Additional Options which customer needs to specify.

With or without alternate voice
(Note: If unattended answer is specified, alternate voice is required.)

Permanent or selective unattended answer.
(Permanent unattended answer always answers the telephone. Selective will answer only if "Auto" button on handset is in auto position.)

Without automatic calling

For operation on 4 wire line the only two changes are:

Full duplex instead of half duplex and 7 ms CTS delay

MODEM STRAPPING OPTIONS - CONTINUED

For 208 Modem or equivalent on 2 wire line.

Required Options for Four-Phase operation.

SIA	Down	DSR off in AL mode
SIB	Up	No comp equalizer test
SIC	Down	Switched request-to-send
S3A	Down	Xmit internally timed
S3B	Down	Retrain not used
S4A	Up	1-Sec Holdover disable
S4B	Down	Switched carrier
S4C	Down	New synch not used by customer

Following items are installation options

S3C	Auxiliary data set used or not
S2	Equalizer adjustment (normally use factory settings; all up)

For 4 wire use simply change

S4B	Up	Continuous carrier

DATASCOPE D601B

GETTING STARTED

Inspect for obvious shipping damage. If in doubt, remove the dust cover (two-screws, top rear) and reseat all plug-in boards and connectors. Replace cover.

NOTE-If shipping damage is suspected, notify the carrier that delivered the unit immediately. Do not destroy any shipping material. (This should be saved for future use in any case.)

Turn on power -- fan runs, power light on
Connect EIA interfaces at rear. CPU to upper connector, Modem to lower.

Set switches as follows for 8-bit synchronous data:
SEND/RCV --- HDX-4 (four-wire line) or HDX-2 (two-wire line)
FRAMING --- SYNC-8
DISPLAY --- A = ASCII B = EBCDIC HEX = HEXADECIMAL
MARKER --- SEE DISPLAY NOTES BELOW
LINE SPEED --- MODEM
*TWO-CHARACTER SYNC --- OFF
FRAMING PATTERN ---
Normal Setting for: ASCII - switches 2, 3, & 5 up
switches 1, 4, 6, 7, 8 down

EBCDIC- switches 2, 5, & 6 up
switches 1, 3, 4, 7, 8 down
SYNC RESET
Normal Setting for: ASCII - switch 3 up
switches 1, 2, 4, 5, 6, 7, 8 down
EBCDIC- switches 1, 2, 3, 5, 6 up
switches 4, 7, 8 down
SEND (Invert) --- DOWN
REC (Invert) --- DOWN
BOTH (Invert) --- DOWN
1-8/8-1 --- DOWN
*AUTO STOP --- OFF
*SUPPRESS --- OFF
REPLAY SPEED --- VARY
RUN STOP --- RUN
*SAVE Cont. --- PRESS down - leave in center position

* OPTIONAL FEATURE - may not be installed on all units.

DISPLAY NOTES

The display operates in low intensity when the DATASCOPE is searching for character phase (SYNC) and in high intensity when the unit is in SYNC.
Receive data is identified by an underline; transmitted data is not underlined.
The MARKER switch causes an inverted (black-on-white) highlighted display when the selected signal line is high:
CD - Carrier Detect (highlights received data)
RTS - Request to Send (highlights tranmitted data)
EVENT - Event Mark

PLEASE READ YOUR INSTRUCTION MANUAL FOR FURTHER INFORMATION

SYNTECH MODEM OPTIONS

SYNTECH 208+ OPTION	A								B								C							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
STANDARD SET	U	U	U	U	D	U	D	D	D	D	D	D	U	U	U	U	D	U	U	U	D	D	U	D
AUTO ANSWER	U																							
INCOMING ONLY		U																						
ABT - NO TX					U																			
ABT - NO RX						U																		
ANS W/O DTR							U																	
AUTO THRU TST								D																
NO DSR IN TST			U																					
TX LEVEL +1dB																						D	D	D
0dB																						U	D	D
-1dB																						D	U	D
-2dB																						U	U	D
-3dB																						D	D	U
-6dB																						U	D	U
-9dB																						D	U	U
-12dB																						U	U	U
RX LEVEL:																								
0 to -30dB																	U	U						
-10 to -40dB																	D	U						
-20 to -50dB																	U	D						
-30 to -60dB																	D	D						
4800 BPS									D	D	D	D												
2400 BPS									D	U	D	U												
Ext. CMD 48/24									U	D	U	D												
Ext. TX Clock				D																				
CTS = 150 ms.																					D			
CTS = 50 ms.																					U			
Tx Delay EQ in													U											
Tx&Rx Amp EQ														U	D									
TX Amp in														U	U									
Non Syntech Amp in														D	U									
NOT USED																U			U	U				

COMMUNICATION CODE CONVERSION TABLE

EBCDIC			ASCII	
DISPLAY	OCTAL	HEX	OCTAL	GRAPHIC
o	000	00	000	NUL
△	001	01	001	SOH
ƀ	002	02	002	STX
¢	003	03	003	ETX
◢	004	04	234	
	005	05	011	PT,HT
≠	006	06	206	
%	007	07	177	DEL
	010	08	227	
←	011	09	215	
\	012	0A	216	
/	013	0B	013	
£	014	0C	014	FF
■	015	0D	015	
¬	016	0E	016	
|	017	0F	017	
|	020	10	020	DLE
[	021	11	021	SBA
	022	12	022	EUA
\	023	13	023	IC
√	024	14	235	
‾	025	15	205	NL
^	026	16	010	
]	027	17	207	
□	030	18	030	
	031	19	031	EM
■	032	1A	222	
•	033	1B	217	
◀	034	1C	034	DUP
▮	035	1D	035	SF,IGS
▶	036	1E	036	FM,IRS
\	037	1F	037	ITB,IUS
	040	20	200	
!	041	21	201	
"	042	22	202	
#	043	23	203	
$	044	24	204	
%	045	25	012	
&	046	26	027	ETB
'	047	27	033	ESC
(	050	28	210	
)	051	29	211	
*	052	2A	212	
+	053	2B	213	
,	054	2C	214	
-	055	2D	005	ENQ
.	056	2E	006	
/	057	2F	007	BEL
0	060	30	220	
1	061	31	221	
2	062	32	026	SYN
3	063	33	223	
4	064	34	224	
5	065	35	225	
6	066	36	226	
7	067	37	004	EOT
8	070	38	230	
9	071	39	231	
:	072	3A	232	
;	073	3B	233	
<	074	3C	024	RA
=	075	3D	025	NAK
>	076	3E	236	
?	077	3F	032	SUB

COMMUNICATION CODE CONVERSION TABLE

EBCDIC			ASCII	
DISPLAY	OCTAL	HEX	OCTAL	GRAPHIC
@	100	40	040	
A	101	41	240	
B	102	42	241	
C	103	43	242	
D	104	44	243	
E	105	45	244	
F	106	46	245	
G	107	47	246	
H	110	48	247	
I	111	49	250	
J	112	4A	133	÷
K	113	4B	056	.
L	114	4C	074	<
M	115	4D	050	(
N	116	4E	053	+
O	117	4F	136	↑
P	120	50	046	&
Q	121	51	251	
R	122	52	252	
S	123	53	253	
T	124	54	254	
U	125	55	255	L-PAD
V	126	56	256	
W	127	57	257	
X	130	58	260	
Y	131	59	261	
Z	132	5A	041	!
÷	133	5B	044	$
X	134	5C	052	*
¦	135	5D	051	)
↑	136	5E	073	;
—	137	5F	135	|
	140	60	055	-
a	141	61	057	/
b	142	62	262	
c	143	63	263	
d	144	64	264	
e	145	65	265	
f	146	66	266	
g	147	67	267	
h	150	68	270	
i	151	69	271	
j	152	6A	174	¦
k	153	6B	054	,
l	154	6C	045	%
m	155	6D	137	—
n	156	6E	076	>
o	157	6F	077	?
p	160	70	272	
q	161	71	273	
r	162	72	274	
s	163	73	275	
t	164	74	276	
u	165	75	277	
v	166	76	300	
w	167	77	301	
x	170	78	302	
y	171	79	140	`
z	172	7A	072	:
	173	7B	043	#
	174	7C	100	@
	175	7D	047	'
	176	7E	075	=
	177	7F	042	"

COMMUNICATION CODE CONVERSION TABLE

EBCDIC			ASCII	
DISPLAY	OCTAL	HEX	OCTAL	GRAPHIC
	200	80	303	
	201	81	141	a
	202	82	142	b
	203	83	143	c
	204	84	144	d
	205	85	145	e
	206	86	146	f
	207	87	147	g
	210	88	150	h
	211	89	151	i
	212	8A	304	
	213	8B	305	
	214	8C	306	
	215	8D	307	
	216	8E	310	
	217	8F	311	
	220	90	312	
	221	91	152	j
	222	92	153	k
	223	93	154	l
	224	94	155	m
	225	95	156	n
	226	96	157	o
	227	97	160	p
	230	98	161	q
	231	99	162	r
	232	9A	313	
	233	9B	314	
	234	9C	315	
	235	9D	316	
	236	9E	317	
	237	9F	320	
	240	A0	321	
	241	A1	176	-
	242	A2	163	s
	243	A3	164	t
	244	A4	165	u
	245	A5	166	v
	246	A6	167	w
	247	A7	170	x
	250	A8	171	y
	251	A9	172	z
	252	AA	322	
	253	AB	323	
	254	AC	324	
	255	AD	325	
	256	AE	326	
	257	AF	327	
	260	B0	330	
	261	B1	331	
	262	B2	332	
	263	B3	333	
	264	B4	334	
	265	B5	335	
	266	B6	336	
	267	B7	337	
	270	B8	340	
	271	B9	341	
	272	BA	342	
	273	BB	343	
	274	BC	344	
	275	BD	345	
	276	BE	346	
	277	BF	347	

COMMUNICATION CODE CONVERSION TABLE

EBCDIC			ASCII	
DISPLAY	OCTAL	HEX	OCTAL	GRAPHIC
	300	C0	173	
	301	C1	101	A
	302	C2	102	B
	303	C3	103	C
	304	C4	104	D
	305	C5	105	E
	306	C6	106	F
	307	C7	107	G
	310	C8	110	H
	311	C9	111	I
	312	CA	350	
	313	CB	351	
	314	CC	352	
	315	CD	353	
	316	CE	354	
	317	CF	355	
	320	D0	175	
	321	D1	112	J
	322	D2	113	K
	323	D3	114	L
	324	D4	115	M
	325	D5	116	N
	326	D6	117	O
	327	D7	120	P
	330	D8	121	Q
	331	D9	122	R
	332	DA	356	
	333	DB	357	
	334	DC	360	
	335	DD	361	
	336	DE	362	
	337	DF	363	
	340	E0	134	
	341	E1	237	
	342	E2	123	S
	343	E3	124	T
	344	E4	125	U
	345	E5	126	V
	346	E6	127	W
	347	E7	130	X
	350	E8	131	Y
	351	E9	132	Z
	352	EA	364	
	353	EB	365	
	354	EC	366	
	355	ED	367	
	356	EE	370	
	357	EF	371	
	360	F0	060	0
	361	F1	061	1
	362	F2	062	2
	363	F3	063	3
	364	F4	064	4
	365	F5	065	5
	366	F6	066	6
	367	F7	067	7
	370	F8	070	8
	371	F9	071	9
	372	FA	372	
	373	FB	373	
	374	FC	374	
	375	FD	375	
	376	FE	376	
	377	FF	377	T-PAD

SNA NETWORK GENERATION PROCEDURES

INTRODUCTION

The following document has been prepared by the Network Support Center (NSC), in an attempt to provide a Systems Programmer, already familiar with the IBM software components, with the tools necessary to upgrade or install an SNA network. It is not meant to be a re-write of existing IBM documentation, but rather a quick reference listing of those operands which may cause some confusion when attempting to implement Four-Phase Systems into an SNA environment. In addition, this document only attempts to cover the most current software program levels of the most widely used products. Questions may still arise regarding other software programs, and the Network Support Center has been implemented to assist with these problems. This version covers VTAM, NCP, CICS, and JES2. Future versions will add information on RES, JES3, IMS, TCAM, and other products.

SNA NETWORK GENERATION PROCEDURES

ACF/VTAM - ACF/NCP Generation Considerations

*A.) PCCU (VTAM only Macro Instruction)

Identifies the communications controller to VTAM.

B.) BUILD

Generates specific parameters for the communications controller and the network being defined.

C.) SYSCNTRL

Identifies the dynamic control facilities to be included in the Network Control Program.

D.) HOST

Identifies the parameters specific to the host operating environment.

E.) CSB

Defines the communications scanner.

F.) LUPOOL

Specifies a pool of logical units used by the Network Control Program for dial-up terminals.

ACF/VTAM - ACF/NCP OPERANDS (3271-12/3277)

G1.) GROUP (3271-12/3277)

Defines the parameters common to all lines contained in this group.

1.)	LNCTL = SDLC	All SDLC lines in this group.
2.)	TYPE = NCP	Lines controlled by NCP only.
	PEP	Lines controlled by NCP and Emulation.
3.)	SPEED = 9600	Maximum speed allowed (Specific configuration may further restrict this value.)
4.)	POLLED = YES	Polled terminals on these lines. (Omit this parameter for ACF/NCP Release 3.)

G2.) LINE (3271-12/3277)

Defines specific operating characteristics of the individual line.

1.)	ADDRESS = X'nnn'	Line interface address.(Port on 3705)
2.)	CLOCKING = EXT	Clocking supplied by the modem.
3.)	DUPLEX = HALF or FULL	
4.)	NRZI = YES OR NO	(Modem dependent)
5.)	TRANSFR =	User dependent.
6.)	RETRIES =	User dependent.
* 7.)	ISTATUS =	User dependent.

G3.) SERVICE ORDER = (PU^{1}, PU^{2}, . . . PU^{N}) (3271-12/3277)

Specifies the order in which the PU's on the line are to be serviced.

G4.) PU (3271-12/3277)

Specifies individual physical unit characteristics.

1.)	PUTYPE = 1	Type 1 physical unit.
2.)	BNNSUP = 3270	This PU is an SDLC 3271.
3.)	ADDR = X'nn'	Link level address.(Polling address)
4.)	MAXOUT = 7	7 PIU's can be sent at a time.
5.)	PASSLIM = 7	Max. # of PIU's to this controller.
6.)	MAXDATA = 261	Max. data bytes received in 1 Xfer.
*7.)	SSCPFM = USS3270	Character coded msgs. supported.(Required for 3271)

G5.) LU (3271-12/3277)

Specifies individual logical unit characteristics.

1.)	LOCADDR =	Terminal address.(First LU begins with 0)
2.)	VPACING = (2,1)	(User dependent).
3.)	PACING = (1,1)	Pacing needed on every PIU. (This is required for 3271-12)
*4.)	MODETAB =	See MODETAB considerations - page 7.

ACF/VTAM - ACF/NCP OPERANDS (3770 LEASED LINE)

H1.) GROUP (3770 Leased Line)

1.)	LNCTL = SDLC	All SDLC lines in this group.
2.)	TYPE = NCP	Lines controlled by NCP only.
	PEP	Lines controller by NCP and emulation.
3.)	SPEED = 9600	Maximum speed allowed (Specific configuration may further restrict this value.)
4.)	POLLED = YES	Polled terminals on these lines. (Omit this parameter for ACF/NCP Release 3.)

H2.) LINE (3770 Leased Line)

1.)	ADDRESS = 'nnn'	Line interface address.
2.)	CLOCKING = EXT.	Clocking supplied by the modem.
3.)	DUPLEX = HALF	Half-Duplex line protocol.
4.)	NRZI = YES OR NO	(Modem dependent)
5.)	TRANSFR =	User dependent.
6.)	RETRIES =	User dependent.
*7.)	ISTATUS =	User dependent.

H3.) SERVICE ORDER = (PU^1, PU^2, . . . PU^N) (3770 Leased Line)

H4.) PU (3770 Leased Line)

1.)	PUTYPE = 2	PU type 2.
2.)	ADDR = X'nn'	Link level address.
3.)	MAXDATA = 265	Max. 3770 buffer size.
4.)	MAXOUT = 1	1 PIU to be sent at a time.
5.)	PASSLIM = 1	Max. # of PIU's to this controller.

H5.) LU (3770 Leased Line)

1.)	LOCADDR = 1	Only 1 logical unit on 3770.
2.)	VPACING = (2,1)	(User dependent).
3.)	PACING = (1,1)	Pacing required on every PIU.
*4.)	MODETAB =	See MODETAB considerations - page 7.

ACF/VTAM - ACF/NCP OPERANDS (3770 SWITCHED LINE)

I1.) LUPOOL (3770 Switched Line)

1.)	NUMBER =	Number of LU's available.

I2.) GROUP (3770 Switched Line)

1.)	LNCTL = SDLC	All SDLC lines in this group.
2.)	TYPE = NCP	Lines controlled by NCP only.
	PEP	Lines controlled by NCP and Emulation.
3.)	SPEED = 4800	Normal dial-up maximum speed.
4.)	POLLED = YES	Polled termianls on these lines. (Omit this parameter for ACF/NCP Release 3.)
5.)	DIAL = YES	Lines in this group are switched.

I3.) LINE (3770 Switched Line)

1.)	ADDRESS = X'nnn'	Line interface address.
2.)	CLOCKING = EXT	Clocking supplied by the modem.
3.)	DUPLEX = HALF	Half-Duplex line protocol.
4.)	NRZI = YES or NO	(Modem dependent).
5.)	TRANSFR =	User dependent.
6.)	RETRIES =	User dependent.
*7.)	ISTATUS =	User dependent.

I4.) PU (3770 Switched Line)

1.)	PUTYPE = 2	PU type 2.
2.)	MAXLU = 1	Single logical unit only.

* For 3770 Switched, a second member must be defined under SYS1.VTAMLST.

K1.) VBUILD

Assigns a subarea value to the major mode for VTAM's use in assigning addresses to the minor modes.

K2.) PU

Defines a switched SNA major node.
(Code a PU statement for each physical unit in the switched major node)

1.)	ADDR =	Station address.
2.)	MAXDATA = 265	Max 3770 buffer size.
3.)	MAXOUT = 1	1 PIU to be sent at a time.
4.)	PASSLIM = 1	Max # of PIU's to this controller.
5.)	PUTYPE = 2	PU type 2.
6.)	IDBLK =	12 Bit Binary Block Number.
7.)	IDNUM =	20 Bit Binary Identification Number.

The block number (obtained from the Component Description Manual) together with the Identification number (randomly selected) combine to form a 48 bit station ID that is used in XID exchange during the dial procedure. The contents of IDBLK and IDNUM must agree with the XID jumper bits and XID 3770 type used by SNAFIG. See pg. 2-3 of the Data IV/Vision SNA 3770 User's manual (SIV/70 - 55 - 29B).

K3.) LU

Specifies each logical unit associated with a physical unit within a switched SNA major node.

1.)	LOCADDR = 1	Only one logical unit on 3770.
2.)	PACING = (1,1)	Pacing required on every PIU.
3.)	VPACING = (2,1)	User dependent.
*4.)	MODETAB =	See MODETAB consideration - see page 7.

ACF/VTAM - SYS1.VTAMLST MODETAB CONSIDERATIONS

1.) MODETAB = Specifies the logon mode table name used for the LU.

If the "Modetab" operand is omitted, the IBM-supplied logon mode table is used for the logical unit. The format for the IBM-supplied table is as follows:

```
ISTINALM    MODETAB   (OS/VS1 and OS/VS2 SVS only)
IBM3770     MODEENT  LOGMODE = BATCH,  FMPROF = X'03',
                     TSPROF = X'03', PRIPROT = X'A3',
                     SECPROT = X'A3', COMPROT = X'7080'.

IBMS3270    MODEENT  LOGMODE = S3270, FMPROF = X'02',
                     TSPROF = X'02' PRIPROT = X'71',
                     SECPROT = X'40', COMPROT = X'2000'.
```

*For further information, reference pg. 4-2 of the ACF/VTAM System Programmer's Guide (SC38-0258).

CICS/VS Generation Considerations (Terminal Control Table only)

A.) DFHTCT TYPE = Terminal (3271-12/3277 ONLY)

Defines each individual devices terminal control table characteristics.

1.) TRMTYPE = 3277	Definition for 3271-12.
2.) TRMMODL = 1 or 2	Model # for this terminal.
3.) ACCMETH = VTAM	VTAM controls this terminal.
4.) TIOAL =	Minimum message size. (User dependent)
5.) TRMSTAT = TRANSCEIVE	Automatic transaction initation issued.
6.) RELREQ =	Release to application or VTAM allowed. (User dependent)
7.) TCTUAL =	Process control information field length. (User dependent)
8.) FEATURE = (DCKYBD, UCTRAN, AUDALARM)	Features supported.
9.) RUSIZE = 256	RU size for this terminal.
10.) CHNASSY = NO	Chaining not permitted.

JOB ENTRY SYSTEM (JES2 4.1) Generation Considerations

A.) Line nnn

Specifies one logical unit's characteristics as used during remote job entry.

1.)	UNIT = SNA	All subparameters (except "Password") are ignored.

B.) &NUMLNES = nnn

Number of teleprocessing lines available.

C.) &NUMRJE = nnn

Number of remote terminal definitions (Default = value specified for "&NUMLNES")

D.) &NUMTPBF = nnn

Number of JES2 teleprocessing buffers. (The minimum requirement for SNA is three buffers plus two buffers for every SNA RJE terminal).

E.) RMTnnn

Characteristics of each SNA remote terminal.

1.)	LUTYPE1	This remote is an SNA terminal.
2.)	BUFSIZE = 256,512	Max. size RU for this terminal.
3.)	COMP	Terminal supports the compression/expansion features.
4.)	NOCMPCT	Default value (terminal does not support compaction).
5.)	LUNAME = cccccccc	Logical unit name (must be the same as the name defined to VTAM).
6.)	NUMPR = 1	Max. # of printers supported.
7.)	NUMRD = 1	Max. # of readers supported.
8.)	NUMPU = 1	Max. # of punches supported.
9.)	CONSOLE	Operator console is supported.

SAMPLE SNA GEN CONFIGURATIONS

```
******************************************************************************
                           3270 AND 3770 SDLC LINES
******************************************************************************

MEMBER   NCP001    SYS1.VTAMLST

     SNAGRP   GROUP LNCTL=SDLC,
                    TYPE=NCP,
                    SPEED=2400,
                    POLLED=YES          Omit for ACF/NCP Release 3

     LNE01    LINE  ADDRESS=001,
                    CLOCKNG=EXT,
                    DUPLEX=HALF,
                    NRZI=NO

              SERVICE ORDER=(PU3270,PU3770)

     PU3270   PU    PUTYPE=1,
                    BNNSUP=3270,
                    ADDR=C1,
                    MAXOUT=7,
                    PASSLIM=7,
                    MAXDATA=261,
                    SSCPFM=USS3270,
                    VPACING=(2,1),
                    PACING=(1,1)

     LU327700 LU    LOCADDR=0,
                    USSTAB=USS3271,
                    BATCH=NO

     LU327701 LU    LOCADDR=1,
                    USSTAB=USS3271,
                    BATCH=NO

     PU3770   PU    PUTYPE=2,
                    ADDR=D6
                    MAXDATA=265,
                    MAXOUT=1,
                    PASSLIM=1,
                    VPACING=(2,1),
                    PACING=(1,1)

    LU377000  LU   LOCADDR=1
```

```
********************************************************************************
                                CICS SDLC 3270
********************************************************************************

A01A  DFHTCT  TYPE=TERMINAL,
              TRMIDNT=A01A,
              TRMTYPE=3277,TRMMODL=2,
              ACCMETH=VTAM,
              NETNAME=LU327000,
              TIOAL=1500,
              TRMSTAT=TRANSCEIVE,
              GMMSG=YES,
              RELREQ=(YES,YES),
              CONNECT=AUTO,
              TCTUAL=20,
              FEATURE=(DCKYBD,UCTRAN,AUDALARM)

A01B  DFHTCT  TYPE=TERMINAL,
              TRMIDNT=A01B,
              TRMTYPE=3277,TRMMODL=2,
              ACCMETH=VTAM,
              NETNAME=LU327001,
              TIOAL=1500,
              TRMSTAT=TRANSCEIVE,
              GMMSG=YES,
              RELREQ=(YES,YES),
              CONNECT=AUTO,
              TCTUAL=20,
              FEATURE=(DCKYBD,UCTRAN,AUDALARM)
```

IV/70 Display Characters

Third Octal Digit	First & Second Octal Digits															
	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
0	○	⧻‡	|	☐	SP	(	0	8	@	H	P	X	`	h	p	x
1	△†	←	[	ɣ‡	!	)	1	9	A	I	Q	Y	a	i	q	y
2	ƀ†	\	╲	▩	"	*	2	:	B	J	R	Z	b	j	r	z
3	¢	/	▬	°	#	+	3	;	C	K	S	÷	c	k	s	{
4	◢	£	√†	◀	$	,	4	<	D	L	T	×	d	l	t	¦
5	ǂ‡	■	—	▌	%	-	5	=	E	M	U	|	e	m	u	}
6	‡†	¬	^	▶	&	.	6	>	F	N	V	↑	f	n	v	~
7	⌑†	|	]	\	'	/	7	?	G	O	W	_	g	o	w	///

†These symbols are currently displayed but not supported. Other symbols may be substituted on later models.

‡Used as dual intensity characters and not displayed on 4300,4500,5001,6001,6501,7002, and 7009.

A107D

POWERS OF 2 AND 8

2^n & 8^m	m	n	2^{-n} & 8^{-m}
1	0	0	1.0
2		1	0.5
4		2	0.25
8	1	3	0.125
16		4	0.062 5
32		5	0.031 25
64	2	6	0.015 625
128		7	0.007 812 5
256		8	0.003 906 25
512	3	9	0.001 953 125
1 024		10	0.000 976 562 5
2 048		11	0.000 488 281 25
4 096	4	12	0.000 244 140 625
8 192		13	0.000 122 070 312 5
16 384		14	0.000 061 035 156 25
32 768	5	15	0.000 030 517 578 125
65 536		16	0.000 015 258 789 062 5
131 072		17	0.000 007 629 394 531 25
262 144	6	18	0.000 003 814 697 265 625
524 288		19	0.000 001 907 348 632 812 5
1 048 576		20	0.000 000 953 674 316 406 25
2 097 152	7	21	0.000 000 476 837 158 203 125
4 194 304		22	0.000 000 238 418 579 101 562 5
8 388 608		23	0.000 000 119 209 289 550 781 25
16 777 216	8	24	0.000 000 059 604 644 775 390 625
33 554 432		25	0.000 000 029 802 322 387 695 312 5
67 108 864		26	0.000 000 014 901 161 193 847 656 25
134 217 728	9	27	0.000 000 007 450 580 596 923 828 125
268 435 456		28	0.000 000 003 725 290 298 461 914 062 5
536 870 912		29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	10	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648		31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296		32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	11	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184		34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368		35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	12	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472		37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944		38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	13	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776		40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552		41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	14	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208		43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416		44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	15	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664		46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328		47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	16	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312		49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 624		50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25

ASCII Code Set for Line Printers

Third Octal Digit	First & Second Octal Digits															
	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
0					SP	(	0	8	@	H	P	X	`	h	p	x
1					!	)	1	9	A	I	Q	Y	a	i	q	y
2					"	*	2	:	B	J	R	Z	b	j	r	z
3					#	+	3	;	C	K	S	[	c	k	s	{
4					$	,	4	<	D	L	T	\	d	l	t	¦
5					%	-	5	=	E	M	U	]	e	m	u	}
6					&	.	6	>	F	N	V	↑	f	n	v	¬
7					'	/	7	?	G	O	W	←	g	o	w	□

Control and form feed characters. The only legal characters below 040 are:

012 line feed
014 form feed
015 carriage return (not necessary)

64-Character ASCII subset, recognized by all line printers

32 additional codes recognized by 8147 and 8152 printers

Above 200, only vertical format codes are legal (200 to 232), all others are illegal.

The 8135 character printer is actually a line printer from a programming standpoint.

A230C

ASCII Code Set for Character Printers

Third Octal Digit	First & Second Octal Digits															
	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
0	NUL				SP	(	0	8	@	H	P	X	`	h	p	x
1					!	)	1	9	A	I	Q	Y	a	i	q	y
2		LF			"	*	2	:	B	J	R	Z	b	j	r	z
3				ESC	#	+	3	;	C	K	S	[	c	k	s	{
4		FF			$	,	4	<	D	L	T	\	d	l	t	¦
5		CR			%	-	5	=	E	M	U	]	e	m	u	}
6					&	.	6	>	F	N	V	^	f	n	v	~
7	BEL				'	/	7	?	G	O	W	_	g	o	w	

NUL, BEL, LF, and CR are recognized by Teletype Printers (8100 Controller). LF, FF, CR and DEL are recognized by the 8131 Printer. DEL (0377) is used as the null character with this printer.

SYSOUT converts any code below 040 into carriage return and line feed, except 014 (FF) is accepted as a form feed after the carriage return.

64-Character ASCII subset, recognized by all character printers. Note that codes 133 ([), 134 (\), 135 (]), and 136 (^) are displayed as ÷, X, |, and ↑ respectively on the 7100/7101 video displays.

31 additional characters that may be recognized by some printers. The printer converts 177 to 137 and all 0140 to 0177 codes into a 64 character subset.

A087E

EBCDIC Code Set

Second Hex Digit	First Hex Digit															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	●	|			SP	&	—						{	}	X	0
1	△	[					/		a	j	~		A	J		1
2	ƀ	\		^					b	k	s		B	K	S	2
3	¢	▬							c	l	t		C	L	T	3
4									d	m	u		D	M	U	4
5	←	▬	\						e	n	v		E	N	V	5
6		⧻	]						f	o	w		F	O	W	6
7	⫽		°	◢					g	p	x		G	P	X	7
8		☐							h	q	y		H	Q	Y	8
9		ϒ						`	i	r	z		I	R	Z	9
A					▶	|	▬	:								
B	/				.	$	,	#								
C	£	◀		√	<	*	%	@								
D	■	▮	ǂ	¯	(	)	_	'								
E	¬	▶	‡		+	;	>	=								
F	|	\	¤	█	|	¬	?	■								

A232A

Card Reader Code Conversion

ASCII Code	ASCII Graphic	Hollerith Code	029 Graphic	026 Graphic Business Science	ASCII Code	ASCII Graphic	Hollerith Code	029 Graphic	026 Graphic Business Science
240	SP		SP	SP	055	-	11	-	-
261	1	1	1	1	312	J	11-1	J	J
262	2	2	2	2	113	K	11-2	K	K
063	3	3	3	3	314	L	11-3	L	L
264	4	4	4	4	115	M	11-4	M	M
065	5	5	5	5	116	N	11-5	N	N
066	6	6	6	6	317	O	11-6	O	O
267	7	7	7	7	120	P	11-7	P	P
270	8	8	8	8	321	Q	11-8	Q	Q
071	9	9	9	9	322	R	11-9	R	R
072	:	8-2	:		041	!	11-2-8	!	
243	#	8-3	#	# or =	044	$	11-3-8	$	$
300	@	8-4	@	@ or ,	252	*	11-4-8	*	*
047	'	8-5	'		251	)	11-5-8	)	
275	=	8-6	=		273	;	11-6-8	;	
042	"	8-7	"		335	] †	11-7-8	¬	
060	0	0	0		246	&	12	&	& or +
257	/	0-1	/	/	101	A	12-1	A	A
123	S	0-2	S	S	102	B	12-2	B	B
324	T	0-3	T	T	303	C	12-3	C	C
125	U	0-4	U	U	104	D	12-4	D	D
126	V	0-5	V	V	305	E	12-5	E	E
327	W	0-6	W	W	306	F	12-6	F	F
330	X	0-7	X	X	107	G	12-7	G	G
131	Y	0-8	Y	Y	110	H	12-8	H	H
132	Z	0-9	Z	Z	311	I	12-9	I	I
134	\ †	0-2-8	(0-2-8)		333	| †	12-2-8	¢	
254	,	0-3-8	,		056	.	12-3-8	.	
245	%	0-4-8	%	% or (	074	<	12-4-8	<	¤ or)
137	_	0-5-8	_		050	(	12-5-8	(	
276	>	0-6-8	>		053	+	12-6-8	+	
077	?	0-7-8	?		336	^ †	12-7-8	|	

† ASCII Codes 333, 134, 335, and 336 display as ÷, X, |, and ↑ on the System IV/70 video display.

A097C

ASCII To EBCDIC As Recommended By Four-Phase Systems

	000	020	040	060	100	120	140	160	200	220	240	260	300	320	340	360
000	00	10	40	F0	7C	D7	79	97	20	30	41	58	76	9F	B8	DC
001	01	11	5A	F1	C1	D8	81	98	21	31	42	59	77	A0	B9	DD
002	02	12	7F	F2	C2	D9	82	99	22	1A	43	62	78	AA	BA	DE
003	03	13	7B	F3	C3	E2	83	A2	23	33	44	63	80	AB	BB	DF
004	37	3C	5B	F4	C4	E3	84	A3	24	34	45	64	8A	AC	BC	EA
005	2D	3D	6C	F5	C5	E4	85	A4	15	35	46	65	8B	AD	BD	EB
006	2E	32	50	F6	C6	E5	86	A5	06	36	47	66	8C	AE	BE	EC
007	2F	26	7D	F7	C7	E6	87	A6	17	08	48	67	8D	AF	BF	ED
010	16	18	4D	F8	C8	E7	88	A7	28	38	49	68	8E	B0	CA	EE
011	05	19	5D	F9	C9	E8	89	A8	29	39	51	69	8F	B1	CB	EF
012	25	3F	5C	7A	D1	E9	91	A9	2A	3A	52	70	90	B2	CC	FA
013	0B	27	4E	5E	D2	4A	92	C0	2B	3B	53	71	9A	B3	CD	FB
014	0C	1C	6B	4C	D3	E0	93	6A	2C	04	54	72	9B	B4	CE	FC
015	0D	1D	60	7E	D4	5F	94	D0	09	14	55	73	9C	B5	CF	FD
016	0E	1E	4B	6E	D5	4F	95	A1	0A	3E	56	74	9D	B6	DA	FE
017	0F	1F	61	6F	D6	6D	96	07	1B	E1	57	75	9E	B7	DB	FF

ASCII To EBCDIC As Recommended By Four-Phase Systems

	000	020	040	060	100	120	140	160	200	220	240	260	300	320	340	360
000	00	10	40	F0	7C	D7	79	97	20	30	41	58	76	9F	B8	DC
001	01	11	4F	F1	C1	D8	81	98	21	31	42	59	77	A0	B9	DD
002	02	12	7F	F2	C2	D9	82	99	22	1A	43	62	78	AA	BA	DE
003	03	13	7B	F3	C3	E2	83	A2	23	33	44	63	80	AB	BB	DF
004	37	3C	5B	F4	C4	E3	84	A3	24	34	45	64	8A	AC	BC	EA
005	2D	3D	6C	F5	C5	E4	85	A4	15	35	46	65	8B	AD	BD	EB
006	2E	32	50	F6	C6	E5	86	A5	06	36	47	66	8C	AE	BE	EC
007	2F	26	7D	F7	C7	E6	87	A6	17	08	48	67	8D	AF	BF	ED
010	16	18	4D	F8	C8	E7	88	A7	28	38	49	68	8E	B0	CA	EE
011	05	19	5D	F9	C9	E8	89	A8	29	39	51	69	8F	B1	CB	EF
012	25	3F	5C	7A	D1	E9	91	A9	2A	3A	52	70	90	B2	CC	FA
013	0B	27	4E	5E	D2	4A	92	C0	2B	3B	53	71	9A	B3	CD	FB
014	0C	1C	6B	4C	D3	E0	93	6A	2C	04	54	72	9B	B4	CE	FC
015	0D	1D	60	7E	D4	5A	94	D0	09	14	55	73	9C	B5	CF	FD
016	0E	1E	4B	6E	D5	5F	95	A1	0A	3E	56	74	9D	B6	DA	FE
017	0F	1F	61	6F	D6	6D	96	07	1B	E1	57	75	9E	B7	DB	FF

A591B

ASCII To EBCDIC As Recommended By ANSII X3.26-1970

EBCDIC To ASCII As Recommended by Four-Phase Systems

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00	000	020	200	220	040	046	055	272	303	312	321	330	173	175	134	060
01	001	021	201	221	240	251	057	273	141	152	176	331	101	112	237	061
02	002	022	202	026	241	252	262	274	142	153	163	332	102	113	123	062
03	003	023	203	223	242	253	263	275	143	154	164	333	103	114	124	063
04	234	235	204	224	243	254	264	276	144	155	165	334	104	115	125	064
05	011	205	012	225	244	255	265	277	145	156	166	335	105	116	126	065
06	206	010	027	226	245	256	266	300	146	157	167	336	106	117	127	066
07	177	207	033	004	246	257	267	301	147	160	170	337	107	120	130	067
08	227	030	210	230	247	260	270	302	150	161	171	340	110	121	131	070
09	215	031	211	231	250	261	271	140	151	162	172	341	111	122	132	071
0A	216	222	212	232	133	041	174	072	304	313	322	342	350	356	364	372
0B	013	217	213	233	056	044	054	043	305	314	323	343	351	357	365	373
0C	014	034	214	024	074	052	045	100	306	315	324	344	352	360	366	374
0D	015	035	005	025	050	051	137	047	307	316	325	345	353	361	367	375
0E	016	036	006	236	053	073	076	075	310	317	326	346	354	362	370	376
0F	017	037	007	032	136	135	077	042	311	320	327	347	355	363	371	377

EBCDIC To ASCII As Recommended By Four-Phase Systems

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00	000	020	200	220	040	046	055	272	303	312	321	330	173	175	134	060
01	001	021	201	221	240	251	057	273	141	152	176	331	101	112	237	061
02	002	022	202	026	241	252	262	274	142	153	163	332	102	113	123	062
03	003	023	203	223	242	253	263	275	143	154	164	333	103	114	124	063
04	234	235	204	224	243	254	264	276	144	155	165	334	104	115	125	064
05	011	205	012	225	244	255	265	277	145	156	166	335	105	116	126	065
06	206	010	027	226	245	256	266	300	146	157	167	336	106	117	127	066
07	177	207	033	004	246	257	267	301	147	160	170	337	107	120	130	067
08	227	030	210	230	247	260	270	302	150	161	171	340	110	121	131	070
09	215	031	211	231	250	261	271	140	151	162	172	341	111	122	132	071
0A	216	222	212	232	133	135	174	072	304	313	322	342	350	356	364	372
0B	013	217	213	233	056	044	054	043	305	314	323	343	351	357	365	373
0C	014	034	214	024	074	052	045	100	306	315	324	344	352	360	366	374
0D	015	035	005	025	050	051	137	047	307	316	325	345	353	361	367	375
0E	016	036	006	236	053	073	076	075	310	317	326	346	354	362	370	376
0F	017	037	007	032	041	136	077	042	311	320	327	347	355	363	371	377

A592A

EBCDIC To ASCII As Recommended By ANSII X3.26-1970

COBOL and RPG Signed Numeric Fields

		ASCII	EBCDIC		EBCDIC
Value	ASCII	(Character)	Octal	Hex	(Character)
-0	0120	P	0175	7E	=
-1	0121	Q	0112		
-2	0122	R	0113		
-3	0123	S	0114		
-4	0124	T	0115		
-5	0125	U	0116		
-6	0126	V	0117		
-7	0127	W	0120		
-8	0130	X	0121		
-9	0131	Y	0122		
+0	0100	@	0173		
+1	0101	A	0101		
+2	0102	B	0102		
+3	0103	C	0103		
+4	0104	D	0104		
+5	0105	E	0105		
+6	0106	F	0106		
+7	0107	G	0107		
+8	0110	H	0110		
+9	0111	I	0111		

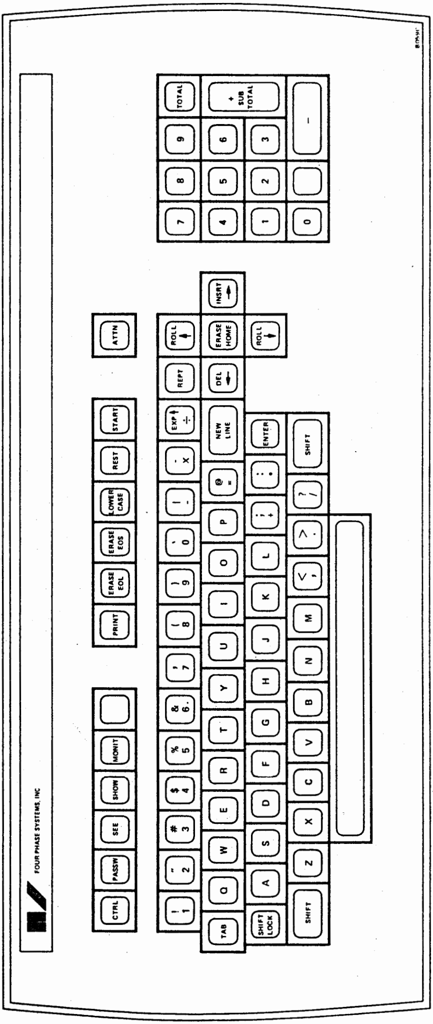
The signs for numeric data are attached to the rightmost byte of the data item. The format of this byte as as follows:

010X YYYY

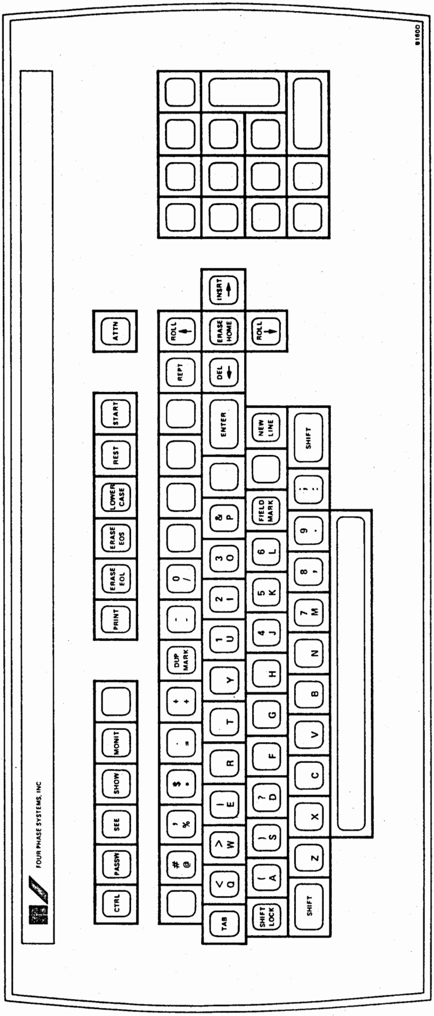
X: 0 = Positive
1 = Negative

Y: Ranges from 0-9

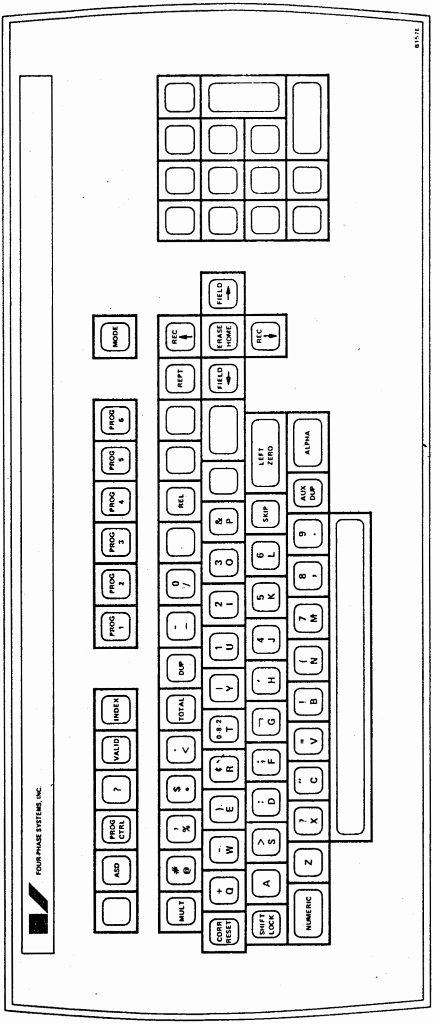
7201 KEYBOARD - 2260 TYPEWRITER STYLE



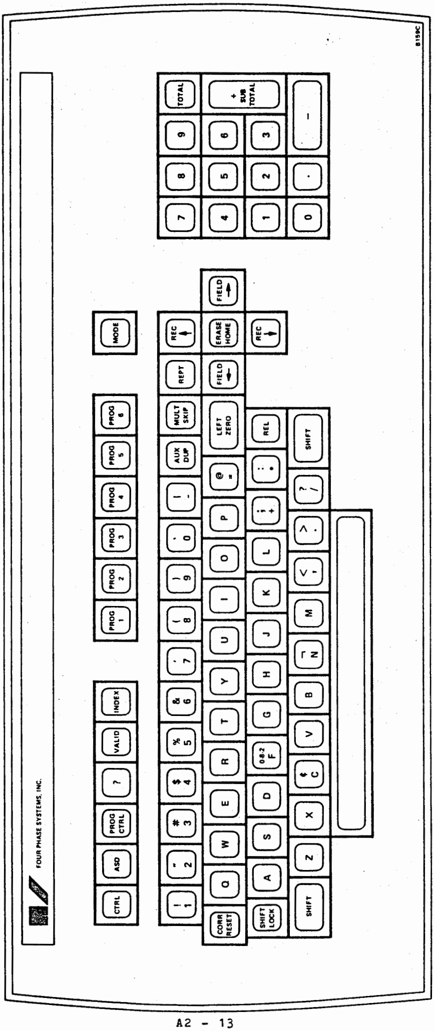
7204 KEYBOARD - 2260 DATA ENTRY STYLE



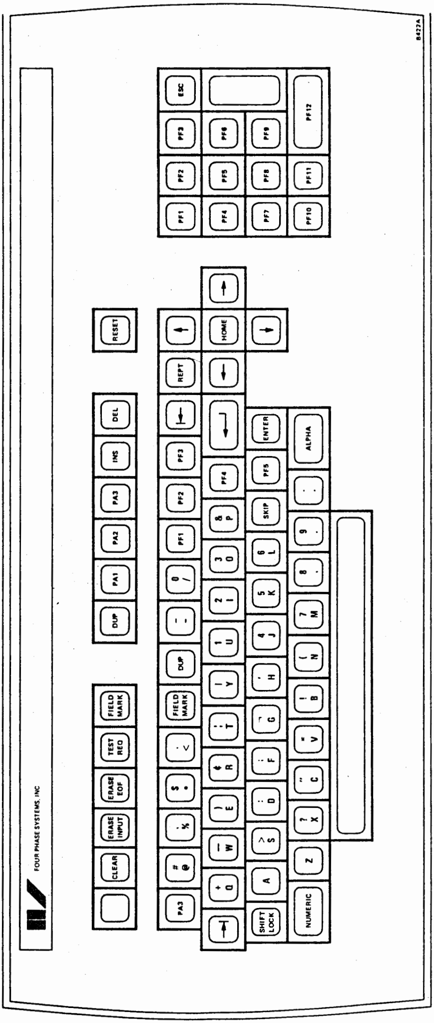
7202/7242 KEYBOARD - DATA IV KEYPUNCH STYLE



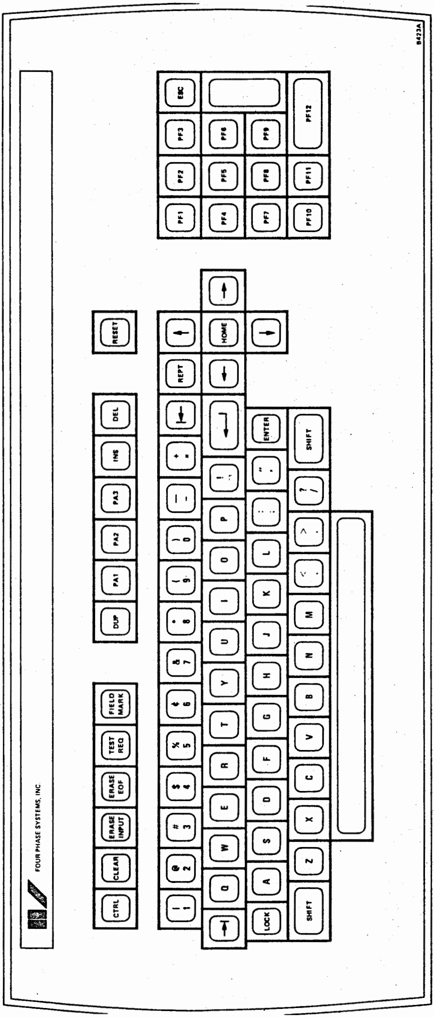
7203/7243 KEYBOARD - CHARACTER SET, DATA IV TYPEWRITER STYLE



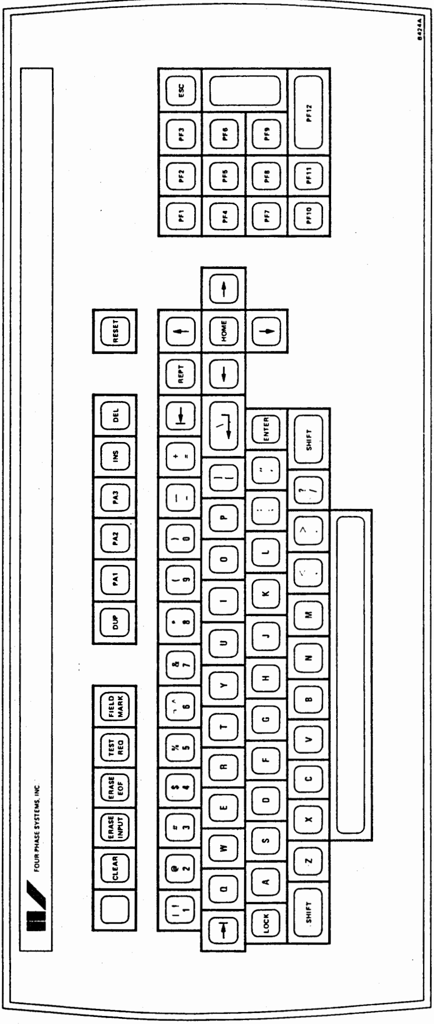
7225 KEYBOARD - NTP100,150,250 (3270) KEYPUNCH STYLE



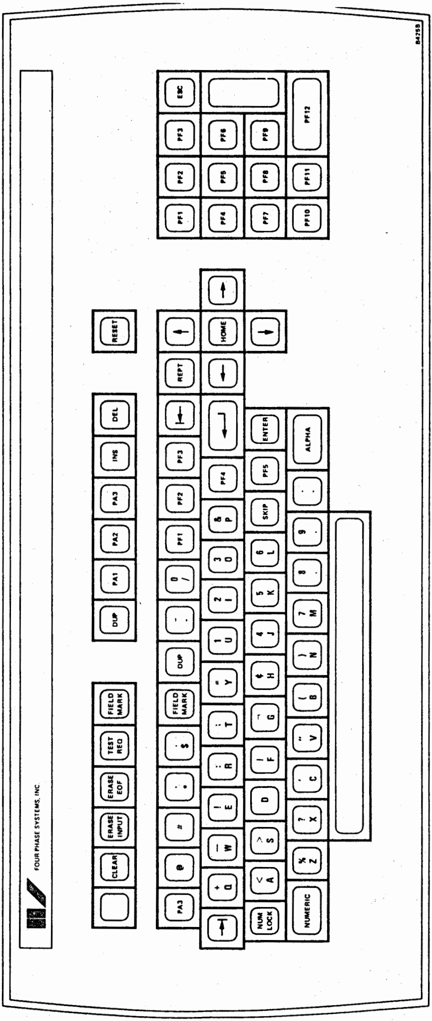
7226 KEYBOARD - NTP100,150,250 (3270) EBCDIC TYPEWRITER STYLE



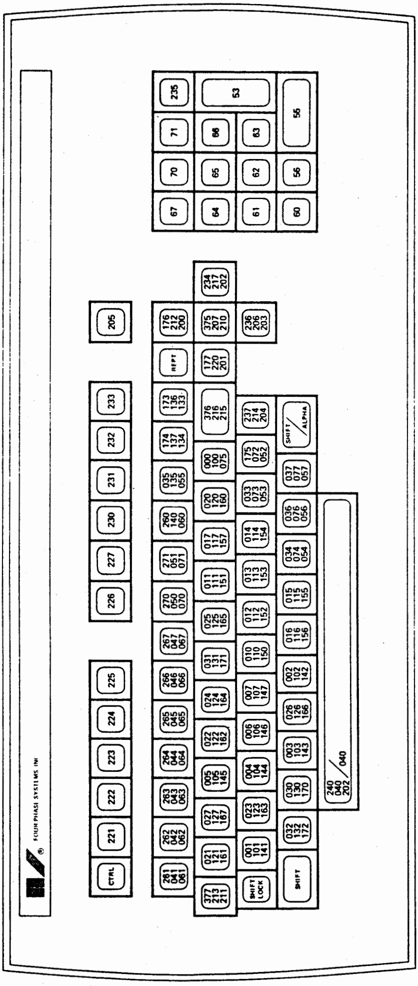
7227 KEYBOARD - NTP100,150,250 (3270) ASCII TYPEWRITER STYLE



7228 KEYBOARD - NTP100,150,250 (3270) DATA ENTRY STYLE

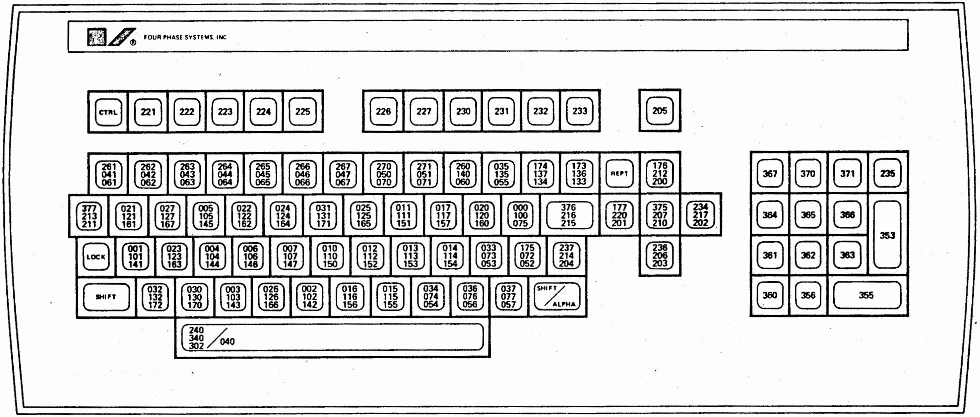


CODES GENERATED FOR 7200/7220/7240, 7201, 7202/7242, 7204 AND 7207



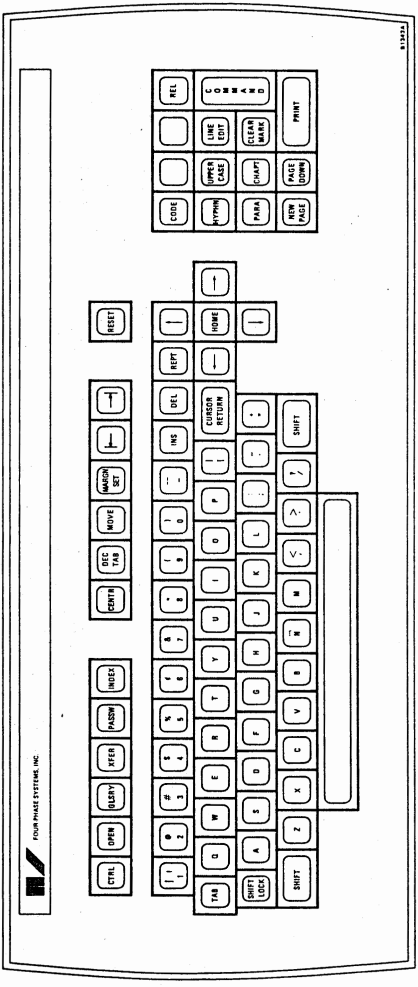
7200/7220/7240, 7201, 7202/7242, 7203/7243, 7204 and 7207 Keyboards

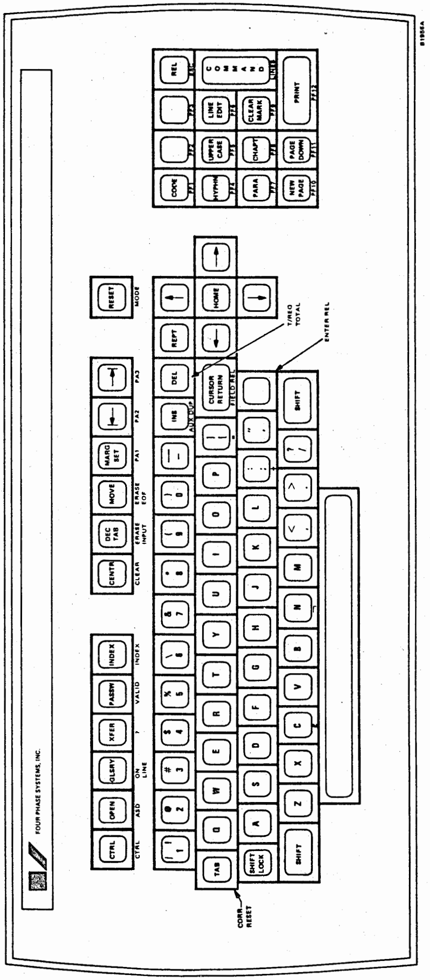
CODES GENERATED FOR 7225,7275,7226/7246,7227,7240,7251

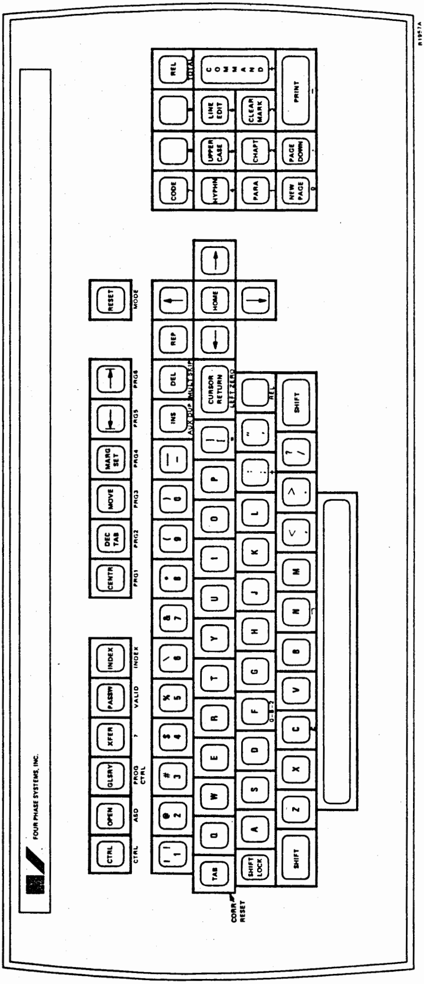


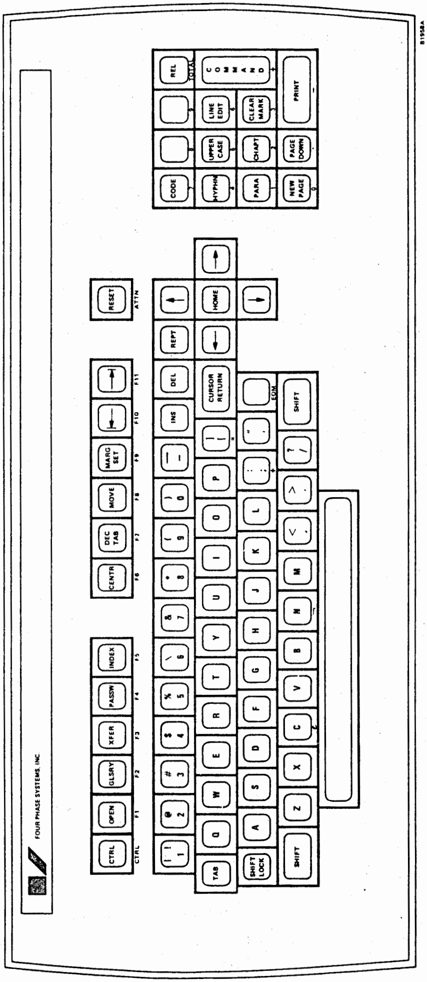
7225,7226/722 7225,7226/7246,7227,7228,7249,7250, and 7251 KEYBOARDS

ForeWord Keyboard




FOUR PHASE SYSTEMS, INC.
CTRL
OPEN
GLSRY
XFER
PASSW
INDEX
CTRL
ASD
ON LINE
?
VALID
INDEX
CENTR
DEC TAB
MOVE
MARG SET
CLEAR
ERASE INPUT
ERASE EOF
PA1
PA2
PA3
RESET
MODE
! 1
@ 2
# 3
$ 4
% 5
\ 6
& 7
* 8
(9
) 0
—
INS
DEL
REPT
TAB
Q
W
E
R
T
Y
U
I
O
P
CURSOR RETURN
AUX DUP
FIELD REL
SHIFT LOCK
A
S
D
F
G
H
J
K
L
: ;
" ,
SHIFT
Z
X
C
V
B
N
M
< ,
> .
? /
SHIFT
HOME
T/REQ TOTAL
ENTER REL
CORR RESET
CODE
REL
ESC
PF1
PF2
PF3
HYPHN
UPPER CASE
LINE EDIT
PF4
PF5
PF6
PARA
CHAPT
CLEAR MARK
PF7
PF8
PF9
NEW PAGE
PAGE DOWN
PRINT
PF10
PF11
PF12
COMMAND
LINE 5
81956A


FOUR PHASE SYSTEMS, INC.
CTRL
OPEN
GLSRY
XFER
PASSW
INDEX
CTRL
ASD
PROG CTRL
VALID
INDEX
CENTR
DEC TAB
MOVE
MARG SET
PRG1
PRG2
PRG3
PRG4
PRG5
PRG6
RESET
MODE
INS
DEL
REP
CURSOR RETURN
HOME
AUX DUP
MULT SKIP
LEFT ZERO
REL
SHIFT
SHIFT LOCK
TAB
CORR RESET
0-8-2
REL
TOTAL
COMMAND
CODE
HYPHN
PARA
NEW PAGE
UPPER CASE
CHAPT
PAGE DOWN
LINE EDIT
CLEAR MARK
PRINT
R1957A


81958A
FOUR PHASE SYSTEMS, INC.
CTRL
OPEN
GLSRY
XFER
PASSW
INDEX
CENTR
DEC TAB
MOVE
MARG SET
RESET
CTRL
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
ATTN
! 1
@ 2
# 3
$ 4
% 5
\ 6
& 7
* 8
(9
) 0
— -
INS
DEL
REPT
TAB
Q
W
E
R
T
Y
U
I
O
P
CURSOR RETURN
HOME
SHIFT LOCK
A
S
D
F
G
H
J
K
L
: ;
" '
EOM
SHIFT
Z
X
C
V
B
N
M
< ,
> .
? /
SHIFT
CODE
REL
TOTAL
HYPHN
UPPER CASE
LINE EDIT
COMMAND
PARA
CHAPT
CLEAR MARK
NEW PAGE
PAGE DOWN
PRINT
7
8
9
4
5
6
1
2
3
0
.
-
+

PARAMETERS FOR COMMONLY USED UTILITIES

```
***************************************************************************
***************************************************************************
***************************************************************************
```

```
                                                                | ALLOC  |
// ALLOC                     Allocate contiguous files or produce a map
/OUTPUT=name@drive           /O, /R, or /I is required
/COUNT=number                optional
/STARTING SECTOR             optional
/END                         optional (place at highgest possible address)
/LOADPOINT=address           optional; default is data file
/REPORT=CHK@drive            /R, /O, or /I required (checks for CRC errs)
/INPUT=name                  /I, /O, or /I required (write pattern to
                                                        existing file)
/PATTERN=xxx or n            optional; default is blanks
/BOUNDARY                    optional
                Blank record allows additional set of parameters
//                           Required.  Terminates parameter selection.
```

```
// ALMCLM               Create, mod, list Area Definition Table | ALMCLM |
/MODE=mode@phys dr      CREATE, MODIFY, or LIST
/NAME=4-char name
/AREA=number            0-127
/USER=x                 0-7 or ALL
/READ ONLY=YES or NO    optional, default is NO
/SECTORS=number         number of sectors or REST
```

```
// ALMCLX               Change the IDOS-UAD translation table   | ALMCLX |
/X=logical drive        IDOS drive index number
/USER=n                 CPU number, 0-3
/AREA=n                 area number or U for unassign
/Drive=n                physical drive number
/S                      soft boot option if changing area 0 (NU02)
//  (or blank line for more parameter sets)
```

```
// ALMCLZ               Create or delete dummy ..XX.. file      | ALMCLZ |
/L=logical drive #      create ..XX..  logical drive may not be 0
     or
/L=logical drive #,D    delete ..XX..  logical drive may not be 0
//
```

```
// ALMFMD                        Formatted NP/80 memory dump       | ALMFMD |
/EXTERNAL             optional, print external section only
/LOW                  optional, starting address for hex memory dump
/HIGH                 optional, ending address for hex memory dump
/MEMORY               optional, print memory section only
/SCREEN               optional, screen display only, see functions below
//
```

SCREEN function key descriptions:

F1	Scroll forward 1 line
F2	Scroll forward 12 lines
F3	Scroll forward 24 lines
F4	Scroll forward 100 lines
F6	Search for specified string
F7	Enter string for search
F8	Enter string for search start with "SECTION"
F9	Abort string search
P	Print screen (132 columns x 24 lines)
8	81 column screen type
4	48 column screen type
Left arrow	Left shift display 3 characters
Right arrow	Right shift display 3 characters
Home	Reset horizontal shift
ATTN	Return to IDOS

To execute Formatted Dump if the NP/80 halts with an MPE resident.

1. NP/80 - Note the major and minor codes if halt light on.
   a. the rightmost 6 console display lights are the major code
   b. set the AUTO/MANUAL switch to MANUAL
   c. select RAM(RPO) by setting the REG SELECT switches to 1000. The minor code is displayed in the 16 display lights.
   d. note the CUR STS register by setting the REG SELECT switches to 1010.
2. NP/80 - With the REG SELECT switches down and AUTO/MANUAL set to MANUAL, set data switch 1 up and all others down ($4000).
3. IV/70 - Reboot the IV/70 from a device other than the 8260.
4. NP/80 - Set the AUTO/MANUAL switch to AUTO. The NP/80 is now executing in the ROM code.
5. IV/70 - Enter // ALMFMD on the system keyboard as described above.

```
                                                                    ! ASGDEV !
// ASGDEV                     Assign logical device numbers
/LOGICAL=n,PHYSICAL=type@n    assign a disc; optional
(blank record)
/LOGICAL                      deassign a disc; optional
(blank record)
/Q=n,PHYSICAL=type@unit       assign a printer; optional (blank record)
/Q=n                          deassign a printer; optional
(blank record)
/SCREEN=file name@unit        screen 0 display; optional
(blank record)
/C                            change sign-on and/or autoboot
(sign-on message and/or
 name of boot file)
```

```
                                                                    !  ASM   !
// ASM                   Assemble source code
/INPUT=NAMEI@DRIVE.      Required
/OUTPUT=FILEO@DRIVE.     Required (recognizes RELOC also)
/XREF.                   Default = No XREF
/DCA SUPPRESS.           Default = No Suppression
/NOPRINT.                Default = Print
//                       Required.  Terminates parameter selection.
```

```
// BACK80                          Back up 8280                     ! BACK80 !
/PDRIVE=physical drive number      optional, default is 0
/RESTORE                           optional, default is BACKUP
/V                                 optional, include removable pack backup
/NAME=pack name                    required beginning with NU01-A
(user supplied data record
 or // )
```

```
// BACK90                     Back up 8290                          ! BACK90 !
/IDRIVE=physical drive #      default is 0
/ODRIVE=physical drive #      default is 1
/NAME=pack id                 required, name created by NPVOL
/RESTORE                      required for restore
//
```

```
                                                                    !  BOJ   !
// BOJ                   Return sectors not in use by files in directory
                         and check the integrity of chained files.
/DRIVE=logical drive#    optional; default is all assigned drives
//
(or blank line followed by additional parameters)
```

```
// CDDC                               Transfer card files to disk files
/OUTPUT=NAME@DRIVE.                   Required
/TYPE=OCTAL OR SOURCE OR RELOC.       Required
/SECTORS=NUMBER.                      Default=1
/PROTECT.                             Default=no
/END=3 CHARS.                         Default=Slash Slash Space
/MONITOR.                             Forces disc files to sectors 0-5
/CLEAR.                               /M and /C clears disc directory
Blank record allows additional sets of parameters
//                                    Required.  Termiates parameter selection.
```

Format of Octal Cards:

Each card is divided into ten 8 columns fields with the first field being divided into 2 fields of 3 and 5 columns, respectively.

Field 1 (1-3) = Program identification (checked for consistency)
Field 2 (4-8) = Card sequence # (must be ascending)
Field 3-10 = Can be any of four types identified by the first column contents as follows:

0-9= Data field. The 8 columns contain an instruction or data which would be loaded at the current location counter address.

& = Origin field. The remaining 7 columns contain a number to which the current location counter is set.

- = Transfer field. The remaining 7 columns contain the address to which control is transferred after all object code is loaded. Any remaining fields on the card are skipped.

Anything Else = Checksum field. Use to check validity of preceeding cards. Must be last field on the card. If no checksum field is given, no check will be made.

The following example will cause a 30 sector contiguous field named "TEST" to be allocated:

```
// CDDC
/OUTPUT=TEST,TYPE=O,SECTORS=30.
ABC00000&0000000-0000000
//
```

```
                                                              ! COPY   !
// COPY                   Copy files between discs and/or tapes
/INPUT=NAMEI@drive.       Default = all files
/OUTPUT=NAME@drive.       Default = same file(s) as input
/FORCE.                   Optional - force file to start of cylinder
/QSWITCH.                 Optional - file compare instead of copy
/JINPUT=NAMEJ@drive.      Replaces /I for tape input
/POUTPUT=NAMEP@drive.     Replaced /O for tape output
/TAPE=(TAPE8/TAPE16/TAPE7).         Required for tape
/REWIND=@drive.           Optional
/SKIP=number@drive.       Optional - advance n tape files
/HEADER.                  Optional - data card at end of parameter string
                                     becomes tape header information
Blank record will allow additional sets of parameters (or is header)
//
```

```
                                                              ! COPYA !
// COPYA                            Copy virtual disc areas
/INPUT=area@physical drive          required, area is name or number
/OUTPUT=area@physical drive         required, area is name or number
/BATCH                              optional, don't halt after execution
/ALL                                optional, whole disk copied
/SOURCE=xxxxxx                      required with ALL, pack id
/DESTINATION=xxxxxx                 required with ALL, pack id
 (prompt line)
       or
// (prompt line)
```

```
                                                              ! COPY01 !
// COPY01                 Copy all data from 8231 or 8271 disc to another
/INPUT=drive              optional; default is 0
/OUTPUT=drive             optional; default is 1
/COMPARE ONLY             optional
/DUPLICATE ONLY           optional
/HALT INHIBITED           optional
/ABORT                    optional; stop when console switch 0 up
```

```
// CRTDMP                 Display disk or memory on screen.     ! CRTDMP !
/SECTOR=n@d        CURSOR RETURN twice;  /S, /I, or /M required.
/MEMORY=n.         CURSOR RETURN twice;  /M, /I, or /S required.
/INPUT=filename    CURSOR RETURN twice;  /I, /M, or /S required.

Control keys:
    ARROW keys            position cursor
    Shifted up arrow      next quarter sector or memory block
    0-7                   enter octal character
    CTRL up arrow         forward one full sector
    CTRL down arrow       back one full sector
    Shifted EOM           write screen to sector/memory
    F9                    back one in chained file
    F10                   forward one in chained file
    F11                   print sector or memory display via $DUMP
    ATTN                  select a new address
                            memory or 8230   exactly 5 digits octal
                            all others       exactly 6 digits octal
    Shifted down arrow    return to $BATCH (don't use if you've altered
                                             COMM region)
```

```
                                                                     ! DIRDMP !
// DIRDMP                    Print contents of disc directory
/DRIVE=n                     optional; default is all assigned drives
/NUMBER=n                    optional; start print with specified file number
//
```

```
                                                                     ! DIRMOD !
// DIRMOD                    Modify or delete directory entries
/INPUT=name@drive.           required
/OUTPUT=name@drive.          If not given, INPUT file will be deleted
/PROTECT.                    optional
/Q.                          optional, unprotect INPUT file.
/FLAG=X.                     optional
/LOAD=address.               optional
/5=value.                    optional, enter value in X+5           (BJ04)
/6=value                     optional, enter value in X+6           (BJ04)
/CPUID=3-bit value           optional                               (BJ04)
/R-LOCKED-OUT-BIT            optional, set or clear locked out bit  (BJ04)
/J-INPUT-IN-DATA-CARD        optional                               (BJ04)
Blank record will allow additional set of parameters
//
```

```
                                                                     ! DIRSRT !
// DIRSRT                    Sort disc directory entries
/DRIVE=n.                    optional, default = 0
/REVERSE                     optional, default is ascending order
/ENTRY=n                     optional, entry number to swap with /F (BJ04)
/FILE=filename               optional, swap positions with /E file  (BJ04)
//
```

```
                                                                     ! DIRVID !
// DIRVID                    Display directory information on Screen 0
/DRIVE=n                     optional, default is 0
/I=filename@n                optional, displays details             (BJ04)
//
    CONTROLS:
         F1         scroll forward one line
         F2         scroll forward page minus 1 line
         F6         scroll back 1 line
         F7         scroll back page minus 1 line
         F11        print the /I file information                   (BJ04)
      UP ARROW-S    return to start of directory
       ATTN         exit
```

```
                                                                     ! DTUX !
// DTUX                                         Dump all or part of a disc to tape
/INPUT=DISC@drive.                              required
/OUTPUT=TAPE8/TAPE16/TAPE7@deck.                required
/LOW=NUMBER.                                    optional, default is 0
/HIGH=NUMBER.                                   optional, default depends on device
Blank record will allow additional sets of parameters
//
```

```
                                                                 ! FILDMP !
// FILDMP                   Dump disc files to the system printer
/INPUT=name@drive or xx yyyyyy@drive   where x is count, y is 1st sector
/A=number or /A             optional, start count with number or load point
/EJECT                      optional
/CONDENSE                   optional
/BASE=OCTAL or HEX          optional, default is OCTAL
/DISPLAY=ASCII or EBCDIC optional, default is ASCII
/LENGTH=16 or 24            optional, default is 24                (BJ04)
//
```

```
                                                                   ! FMTX !
// FMTX                    (BJ04)
/MODE=mode name            FORMAT, CHECK, VALIDATE, REDIRECT, or HEADER
/DRIVE=type@phys drive     default is 8270@0. Allowed: 6270,8230,8240,8270
/LINE-PRINTER=type         required when booting from tape (Pxxxx or NOPRNT)
/SECTOR=yyyyyy             required for REDIRECT optional for others
/PASSES=number             optional, default is eternity
/RETRIES=number            optional
/VALUE=number              optional, default is 025252525
/AUTO-REDIRECT             optional, eliminates operator-intervention pause
//
```

```
                                                                 ! GENCTR !
// GENCTR                  Utility for control file maintenance
/OUTPUT=name@drive.        required
/D=xd@drive.               char string source, default /D=0@0
/END=xe.                   end card identifier, default /E=//
/C=xc.                     filler record identifier, default all blanks
/FLAG=xf.                  output file flag, default none
/G=xg.                     flag select, optional, only used with /D=0
/H=xh.                     header/trailer-record identifier default: //
/S=ss                      end header-record identifier, optional
/A=xa.                     match char, default /A=0-2-8 punch
/B=xb.                     mismatch char, default /B=0-5-8 punch
//
.
. Model Statements
.
. End Statement
. Character strings of candidate data (only if /D=1)
.
. End Statement
For example, to copy all files except $BATCH from drive 0 to drive 1:
    // GENCTR
    /0=TEMP,A=#,B=%,E=$$,D=0.
    // COPY
    /I=%MONITR,0=#@1.
    //
    $$
    // TEMP1
```

```
                                                                  ¦  JOB  ¦
// JOB                  Deallocate sectors held by blank or TEMP filenames
/D=logical drive        optional, default is all assigned drives
//
```

```
// KCHEK                Verify consistency of an MKAM file           ¦ KCHEK ¦
/AREA=number            required
/DRIVE=physical drive   required
/ZONESET=name           required; 3 char name
/FILE=filename          required; 3 char name
/CHECKALL               required if KEY not given; verify all keys
/KEY=key name           required if /C not given
          (subsequent /KEY's are preceded by a blank line)
//
```

```
// KCHK2                Verify MKAM zonesets after NP/80 crash       ¦ KCHK2 ¦
/ZONESET=name           required, 3 bytes
/AREA=number            required
/DRIVE=physical drive   required
/S                      optional, messages to screen instead of printer
//
                        key functions are same as for ALMFMD
```

```
// KDVID                List or display a zoneset's files            ¦ KDVID ¦
/ZONESET=name           required
/AREA=number            required
/DRIVE=number           required
/FILE=filename          optional, display this file only
/SCREEN                 optional, information to screen instead of printer
/LIST=file name@number  optional, create an IDOS file with MKAM data list
//                      key functions are same as for ALMFMD
```

```
// KFILE                     Create, delete and purge MKAM files ¦ KFILE ¦
1      8      15     22      (parameters must be aligned to these columns)
¦      ¦      ¦      ¦
OPER   op                    required, CREATE, DELETE, or PURGE
AREA   anum                  required
DRIVE  dnum                  required, physical
ZONSET zname                 required
FILE   flname                required
VERS   vname                 optional, version name, default 3 blanks
KEY    kname  NODUPS         required
              fdname         required
FIELD  fdname sc     ec      required, starting and ending columns
END                          optional, used only with CREATE
```

```
                                                                  ¦  KMOD  ¦
// KMOD                 View and modify physical, logical, or MKAM sectors
/ZONESET=zone name      optional, default is physical and IDOS sectors
/DRIVE=physical #       default is drive 0
/AREA=area number       default is area 0, irrelevant for physical sectors
//
Please see NPOS Utilities Manual (NU02).  To view a sector, press ATTN
until cursor appears below LOGICAL Z/D, enter P (physical), or I (IDOS),
then enter sector #.  Use % for hex, 0 for octal or non-0 for decimal.
Shifted and CTRL arrows move display, CTRL EOM writes to disk.
```

```
// KTOSD                     Copy records betw MKAM and SD files ¦ KTOSD ¦
/A=area number               required
/D=drive number              required
/Z=zoneset name              required
/F=file name                 required
/K=key name                  required to copy MKAM to SD
/O=name@logical drive        required to copy MKAM to SD
/S=direction                 A or D, optional for MKAM to SD
/I=name@logical drive        required to copy SD to MKAM
/UNFORMATTED                 optional
/BYTESAVE                    optional
//
```

```
// KZSET                     Create a zoneset                    ! KZSET !
/CREATE                      optional, required if zoneset doesn't exist
/AREA=number                 required
/DRIVE=number                required
/ZONESET=name                required
/SIZE=size                   optional, default 4032 sectors or max avail
//
```

```
                                                                  ! LIST !
// LIST  (BJ04)                   Print control and source files
/INPUT=FILE@drive.                required
/EJECT=pages before@pages after   optional
/START=line number                optional
    or=label                      optional, 1st 6 characters
    or=label@column               optional, 6 characters beginning at col
/FINISH=line number               optional, defines end of print
     or=label
     or=label@column
/NO-DON'T-PRINT-OPTION            optional, suppress printing of all
//                                          OPTION JCL within the file
                                            being listed
```

```
                                                                ! LOADOV !
// LOADOV                  Program load with overlay capability
/INPUT=file@drive.         required
/INPUT=filen@DRIVE.        optional, up to 151 inputs
/POSITION OF LOADER=LO.    default = HIGH
/LIBRARY=libraryname.      optional
/UNRESOLVED VIRTUALS OK.   optional, default is not ok
/HALT BEFORE ABORT         optional
/OUTPUT=file@drive.        optional in main, illegal higher modules
/MODULE=NAME.              not in main, required in higher modules
/ROOT=rootname.            not in main
/BOTTOM OF MODULE=name or address      optional
/TOP OF MODULE=name or address         optional
/EXECUTE=name or address.              optional
/SORT=(ALPHA/NUMBER/BOTH).             default = NO SORT
//
```

```
--------------------------------------------------------------------------
                                                                ! LUCONV !
// LUCONV               SYSIN lower/upper case toggle
/UPPER                  optional, specifies upper case
/LOWER                  optional, specifies lower case
/PRINT=UPPER or LOWER   optional, converts lower to upper for print (BJ04)
//
```

```
// NPDTUX                         Disk to tape                        ¦ NPDTUX ¦
/INPUT=DISC@physical drive        required
/OUTPUT=TAPEx@deck                required, x = 7, 8, or 16
/LOW=n                            optional, low physical address of first range
/HIGH=n                           optional, high physical address of first range
.
.//
```

Files on NPDTUX tape: Bootable loader, TPMON, NPTBMP, L00, NPTFX2, NPDTUX.

Tape to disk

1. Boot from the NPDTUX tape.
2. Enter // NPTBMP to load an MPE into NP/80 memory. Follow instructions on screen.
3. Rewind and reboot the tape.
4. If the pack needs to be formatted use NPTFX2 (NPTFX1 1st for pre-NU02).
5. Enter // NPDTUX. Enter // to accept parameters from tape which are displayed on screen, else override with options as above.

```
                                                                      ¦ NPFMTX ¦
// NPFMTX                                Format an NP/80 supported disk
/DRIVE=physical device number
/STAGGER=number                          9 is standard.
/RETRY=number                            1 to 9
/MODE=mode                               FORMAT, VALIDATE, CHECK, or REDIRECT
/F                                       8280 fixed portion  (NU01)
/T=1                                     8290 only  (NU02)
//
```

NPTFX1 AND NPTFX2 - Format an NP/80 supported pack from tape

1. Boot from an NPDTUX tape
2. Enter // NPTFX1. Follow screen prompts to load NP/80 memory.
3. Reboot from tape without resetting NP/80 memory.
4. Enter // NPTFX2 and use options from NPFMTX above.

Note: NPTFX1 disappears, and NPTFX2 assumes the task or loading the NP/80, beginning with release NU02.

```
// NPVOL       Initialize pack id and volume sequence number           ¦ NPVOL ¦
/PDRIVE=physical drive number
/NAME=pack name
/VOL=disk sequence number
/SCREEN        may be used instead of /P, /N, and /V for keyboard entry
//
```

```
                                                                      ¦ RDTAPE ¦
// RDTAPE or TRDTAP
/TAPE=TAPE7, TAPE8, or TAPE16               required if booted from disk
/DISC=8230,8240,8260,or8270@phys dr         required if booted from tape
/INCLUSIVE                                  I or E required
/EXCLUSIVE                                  E or I required
/ALL SELECTED FILES TO DISC                 A, O, or N required
/OLD SELECTED FILES TO DISC                 A, O, or N required
/NEW SELEDTED FILES TO DISC                 A, O, or N required
/FILE=name                                  optional
/CATEGORY=x                                 optional
/MESSAGE                                    optional
(message record less than 80 characters or blank if no /M)
.
.
//
----------------------------------------------------------------------------
```

! SIMED !

```
// SIMED                      Create, view, and edit source files
                              If no IFILE is specified, OFILE will be created,
                              If no OFILE, changes cannot be made to IFILE,
                              If a blinking message appears, ATTN will reset it.
                              ATTN - acknowledge flashing message
                                     cause the remainder of the file to be output
                                     enter IFILE and OFILE names.
                                     exit to $BATCH
                              Arrows - move cursor
                               F1 - scroll 1 line
                               F2 - scroll 1/2 display
                               F3 - scroll full display
                               F4 - duplicate line that cursor is in
                               F5 - upper/lower case toggle
                               F6 - abort - no output
                               F7 - insert a line of text if not in create state
                              F11 - print the screen
                             HOME - cursor to upper left corner
                              TAB - cursor to next tab stop
                        SHIFT TAB - cursor to previous tab stop
                         CTRL TAB - set/clear tab stops
                SHIFT RIGHT ARROW - insert character
                 SHIFT LEFT ARROW - delete character
                 CTRL RIGHT ARROW - insert record
                  CTRL LEFT ARROW - delete record
                 SHIFT DOWN ARROW - write the output file and display it

                        CTRL HOME - **MULTI-STEP COMMAND (EXIT:ATTN)
                                    enter #nnnnnn SINDSK number to be searched
                                    for, and press CURSOR RETURN to start
                                    search.  # without a number moves to end of
                                    file.
```

! SINDSK !

```
// SINDSK or SNEDIT           Source file maintenance
/OUTPUT=name@drive. required
/INPUT=name@drive. optional
/END=xxx.                     optional
//ALTER=x.                    optional
/FLAG=x.                      optional
/MERGE CHARACTER=x.           optional
/PROTECT.                     optional
/RENUMBER=x.                  optional
/DELTA=n.                     optional
/LIST.                        optional
.
. SYSIN stream
.
xxx
//
```

! WINDOW !

```
// WINDOW Display source and SPOOL files on screen 0
/INPUT=name@drive.                      default=$SPOOL@0
/HEIGHT=6,12, or 24                     default=12
/WIDTH=(48 or 81)                       default=48
/STARTING LOCATION=address              default tied to /W
/TAB=number                             default=15 characters
/RETURN=number                          default=full line
/COUNTER DISPLAY=NO.                    default=display the counter
/MARGINS=number                         default=133
/FORM SEPARATORS=NO.                    default=display dashes between page
     Blank record allows additional sets of parameters
//
```

The following keys have the indicated functions:

```
    Up and Down Arrows move display up or down 1 line.
    Right and Left Arrows move display right or left 1 char.
    TAB moves the screen left /T=number characters.
    CR moves the screen right /R=number characters.
    F1 moves the file down 100 lines.
    F2 moves the file down 50 lines.
    F3 moves the file down 10 lines.
    F4 moves the file up 10 lines.
    F5 moves the file up 50 lines.
    F6 moves the file up 100 lines.
    F10 restarts the file at the beginning.
    F11/ATTN returns to take more parameters or to monitor
```

! WRTAPE !

```
// WRTAPE                  Transfer files from disc to tape
/TAPE=TAPEx                x=7,8 or 16                        (required)
/DISC=@logical drive                                          (required)
/SIZE OF TARGET MACHINE = 48 or 72                            (optional)
/FILE=name
/CATEGORY=x                                                   (optional)
/MESSAGE.                                                     (optional)
(message record less than 80 characters or blank record if no /M)
.
.
//
```

```
*******************************************************************************
****************************** DKOS PROGRAMS **********************************
*******************************************************************************

_____________________________________________________________________________
                                                                  | CAU-CA |
// CAU-CA          Assembles CPU diagnostic for non-decimal CPU's

_____________________________________________________________________________
                                                                  | CAU-CB |
// CAU-CB          Assembles CPU diagnostic for decimal chip CPU's

_____________________________________________________________________________
                                                                  | CAU-R  |
// CAU-R           Asembles RAM diagnostic

_____________________________________________________________________________
                                                                  | COPYF  |
// COPYF           Copies/deletes/renames files on diskette
/INPUT=NAMEI.           Required
/OUTPUT=NAMEO.          If /O, the file will be deleted
/MONITR.                Optional - causes write to track 0
/PROTECT.               Optional - make nameo protected
/QUASH protection,      Optional - unprotect nameo
/B. or /U=FILENAME.     Optional - Auto Boot program name
/U. or /U=DIAGNOSTIC.   Optional - U=0 means no mini-CPU diagnostic.
U=D                     means CPU diagnostic w/decimal
U.                      means CPU diagnostic w/o decimal
/R=0.                   Optional - means no mini-RAM diagnostic.
/R.                     Optional - means mini-RAM diagnostic during boot
        Blank record for additional sets of parameters
//                      Required.  Terminates parameter selection.

_____________________________________________________________________________
                                                                  | DCDKT  |
// DCDKT                Copy all or part of a DIABLO disc onto diskettes
/LOW sector=number      Default=0
/HIGH sector=number     Default=06177
/DRIVE=number.          Default=0
/COMPLETE backup.       Default=Allocated sectors only
//

_____________________________________________________________________________
                                                                  | DIRDSP |
// DIRDSP               Display on tube 0 or print the diskette directory
/SIZE=80,               Required if NOPRNT & 80 character tubes
//                      Required if NOPRNT

_____________________________________________________________________________
                                                                  | DKTGEN |
// DKTGEN          Transfer utility programs to diskette from DKOS source
                     master disc.
                   FMONTR should already be on diskette
                   CAU-CA or CAU-CB should already be run
```

```
                                                                 ¦ FLCOPY ¦
// FLCOPY                 Transfer files to/from disc and diskette
/INPUT=NAMEI@DRIVE.       Required
/OUTPUT=NAMEO@DRIVE.      Required, if /O and to diskette NAMI will be
/.                        written to track 0
/DESTINATION.             Default - diskette.  DIA=DOS pack, IDOS=IDOS pack
/MONITR.                  Optional, causes write to track 0,
/.                          /D=diskette only.
/CLEAR.                   Optional, use with care, to diskette only
/SIZE=(0/24/48/72/96).    Only used with /C, default=96, /D=diskette only.
/ADD checkpoint=address   Optional, default=no, /A. Puts checkpoint at
/.                          max. loc, /D=diskette only
/PROTECT.                 Default - nameo no protected
/QUASH protection.        Optional
/BOOT programe=nameb.     Optional - /B = make not auto-book pack
/JCL.                     Optional - make output a JCL file.
/U. or /U=Diagnostic.
/R. or /R=Diagnostic.     Optional - SEE COPYFBlank record for additional
parameters
//                        Required.  Terminates parameter selection.
```

```
                                                                 ¦ FPYDEL ¦
// FPYDEL          Delete DOS source files to provide space for DKOS
```

```
                                                                 ¦ LDFLCP ¦
// LDFLCP          Updates DKOS portion of DKOS source disc to latest level
                     of DOS/IDOS.  Run after SYSGEN.
```

```
                                                                 ¦ LFCTL  ¦
// LFCTL           Produce a list of all DKOS control files
```

```
                                                                 ¦ MKDSKT ¦
// MKDSKT          Assemble DKOS monitor and utilities
```

```
                                                                 ¦ PACK   ¦
// PACK            Return all tracks of deleted files and consolidated
                     unused tracks
```

```
                                                                 ¦ PATCH  ¦
// PATCH                        Modify files stored on diskette
/WIDTH=(48/81).                 Required
/HEIGHT=(6/12/24).              Required
/SCREEN SIZE.                   Optional, flush any changes & restart
/INPUT=(program name/track)     Required
/LOCATION=number.               Default=0, if /A. number=RAM address
/.                                Else=word on track
/VALUE=number.                  Value that word at location will assume
/.                                only valid with /M.
/DISPLAY.                       Display mode, no values allowed
/MODIFY.                        Modify mode - be careful
/ABSOLUTE.                      Used with /I=PROG name to indicate
/.                                RAM address on left
/FLUSH.                         Flush all modifications to diskette
/.                                done automatically at exit or new /L
//                              Required.  Teminates parameter selection.
```

```
// XDSKT        Transfer all DKOS files from Diablo to Diablo    ¦ XDSKT ¦
```

FOUR-PHASE COBOL DIFFERENCES

Four-Phase COBOL includes the following modules from ANSI COBOL '68: Level 1 Nucleus, Sequential Access, Random Access, and Library; Level 2 Table Handling. The most notable features lacking in these modules but found in higher level modules are:

1. The COMPUTE verb.
2. Nested if statements (including AT END, SIZE ERROR, etc.).
3. Data name qualification.
4. Multiple filenames on OPEN and CLOSE.
5. Compound relational expressions with AND or OR.
6. READ...INTO and WRITE...FROM.
7. OCCURS...DEPENDING ON.
8. MOVE...CORRESPONDING.
9. PERFORM...VARYING or ...WHILE

Several extensions common to IBM COBOL but not found in Four-Phase COBOL are:

1. COMPUTATIONAL-3 (packed decimal) data.
2. RECORDING MODE.

Common maximums in Four-Phase COBOL:

1. OCCURS, 511.
2. Literal length, 120 bytes.
3. DISPLAY pic 9 size, 18.
4. Subscripts, 3.

Common industry extensions found in Four-Phase COBOL:

1. LINKAGE SECTION.
2. CALL...USING.
3. ISAM.

IF THE COMPILER HALTS OR LOOPS...

1. Check the environment for things that might cause any IDOS program to misbehave.
   Printer off-line
   Card reader error halt while reading options
   Memory parity
   IDOS has miscomputed memory size (BOOT system, check location 0712. The symptom is that the value here is one greater than the true size of memory.)
   Cut word hang because of missing or wrong printer in LPOUT.

2. Take a memory dump to printer:

```
MANUAL
72000002 into TIR
AUTO
```

   OR return to IDOS

```
MANUAL
72000001 into TIR
AUTO
```

DO NOT simply halt the computer and remove the disc or reboot. You will lose sectors from the allocation table.

3. Print compiler temporary files by executing // CBLDMP.

4. If submitting compiler problems on an SER, please include PROCEDURE and DATA DIVISION Load maps.

IF THE OBJECT PROGRAM HALTS OR LOOPS...

It may be the user's problem. A sprinkling of DISPLAY or STOP statements can be used to locate the problem. If necessary a dump may be taken by 1 of 3 methods:

1. A memory dump can be taken with the standalone 3-card dump program if you have a line printer and a card reader.

2. The object program will initialize Location 1 to be a branch to a routine which will dump RAM on SYSOUT, close all files, and return to IDOS. The $DUMP routine wil be eligible for exclusion with LIBGEN. If $DUMP is not excluded, this mechanism will allow the user to obtain a memory dump manually in situations where the /! checkpoint mechanism is inappropriate.

```
AUTO TO MANUAL
720000001 INTO TIR
MANUAL TO AUTO
```

   For releases E1 and below.

   A copy of $DUMP can be linked to the object program in the LOADOV step. One way to do this is to put a CALL "$DUMP" in the source code. This statement should never actually be executed, since the COBOL-generated calling sequence is incompatible with $DUMP. Instead, do a manual 710nnnnn when the problem occurs. (where nnnnr is the load map address of $DUMP).

IF THE OBJECT PROGRAM HALTS OR LOOPS - CONTINUED

3. Memory can be dumped to disc and later printed. Execute the COBOL program with the option card /!=filename. When reading to take the dump:

```
AUTO TO MANUAL (DO NOT hit SYSTEM RESET)
720nnnnn       (where nnnnn is the load map address of = STOP)
```

List the dump using:

```
// FILDMP
/I=filename,A   (Use the A option so the listing will be numbered
                 by memory address.)
```

Use the "Relocatable Module Map" section of the compiler output listing to interpret the dump. Remember that addresses in the procedure division map must be relocated by the value of the BOTTOM parameter in the LOADOV step. The Data Division map shows the relative locations of items defined in the Data Division. The Procedure Division map can be used to locate the code generated for a statement.

COBOL RESERVED WORDS

ACCEPT
ACCESS
ACTUAL
ADD
ADDRESS
ADVANCING
AFTER
ALL
ALPHABETIC
ALTER
ALTERNATE
AND
APPLY *
ARE
AREA
AREAS
ASCENDING
ASSIGN
AT
AUTHOR
BEFORE
BEGINNING
BLANK
BLOCK
BY
CALL *
CF
CH
CHARACTERS
CLOCK-UNITS
CLOSE
COBOL
CODE
COLUMN
COMMA
COMP
COMPUTATIONAL
COMPUTE
CONFIGURATION
CONTAINS
CONTROL
CONTROLS
COPY
CORR
CORRESPONDING
CURRENCY
DATA
DATE-COMPILED
DATE-WRITTEN
DE
DECIMAL-POINT
DECLARATIVES
DELETE
DEPENDING
DESCENDING
DETAIL
DISPLAY
DIVIDE
DIVISION
DOWN
ELSE
END
ENDING
ENTER
ENVIRONMENT
EQUAL
ERROR
EVERY
EXAMINE
EXIT
FD
FILE
FILE-CONTROL
FILE-LIMIT
FILE-LIMITS
FILLER
FINAL
FIRST
FOOTING
FOR
FOUR-70 *
FROM
GENERATE
GIVING
GO
GREATER
GROUP
HEADING
HIGH-VALUE
HIGH-VALUES
I-O
I-O-CONTROL
IDENTIFICATION
IF
IN
INDEX
INDEXED
INDICATE
INITIATE
INPUT
INPUT-OUTPUT
INSTALLATION
INTO
INVALID
IS
JUST
JUSTIFIED
KEY
KEYBOARD *
KEY-IN *
KEYS
LABEL
LAST
LEADING
LEFT
LESS
LIMIT
LIMITS
LINE
LINE-COUNTER
LINES
LINKAGE *
LOCK
LOW-VALUE
LOW-VALUES
MEMORY
MODE
MODULES
MOVE
MULTIPLE
MULTIPLY
NEGATIVE
NEXT
NO
NOMINAL
NOT
NOTE
NUMBER
NUMERIC
OBJECT-COMPUTER
OCCURS
OF
OFF
OMITTED
ON
OPEN
OPTIONAL
OR
OUTPUT
PAGE
PAGE-COUNTER
PERFORM
PF
PH
PIC
PICTURE
PLUS
POS **
POSITION **
POSITIVE
PROCEDURE
PROCEED
PROCESSING
PROGRAM-ID
QUOTE
QUOTES
RANDOM
RD
READ
RECORD
RECORDS
REDEFINES
REFL
RELEASE
REMAINDER
REMARKS
RENAMES
REPLACING
REPORT
REPORTING
REPORTS
RERUN
RESERVE
RESET
RETURN
REVERSED
REWIND
REWRITE
RF
RH
RIGHT
ROUNDED
RUN
SAME
SCREEN *
SD
SEARCH
SECONDARY
SECTION
SECURITY
SEEK
SEGMENT
SEGMENT-LIMIT
SELECT
SENTENCE
SEQUENTIAL
SET
SIGN
SIZE
SORT
SOURCE
SOURCE-COMPUTE
SPACE
SPACES
SPECIAL-NAMES
STANDARD
START
STATUS
STOP
SUBTRACT
SUM
SYNC
SYNCHRONIZED
TALLY
TALLYING
TAPE
TERMINATE
THAN
THROUGH
THRU
TIMES
TO
TYPE
UNIT
UNTIL
UP
UPON
USAGE
USE
USING **
VALUE
VALUES
VARYING
WHEN
WITH
WORDS
WORKING-STORA
WRITE
ZERO
ZEROES
ZEROS

* Four-Phase extensions to ANSI standard usage.
** As used in this compiler, this word is an extention to COBOL '68 standard.

COBOL LOW MEMORY ALLOCATION

Screen #	User Table Address	(81 x 24) Screen Location
1	01540-01567	0140-01537
2	01570-01617	02140-03537
3	01620-01647	04140-05537
4	01650-01677	06140-07537
5	01700-01727	010140-011537
6	01730-01757	012140-013537
7	01760-02007	014140-015537
8	02010-02037	016140-017537
9	02040-02067	020140-021537
10	02070-02117	022140-023537
11	03540-03567	024140-025537
12	03570-03617	026140-027537
13	03620-03647	030140-031537
14	03650-03677	032140-033537
15	03700-03727	034140-035537
16	03730-03757	036140-037537

COBOL computes the location of User Tables and the IOID at object time. See Routine :KEYI in P710F. Each User Table is 24 (030) words. The IOID table is 64 (0100) words and must begin on an 0100 word boundary.

COBOL LOW MEMORY ALLOCATION

(48 x 24)

Screen #	User Table Address	Screen Location
1	0660-0707	060-0657
2	0710-0737	01060-01657
3	0740-0767	02060-02657
4	0770-01017	03060-03657
5	01020-01047	04060-04657
6	01050-01707	05060-05657
7	01710-01737	06060-06657
8	01740-01767	07060-07657
9	01770-02017	010060-010657
10	02020-02047	011060-011657
11	02050-02707	012060-012657
12	02710-02737	013060-013657
13	02740-02767	014060-014657
14	02770-03017	015060-015657
15	03020-03047	016060-016657
16	03050-03707	017060-017657

COBOL KEYBOARD USER TABLE

+0	+1	+2	+3	+4	+5 -
IENTRY	BRM KEYINT	EENTRY	BRM KEYERR	DEST Destination Address of Cursor	CH Character Position of Cursor

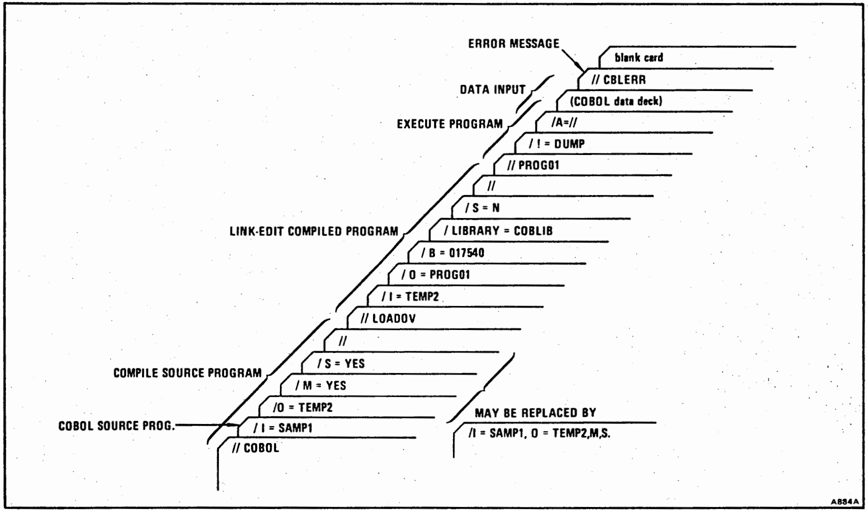
+6	+7	+010	+011	+012	+013 -
FLDST KEY-IN Field Start Address		CUR Character Position of Field Start	KITYP KEY-IN field type	CURC Keyboard status = 0 lock > 0 unlocked	CONT level 7 termination code

+014	+015	+016	+017	+020	+021 -
FLDCNT current # of of bytes counted in field	FLDLIM Max # of bytes in field	SPARE Spare Word	NOK # of buffered keystrokes	BUFF1 keystroke buffer word 1	BUFF2

+022	+023	+024	+025	+026	+027 -
BUFF3	BUFF4	VALSW Validation requested for this field	KYSUB keyboard subscript	KIPIC Address of KEY-IN picture (or zero)	@CONV Address of conversion table for this KBD

COBOL KEYBOARD USER TABLE

Location Relative to X1	Symbolic Name	Description
-2	IENTRY	Operand address of BRM in IOID table (RP save word). Normal keystrokes.
-1	IENTRY+1	BRM KEYINT - the normal keystroke processor.
0	EENTRY	Operand address of BRM in IOID table (RP save word). Hard lost keystrokes.
1	EENTRY+1	BRM KEYERR - the lost keystroke processor.
2	DEST	Word address of present cursor location.
3	CH	Byte offset of cursor in the word specified by DEST (0, 1, or 2).
4	FLDST	Word address of start of field being keyed into.
5	FLDST+1	Byte offset of start of field.
6	CUR	Contents of the cursor address word without a cursor in it (used to blink cursor off).
7	KITYP	Bits 0-7 indicate type of field being keyed.
8	CURC	0 = keyboard locked, 7 0 counts tenths of a second until this cursor is to be flashed on for .1 second.
9	CONT	If a "terminate" code has been generated by this keyboard, it is stored here until the declaratives section is executed (level 7).
10	FLDCNT	Binary column count of current cursor position in field (range: 1-FLDLIM).
11	FLDLIM	Total byte count of the field.
12	SPARE	Spare word.
13	NOK	Number of buffered keystrokes.
14	BUFF1	Keystroke Buffer 1
15	BUFF2	Keystroke Buffer 2
16	BUFF3	Keystroke Buffer 3
17	BUFF4	Keystroke Buffer 4
18	VALSW	Validation has been requested for this field.
19	KYSUB	COBOL subscript for this keyboard (binary screen number:1-32).
20	KIPIC	Address of the KEYIN picture (or zero if none).
21	@CONV	The address of the conversion table to be used with this keyboard.



JOB CONTROL STATEMENTS TO COMPILE, LINK-EDIT, AND EXECUTE

COBOL PROGRAM ORIGINATION POINTS
BOTTOM PARAMETERS FOR COBOL LOADOV JCL

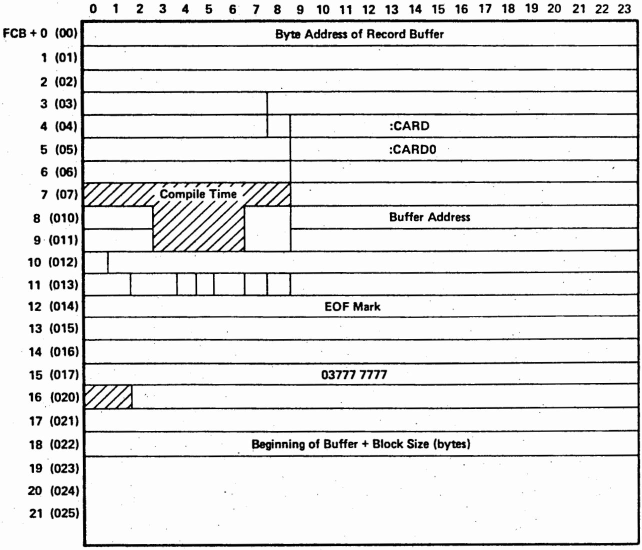
# of Screens	6x48	12x48	24x48	6x81	12x81	24x81
0*	00060	00060	00060	00060	00060	00060
1	02000	02000	02000	01700	01700	01700
2	02000	02000	02000	02000	02000	03540
3	02000	02000	02660	02000	02740	05540
4	02000	02000	03660	02000	03540	07540
5	02000	02360	04660	02440	04740	11540
6	03000	03000	05660	02740	05540	13540
7	03000	03360	06660	03700	06740	15540
8	03000	03660	07660	03700	07540	17540
9	03000	04360	10660	04440	10740	21540
10	03000	04660	11660	04740	11540	23540
11	04000	05360	12660	05240	12740	25540
12	04000	05660	13660	05540	13540	27540
13	04000	06360	14660	06440	14740	31540
14	04000	06660	15660	06740	15540	33540
15	04000	07360	16660	07240	16740	35540
16	05000	07660	17660	07540	17540	37540
17	05000	10360	20660	10440	20740	41540
18	05000	10660	21660	10740	21540	43540
19	05000	11360	22660	11240	22740	45540
20	05000	11660	23660	11540	23540	47540
21	06000	12360	24660	12440	24740	51540
22	06000	12660	25660	12740	25540	53540
23	06000	13360	26660	13240	26740	55540
24	06000	13660	27660	13540	27540	57540
25	06220	14360	30660	14440	30740	
26	07000	14660	31660	14740	31540	
27	07000	15360	32660	15240	32740	
28	07000	15660	33660	15540	33540	
29	07220	16360	34660	16440	34740	
30	07360	16660	35660	16740	35540	
31	10000	17360	36660	17240	36740	

* LOADOV will not correctly process any system with an actual load address less than 0412. COBOL programs can have a B parameter as low as 060 if there is sufficient file section space to put the procedure division code higher than 0412.

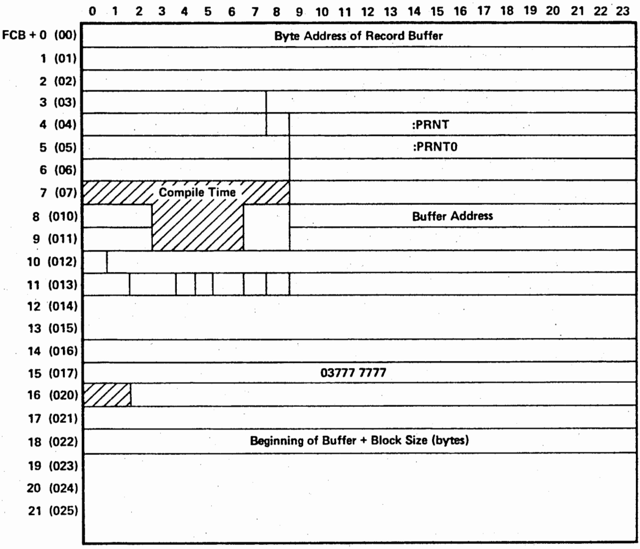
FILE CONTROL BLOCK (FCB) FORMAT

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th></tr>
<tr><td>FCB + 0</td><td colspan="24">Record Base Pointer (byte)</td></tr>
<tr><td>1</td><td colspan="24">Max Record Size (bytes)</td></tr>
<tr><td>2</td><td colspan="24">Buffer Size (bytes)</td></tr>
<tr><td>3</td><td colspan="8">De-Blocking Techn.</td><td colspan="16">Blocking Factor</td></tr>
<tr><td>4</td><td colspan="8">Device Index</td><td>0</td><td colspan="15">Device Block Handler Address</td></tr>
<tr><td>5</td><td colspan="9">BRM,</td><td colspan="15">Label Routine Address</td></tr>
<tr><td>6</td><td colspan="9">BAL/BRM</td><td colspan="15">Error Routine Address</td></tr>
<tr><td>7</td><td colspan="9">Device Type (Compile Time)</td><td colspan="15">Pointer in List of FCB's (0 = no more)</td></tr>
<tr><td>8</td><td colspan="3">Buffer</td><td colspan="4" rowspan="2"></td><td colspan="2" rowspan="2">0</td><td colspan="15">IO Area 1</td></tr>
<tr><td>9</td><td colspan="3">Flags</td><td colspan="15">IO Area 2 (= 0 if not used)</td></tr>
<tr><td>10</td><td>I</td><td colspan="23">File Position (or Key Address if Random)</td></tr>
<tr><td>11</td><td colspan="2">Open</td><td colspan="2">EOF</td><td>*</td><td>**</td><td>***</td><td></td><td>FS</td><td colspan="15">Address of File Status Word</td></tr>
<tr><td>12</td><td colspan="24" rowspan="2">File Name</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td><td colspan="24" rowspan="2">Variable – See Table 10-3</td></tr>
<tr><td>15</td></tr>
<tr><td>16</td><td colspan="2"></td><td colspan="22">Rerun Record Count (= 0 For no Rerun)</td></tr>
<tr><td>17</td><td colspan="24">Rerun Record Counter</td></tr>
<tr><td>18</td><td colspan="24" rowspan="2">Variable – See Table 10-2</td></tr>
<tr><td>43</td></tr>
</table>

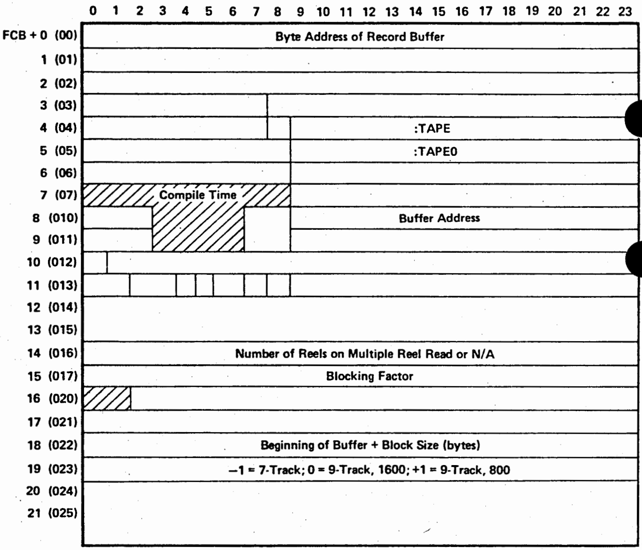
CR FCB


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
FCB + 0 (00)
Byte Address of Record Buffer
1 (01)
2 (02)
3 (03)
4 (04)
:CARD
5 (05)
:CARD0
6 (06)
7 (07)
Compile Time
8 (010)
Buffer Address
9 (011)
10 (012)
11 (013)
12 (014)
EOF Mark
13 (015)
14 (016)
15 (017)
03777 7777
16 (020)
17 (021)
18 (022)
Beginning of Buffer + Block Size (bytes)
19 (023)
20 (024)
21 (025)

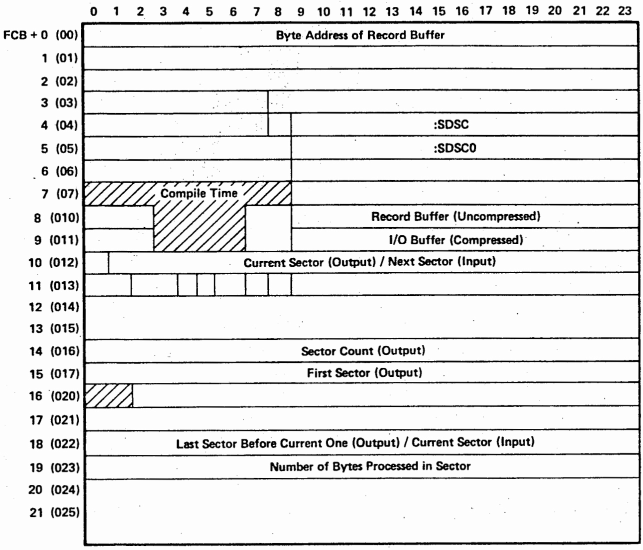
PRFCB


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
FCB + 0 (00)
1 (01)
2 (02)
3 (03)
4 (04)
5 (05)
6 (06)
7 (07)
8 (010)
9 (011)
10 (012)
11 (013)
12 (014)
13 (015)
14 (016)
15 (017)
16 (020)
17 (021)
18 (022)
19 (023)
20 (024)
21 (025)
Byte Address of Record Buffer
:PRNT
:PRNT0
Compile Time
Buffer Address
03777 7777
Beginning of Buffer + Block Size (bytes)

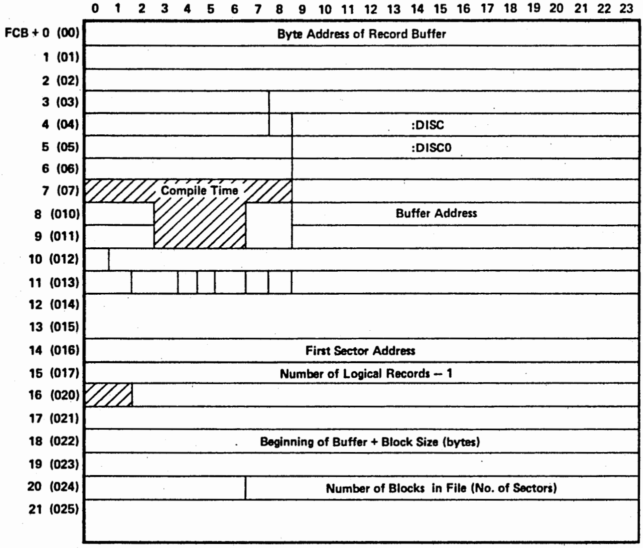
MT FCB


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
FCB + 0 (00)
Byte Address of Record Buffer
1 (01)
2 (02)
3 (03)
4 (04)
:TAPE
5 (05)
:TAPE0
6 (06)
7 (07)
Compile Time
8 (010)
Buffer Address
9 (011)
10 (012)
11 (013)
12 (014)
13 (015)
14 (016)
Number of Reels on Multiple Reel Read or N/A
15 (017)
Blocking Factor
16 (020)
17 (021)
18 (022)
Beginning of Buffer + Block Size (bytes)
19 (023)
−1 = 7-Track; 0 = 9-Track, 1600; +1 = 9-Track, 800
20 (024)
21 (025)

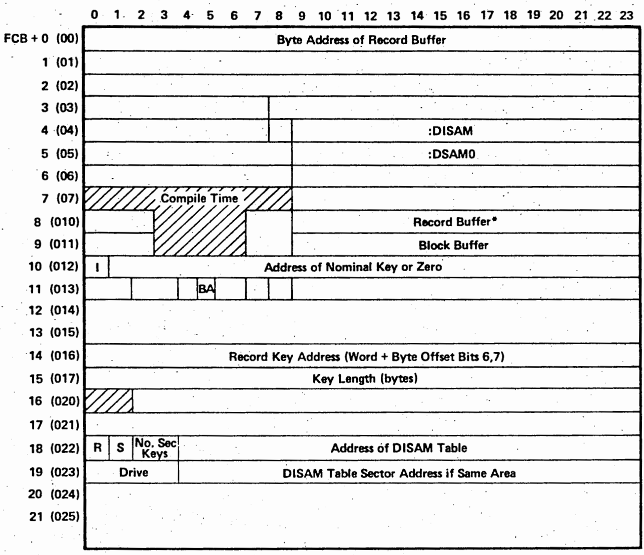
SD FCB


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
FCB + 0 (00)
Byte Address of Record Buffer
1 (01)
2 (02)
3 (03)
4 (04)
:SDSC
5 (05)
:SDSC0
6 (06)
7 (07)
Compile Time
8 (010)
Record Buffer (Uncompressed)
9 (011)
I/O Buffer (Compressed)
10 (012)
Current Sector (Output) / Next Sector (Input)
11 (013)
12 (014)
13 (015)
14 (016)
Sector Count (Output)
15 (017)
First Sector (Output)
16 (020)
17 (021)
18 (022)
Last Sector Before Current One (Output) / Current Sector (Input)
19 (023)
Number of Bytes Processed in Sector
20 (024)
21 (025)

DC FCB


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
FCB + 0 (00)
Byte Address of Record Buffer
1 (01)
2 (02)
3 (03)
4 (04)
:DISC
5 (05)
:DISC0
6 (06)
7 (07)
Compile Time
8 (010)
Buffer Address
9 (011)
10 (012)
11 (013)
12 (014)
13 (015)
14 (016)
First Sector Address
15 (017)
Number of Logical Records – 1
16 (020)
17 (021)
18 (022)
Beginning of Buffer + Block Size (bytes)
19 (023)
20 (024)
Number of Blocks in File (No. of Sectors)
21 (025)

DISAM FCB



I = 0, No Nominal Key	BA = Block Altered	R = Random
1, Nominal Key	S = Same Area for Sec Key	

FCB Word Detail -1

Word	Bit Position	Value	File Types	Meaning
0			NA	Current record address (absolute address in bytes)
1			NA	Maximum record size (bytes)
2			NA	Buffer size (bytes)
3	0-7		NA	Deblocking technique
		0	NA	SD file
		1	NA	Fixed, unblocked
		2	NA	Fixed, blocked
		3	NA	Variable, unblocked
		4	NA	Variable, blocked
		5	NA	Random DC, unblocked
		6	NA	Random DC, blocked
		7	NA	ISAM sequential, fixed
		8	NA	ISAM random, fixed
		9	NA	DISAM sequential, fixed
		10	NA	DISAM random, fixed
		11	NA	DATA IV/70
3	8-23		NA	Blocking factor
4†	0-7		NA	Device index from the @ drive parameter JCL statement
	8	0	NA	Constant
	9-23		NA	I/O record block routine for specific type of file
				:DISAM DISAM file :TAPE tape file
				:ISAM ISAM file :CARD card file
				:SDSC SD file :PRNT print file
				:DISC DC file
5†	0-8	BRM	NA	
	9-23		NA	Open and close routine for specific type of file
				:DSAMO DISAM file :TAPEO tape file
				:ISAMO ISAM file :CARDO card file
				:SDSCO SD file :PRNTO print file
				:DISCO DC file
6	0-8	BAL	NA	
	9-23	:FATAL	NA	Error routine address
6	0-8	BRM	NA	Standard error procedure address, if USE AFTER STANDARD ERROR is used
		:ERRx	NA	
7	0-8		NA	Device type (compile time only)
		1	NA	Card reader
		2	NA	Printer
		3	NA	Magnetic Tape
		4	NA	DC
		5	NA	D4
		6	NA	SD

†In words 4 and 5, :DISAM, :ISAM, etc. are virtuals at compile time (see the Procedure Division Map).

FCB Word Detail -2

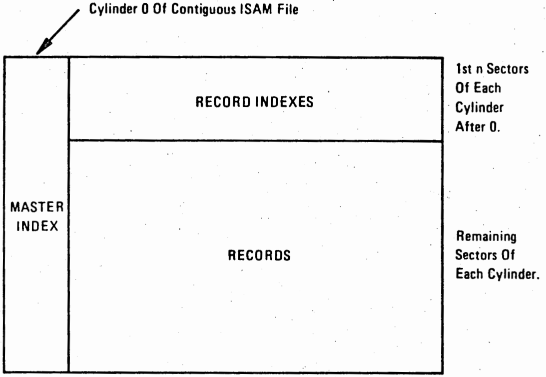
Word	Bit Position	Value	File Types	Meaning
		7	NA	ISAM
		8	NA	DISAM
	9-23		NA	Pointer to chain list of FCB's (0 = no more)
8,	0-2 }		NA	Buffer flags
9	0-2 }	0	NA	No buffer
		1	NA	CPU using
		2	NA	Initiate write
		3	NA	Write in progress
		4	NA	Write complete
		5	NA	Initiate read
		6	NA	Read in progress
		7	NA	Read complete
8,9	3-6		NA	Compile time use
8,9	7-8	0	NA	
8	9-23		NA	I/O area 1 For DISAM: Master record index, DISAM table, secondary key index, reford buffers. For ISAM: Master record index, record index buffer.
9	9-23		NA	I/O area 2 blocking buffer (=0 if not used)
10	0	1	NA	nominal key present
		0	NA	no nominal key
	1-23		NA	File position (address of Nominal or Actual Key if random file).
11	0-1		NA	File status
		00	NA	Closed
		01	NA	Output — open
		10	NA	Input — open
		11	NA	I/O — open
	2-3		NA	End of File value
		00	NA	Beginning label
		01	NA	Close before EOF
		10	NA	EOF encountered
		11	NA	Close after EOF
	4	1	NA	Multi-reel
	5	1	NA	Block altered
	6	1	NA	Same area
	7	Blank	NA	Not used
	8	1	NA	FILE STATUS option used
	9-23		NA	Address of File status word
12,13			NA	Until first file is opened, word 12 contains the byte specified in the COBOL ASSIGN TO XX-a. Then 12 and 13 contains the six byte "file-name" associated with this byte.

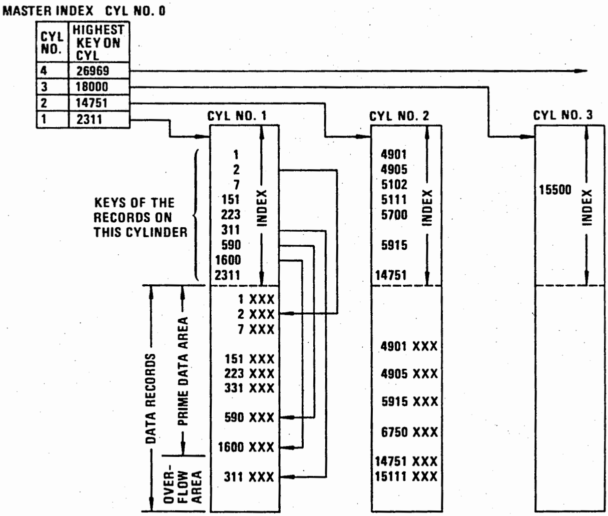
FCB Word Detail -3

Word	Bit Position	Value	File Types	Meaning
14		0	SD,MT,CR,PR	N.A.
			DC	First sector address
			DI, IS	Word + byte offset address of record key
15		7777	SD input	Compile time use
			SD output	First sector and sector count
			DC	Number of logical records -1
		0	MT input	Compile time use
			MT output	Blocking factor
		03777777	CR,PR	Compile time use
			DI	Key length (bytes)
			IS	Third word in Directory Entry Format (value +0 in ISAM)‡
16			All types	Rerun record count (=0 for no rerun) (value +1 in ISAM)
17			All types	Rerun record counter (value +2 in ISAM)
18			SD	Next/last pointer
			DC,MT,CR,PR	Record base and block size
			DI	Number of secondary keys
			IS	(Value +3 in ISAM)‡
19			SD	Bytes processed on sector counter
			DC, CR, PR	Not used
19	0	0	MT	1600 bpi
		1	MT	800 bpi
19	0-7		DI	Drive number
	8-23		DI	DISAM table sector address when same area clause used
19			IS	(Value +4 in ISAM)‡
20	7-23		DC	Number of blocks in the file
20			IS	(Value +5 in ISAM)‡
21 (ISAM only)			IS	Key length (bytes)

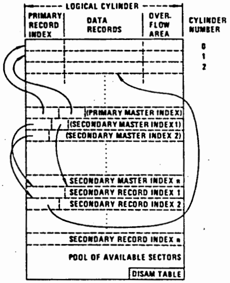
‡ Value +0 through +5 refers to words 2 through 7 in the Directory Entry of the ISAM file.

ISAM File Layout

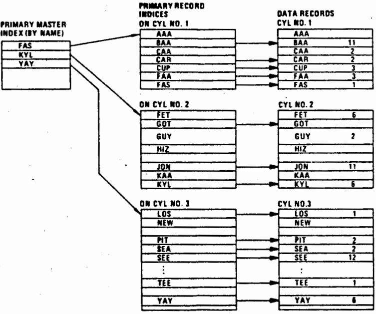


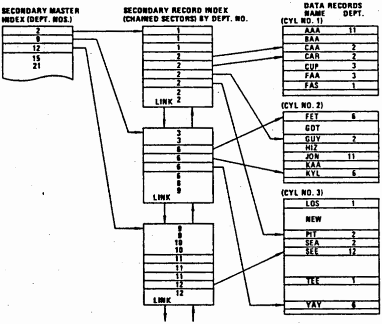


DISAM File Structure



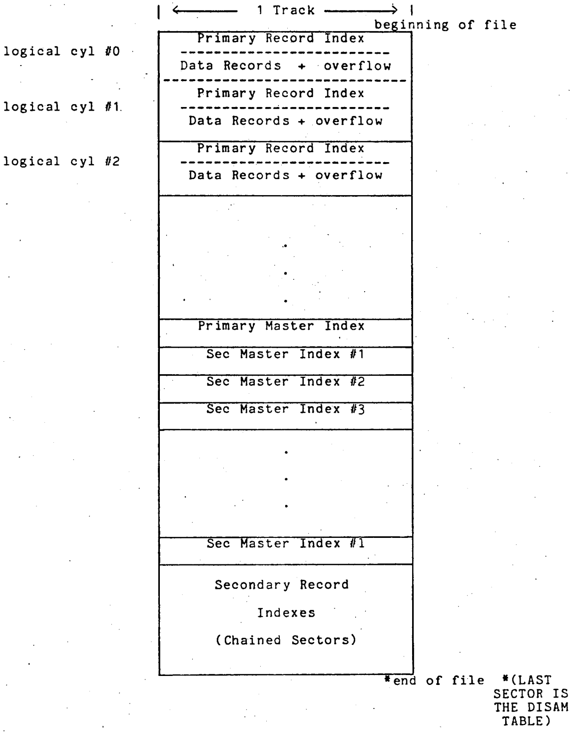
A890A





A890C

DISAM FILE LAYOUT



DISAM TABLE

OFFSET	NAME	DESCRIPTION
01	$IDXCY	# Index Sectors/Logical Cylinder
05	$CYLSI	# Sectors/Logical Cylinder
07	$PMISI	#Primary Master Index Sectors
015	$RNBIT	#bits in maximum cylinder rec #*

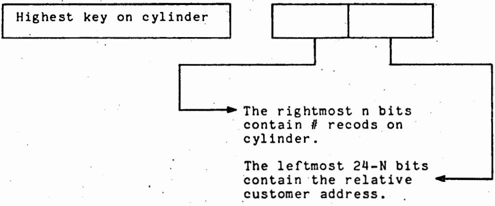
* The value of $RNBIT (n) is determined by the maximum number of bits needed to store the highest record number in a cylinder

DEVICE	MAXIMUM # OF BITS		# SECTORS/ PHYSICAL CYL
	CYL#	REC#	
8230	12	10	16
8240	9	14	160
8260	10	14	110

VALUE OF N

1. Primary Master Index contains one entry for each cylinder where the entry contains the highest value key on the cylinder; a key of all 1's represents te highest possible key.

Each entry has the following format:



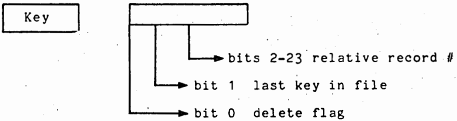
The rightmost n bits contain # recods on cylinder.

The leftmost 24-N bits contain the relative customer address.

(N is dependant on the type of disc)

2. Primary Record (or Cylinder) Index occupies the first tracks of each logical cylinder; it contains one entry for each data record in the cylinder The highest key will appear as is all 1's followed by a one-word pointer with bit 4 set.

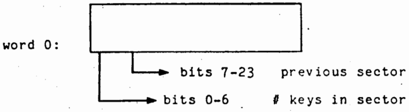
Each entry has the following format:

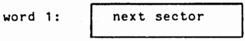


3. Secondary Master Index There is one secondary master index for each secondary key field. Each such index contains one entry for each sector in that secondary record index. Each entry contains the highest valued key in the sector (highest key is all 1's) and has the following format.

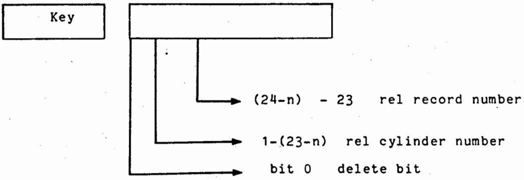
4. Secondary Record Index The sectors in each of the secondary record indexes (one index for each secondary key) are chained. $KEYS+6 in the DISAM Table has the absolute sector address of the first sector in the secondary record index for the first secondary key.* That sector then has a pointer to the next sector in the chain. The pointers are followed by entries, one for each data record.

The first two words in the sector have the following format:





The remainder of the sector consists of entries of the following format:



The highest key has all 1's followed by a pointer of 37777777.

* The absolute sector addresses of the first sector for the other secondary indexes are in the DISAM Table, too.
add the fillowing after DISAM FILE STRUCTURE page

DISAM Indexes

All indexes except Secondary Record Indexes are multiples of tracks because their sectors are read/written every other sector.

physical sector #		logical sector #
1	is	1
2		5
3		2
4		6
5		3
6		7
7		4
8		8

From sector to sector, keys are in ascending order; that is, all keys in logical sector #6 are greater tthan those in logical sector #3.

Within a sector, all keys are in descending sequence.

All pointers in indexes are relative to either start of file, start of cylinder, or start of sector.

DISAM TABLE

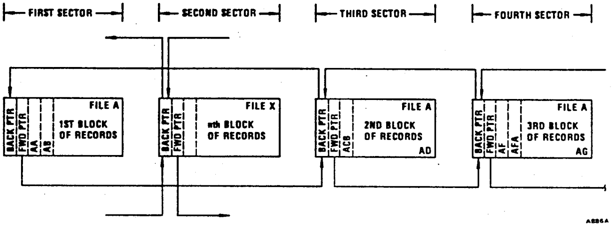
These values are set by ALOCDI and are permanent.

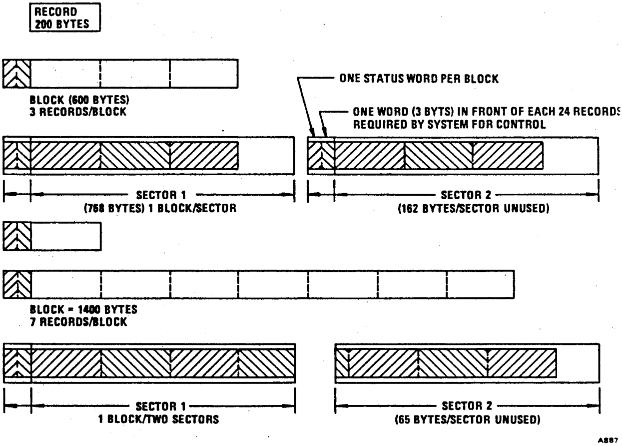
$UTFAC	Utilization factor
$IDXCY	# index sectors/cylinder
$KSIZ	Key length in bytes
$BLKFA	Blocking factor
$RSIZ	Record size
$CYLSI	Logical cylinder size (in sectors)
$OVCYL	Cylinder overflow allowed? 0=yes,greater than 0 = no
$PMISI	Number of primary master index sectors
$MISL	Master index sector location (relative)
$PSEC	# prime data sectors/cylinder
$KSEC	Number of keys per sector
$SECRB	Number of sectors per block
$MREC	Maximum Number of records/cylinder
$RNBIT	# bits in maximum cylinder record number
$RNMAS	Mask for cylinder record number
$KEYSI	Key length in words (including pointer)
$NEXTA	Location of first available free sector
$POOLA	Rel sector addr of free pool
$KEYS	Secondary key info. (rel. word & byte offset of key)
$KEYS1	Secondary key info. (key length in bytes, w/o pointer)
$KEYSM	Master index sector location
$KMBUF	Master index buffer location
$KRBUF(1 word)	Record index buffer location
(1 word)	Master index sector buffer
(1 word)	Abs. secondary record index sector addr
(63 words)	9 more keys

These locations are initialized in $DSAMO and/or set in $DISAM

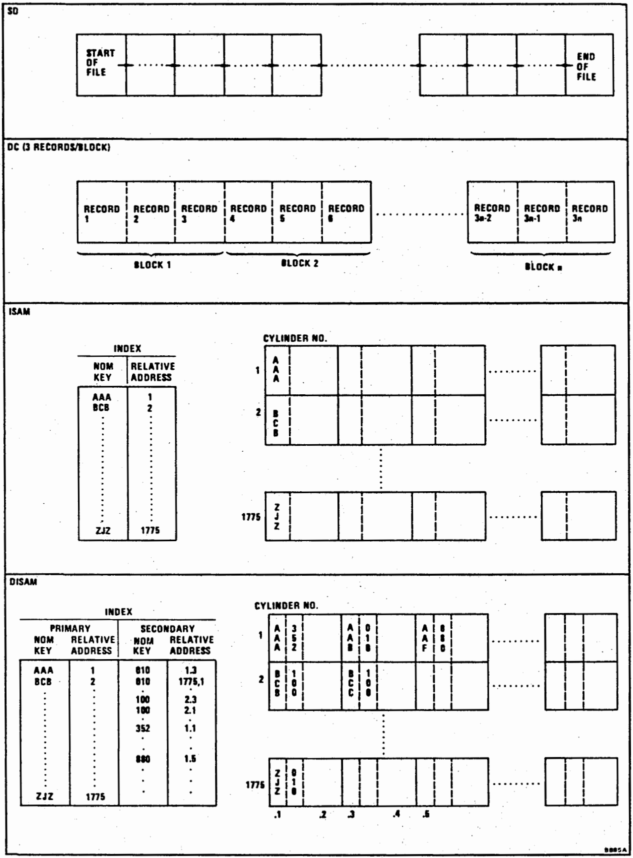
DSTART	
$BLKBU(1 word)	Block buffer address
(1 word)	2nd word of 11th secondary key (always 0)
$CBUFS	Cylinder index buffer ccounter
$CYLAD	Location of record index pseudo-cylinder
$DRIVE	Logical drive number of file
$DSEC	Current # data sectors in cylinder
$FILAD	File address
$KEYSZ	Key length in bytes (including pointer)
$LSEC	Last sector address
$MAS	Master index starting sector address (ABS)
$MASBU	Master index buffer address
$MASPT	Master index buffer word pointer
$MASSE	Master index sector address
$MBUFS	Master index buffer counter
$MISN	Master index sector number
$RBUFS	Block buffer counter
$RBYTE	Byte offset of record buffer
$RECPT	Ptr to start of key in record index buffer
$RECSE	Absolute record index sector address
$RECBU	Record index buffer address
$RQT00	Data block request table
$RQT01	
$RQT02	
$RQT03	
$RQT04	
$RQT10	Record index request table
$RQT11	
$RQT12	
$RQT13	
$RQT14	
$RQT20	Master index request table
$RQT21	
$RQT22	
$RQT23	
$RQT24	
$RISN	Record index sector number
$RWORD	Word location of record buffer
$SAVEA	Same area for more than one file
$SAVCY	Cylinder address save area
$SAVRI	Record index sector number save area
$SBUFF	Same buffer area for secondary keys
$STKEY	Key flags start before read backwards

SD FILE STRUCTURE





DISC FILE ORGANIZATION



LIBGEN QUESTIONS

```
ALTER THE NEW LIBRARY NAME IF YOU WISH. TEMPO300
IS THE DEFAULT.
USE THE TAB KEY TO END A FIELD.
ALTER THE LISTING TITLE IF YOU WISH.
TITLE.  COBOL LIBRARY CUSTOMIZER MODULE.
```

Answer these messages with a Y or N.

```
WILL YOU USE MAGNETIC TAPE?
WILL YOU USE 1600 BPI?
WILL YOU USE 800 BPI?
WILL YOU USE A CARD READER?
IS IT A BUFFERED CARD READER?
WILL YOU USE PRINTER FILES?
DO YOU WISH JCL AND ACCEPT DATA ITEMS
AND TAPE MOUNT MESSAGES TO BE LISTED?
WILL YOU USE A 8121 PRINTER?
WILL YOU USE A 8131 PRINTER?
WILL YOU USE A 8145/8146 LINE PRINTER?
CAUTION: YOU MUST USE AT LEAST ONE TYPE OF DISK.
WILL YOU USE 8230 DISC?
WILL YOU USE 8240 DISC?
WILL YOU USE 8260 DISC?
```

(If no disc is specified, the following message will appear and LIBGEN will return to the beginning and ask the above questions over again when the next key is pressed.)

```
****ERROR NO DISK IS USED.****
PRESS ANY KEY TO CONTINUE.
WILL YOU USE SD TYPE (CHAINED) FILES?
WILL YOU SPOOL PRINTER OUTPUT?
WILL YOU USE DISAM FILES?
WILL YOU USE SECONDARY KEYS IN A DISAM FILE?
WILL A DISAM FILE BE OPEN FOR OUTPUT OR I-O?
WILL A DISAM FILE BE OPEN FOR SEQUENTIAL OUTPUT ONLY
(INITIAL LOAD)?
WILL THE VERB READ BE USED FOR A SEQUENTIAL DISAM FILE?
WILL THE VERB READ BE USED FOR A RANDOM DISAM FILE?
WILL THE VERB WRITE BE USED FOR A DISAM FILE?
WILL THE VERB REWRITE BE USED FOR A DISAM FILE?
WILL THE VERB DELETE BE USED FOR A DISAM FILE?
WILL THE VERB START BE USED FOR A DISAM FILE?
WILL YOU HAVE DISAM FILES THAT ALLOW CYLINDER OVERFLOW?
WILL READ BACKWARD BE USED FOR A SEQUENTIAL DISAM FILE?
WILL READ BACKWARD BE USED WITH A SECONDARY KEY FOR A SEQUENTIAL
DISAM FILE?
WILL YOU DO COMMUNICATION LOGGING TO DISC?
WILL YOU TAKE CHECKPOINTS OR ERROR DUMPS?
ARE YOU A NTP 150, 230, or 250 USER?
IS YOUR BAUD RATE (LINE SPEED) 4800 OR BELOW?
IS YOUR BAUD RATE BETWEEN 2401 AND 4800?
     (BELOW 1200 IS IMPLIED IF YOU ANSWER N).
IS YOUR BAUD RATE BETWEEN 4801 AND 7200?
IS YOUR BAUD RATE BETWEEN 7201 AND 9600?
     (ABOVE 9600 IS IMPLIED IF YOU ANSWER N).
IS YOUR NETWORK MULTI-POINT?
     (POINT-TO-POINT IS IMPLIED IF YOU ANSWER N).
PRESS ANY KEY TO CONTINUE OR BOOT TO BYPASS ASSEMBLY.
```

COMMONLY USED LOW MEMORY LOCATIONS IN MONITR

The first 0100 locations of the bootstrap section of MONITR are considered to be frozen and may be accessed by the programmer as absolute locations. These locations are carried over from earlier releases. In addition, SBUF, the system card image buffer, is considered to be frozen at location 0140 because of the strategic location for both 48 and 81 character-per-line video display systems. Locations 0173-0177 were added to the absolute locations at release B06.

Note that user programs that employ SYSIN and/or SYSOUT in environments where more than one SYSIN or SYSOUT device is in use must have reference to certain of these locations. Specifically, CINPUT, LUCONV, and LPOUT furnish information required in multiple input and output situations. If SYSIN or SYSOUT are being used without MONITR, it will be necessary to read sector 0 of the disc if the status of these locations is required.

Commonly Used Low Memory Locations:

Symbolic Location	Absolute Location	Contents/Significance
BT10-1	0001	Bootstrap entry point and restart location. - Resets SYSTCK (system input stack) to empty. - Resets CINPUT (SYSIN input flag) to take input from keyboard. - Zeros out rest of memory. - Reads rest of MONITR from disc. - Enables memory parity checking. - Transfers control to MONITR.
BT10	0002	Same as 0001 except that SYSTCK and CINPUT are not changed.
MEMORY	0033	Highest memory address +1.
CINPUT	0034	SYSIN input flag : 0 = keyboard, nonzero = cards.
LUCONV	0035	Lower case conversion for SYSIN. = 0 Convert lower case to upper case on input. NOT = 0 No conversion of lower case to upper case.
LPOUT	0036	SYSOUT Printer index: 0 = 8145 1 = 8143 2 = 8121 3 = 8131 4 = No print 5 = 8146, 8148
SYSTCK	0041	SYSIN input stack pointer for disc procedure files.
STWRD	0071	Status word location for $DISC.

Commonly Used Low Memory Locations: (Continued).

SBUF	0140	System input buffer used by SYSIN. Eighty characters long with a line feed in the 81st position. When the system is bootstrapped this buffer will contain "// SYSTEM IV/70 DISC OPERATING SYSTEM 88-0017. XX",
SBUF+1	0141	Used with SBUF+2 to obtain name on current control statement.
SBUF+2	0142	Used with SBUF+1 to obtain name on current control statement.
SBUF+033 to 037	0173-0177	System communication area. This area is used by COBOL to communicate between modules. These locations can also be used by user programs that reside with MONITR in memory, but a call to EXIT or rebootstrapping the program will destroy the information.
$HEAD	0200	Routine used for disc I/O to read or write headers only. Calling sequence same as $DISC
$DISC	0204	Routine used for disc I/O except for reading headers. Calling sequence"

```
BAL   $DISC          Linkage
  PZE REQTAB         Address of disc I/O
                        request table
  HLT  $             Disc error
```

Clearing a Printer Halt Under DOS

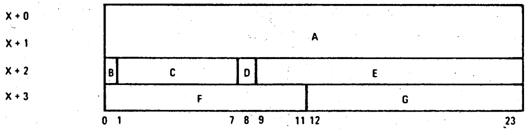
1. AUTO to Manual, RESET then STEP
2. LOAD 036 into RP
3. LOAD 04 into MEM
4. LOAD 72000001 into TIR
5. MANUAL to AUTO

Allocation Table

The allocation table is 200 words long, one word representing each usable cylinder. Within each word, bits 0-15 represent the 16 relative sectors on the corresponding cylinder. If the bit is on, the represented sector is available. If off: the sector is non-existent, in cylinder 0, in use, or in a file whose name is blank. Once allocated to a file, sectors are not available for reuse after the file is deleted until JOB is run. BOJ will make available any sectors missed by JOB.

Disc Directory

The disc directory has room to record 576 files/disc. Each entry is four words long and formatted as follows.

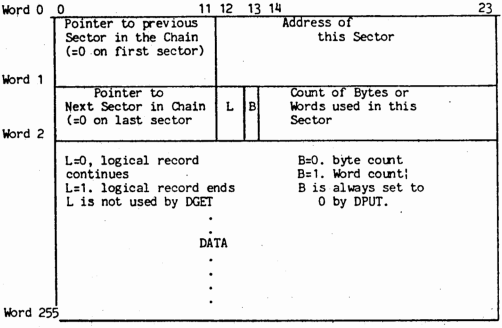


Word	Bits	Symbol	Meaning
X + 0, X + 1	0-23 0-23	A	Six character ASCII name with all parity bits 0.
X + 2	0	B	1 = protected 0 = automatic deletion rule applies, i.e., a file is deleted whenever a new file of the same name is created.
	1-7	C	Flag byte. Seven bits of information available to application programs.
	8	D	Format bit. 0 = contiguous 1 = chained
	9-23	E	Contiguous files: E = 0 if data, otherwise E is load address in octal of memory load. Chained files: E = ending sector in octal; 06177 (3199) maximum.
X + 3	0-11	F	Sector count -1 in octal, F≤3199.
	12-23	G	Starting sector in octal; 06177 (3199) maximum.

A131

The area from the last entry to the end of the directory contains binary zeros. When an entry is deleted, its name is changed to all blanks. That entry area cannot be used until JOB is run which compacts entries, removing all those with blank names and names that start with TEMP.

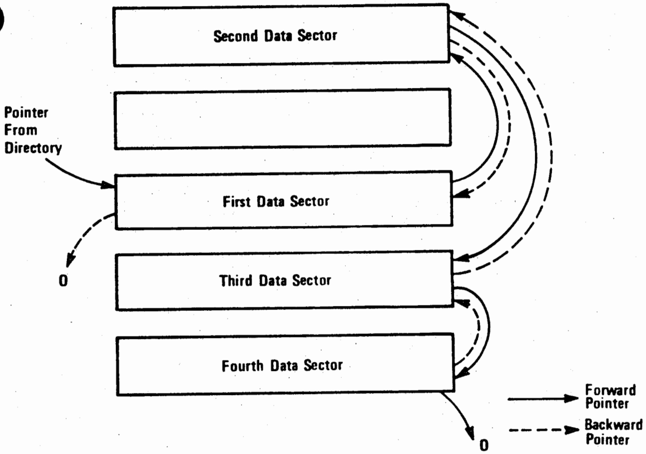
Sector Format of a Chained File

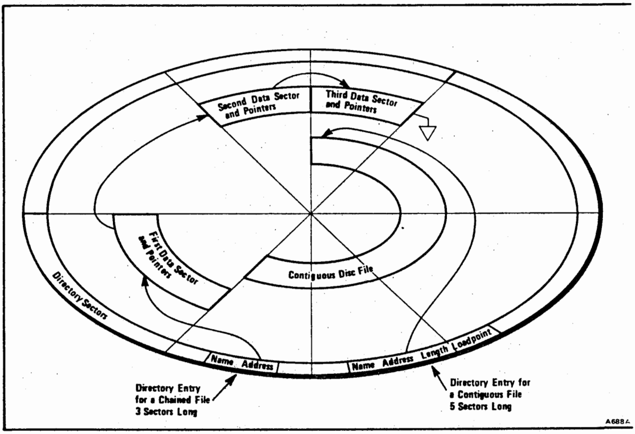


L=0, logical record continues
L=1. logical record ends
L is not used by DGET

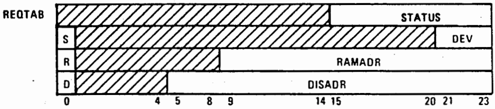
B=0. byte count
B=1. Word count;
B is always set to 0 by DPUT.

Structure of a Chained File





$DISC Request Table



Word	Bits	Symbol	Meaning
REQTAB	23-0	STATUS†	Bit 23 - Not Ready Bit 22 - Busy Bit 21 - Cyclic Redundancy Error Bit 20 - Too Late Bit 19 - Header Error Bit 18 - Head Out of Range Bit 17 - Seek Incomplete Bit 16 - Illegal Request Bit 15 - Incorrect Length Bits 0-14 - Not Used
REQTAB+1	23-21	DEV‡	Index to drive 0 ≤ device ≤ 3
	20-1	—	Not used
	0	S‡	1 Do seek only 0 Do data transfer (with seek if necessary)
REQTAB+2	23-9	RAMADR‡	Location in memory of data buffer
	8-1	—	Not used
	0	R‡	1 Normal return only when status = 0 0 Normal return after operation initiated or stacked
REQTAB+3	23-5	DISADR‡	Disc address request word SECTOR COUNT - 1 | CYLINDER | SECTOR 5 11 12 19 20 23
	4-1	—	Not used
	0	D‡	1 write data on disc 0 read data from disc

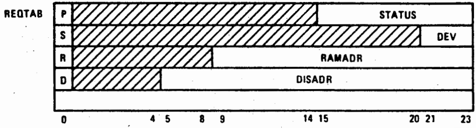
†Furnished by $DISC

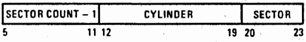
‡Furnished by user program

The reject return is taken if any of the input values are out of range. Note that cylinders 200 through 202 are illegal.

A098F

$IDISC Request Table



Word	Bits	Symbol	Meaning
REQTAB	23-0	STATUS†	Bit 23 - Not Ready Bit 22 - Busy Bit 21 - Cyclic Redundancy Error Bit 20 - Too Late Bit 19 - Header Error Bit 18 - Head Out of Range Bit 17 - Seek Incomplete Bit 16 - Illegal Request Bit 15 - Incorrect Length Bits 1-14 - Not Used
		P†	Bit 0 - Status posted, operation complete
REQTAB+1	23-21	DEV‡	Index to drive (0 ≤ device ≤ 3)
	20-1	—	Not used
	0	S‡	1 Do seek only 0 Do data transfer (with seek if necessary)
REQTAB+2	23-9	RAMADR‡	Location in memory of data buffer
	8-1	—	Not used
	0	R‡	1 Normal return only when status bit 0 (P) = 0 0 Normal return after operation initiated or stacked
REQTAB+3	23-5	DISADR‡	Disc address request word 
	4-1	—	Not used
	0	D‡	1 write data on disc 0 read data from disc
REQTAB+4	23-0	—	BSS 1 — used by queuing software

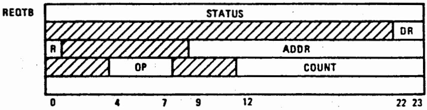
†Furnished by $IDISC

‡Furnished by user program

The reject return is taken (status bits 0, 16 = 1) if any of the input values are out of range, Note that cylinders 200 through 202 are illegal.

A3638

$ITAPE Request Table

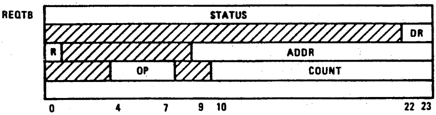


Word	Bits	Symbol	Meaning
REQTB	23-0	STATUS†	Bit 23, Drive not ready Bit 22, Busy (error condition) Bit 21, Parity error Bit 20, Write Protect Bit 19, Beginning of Tape (BOT) Bit 18, End of Tape (EOT) Bit 17, Too Late Bit 16, File mark Bits 15-13, Not used Bits 12 and 11, Byte boundary on a read - 00, Last data word full, operation complete - 01, Last word has 1 byte left justified - 10, Last word has 2 bytes left justified - 11, Last word full, operation complete Bit 10, Short count Bit 9, Device address out of range (see REQTB+2)‡ Bit 8, Incorrect operation type (see REQTB+3)‡ Bit 7, Number of words transmitted is wrong‡ Bit 6, Lost interrupt Bits 5-1, Not used Bit 0 - 1, Status posted, operation complete‡ - 0, Operation incomplete‡
REQTB+1	23-22	DR‡	Drive index: 00, 01, 10, or 11
	21-0		Not used
REQTB+2	23-9	ADDR‡	Memory address of tape data buffer
	8-1		Not used
	0	R‡	Bit 0 - 1, Normal return only when operation complete - 0, Return when operation queued or initiated
REQTB+3	23-12	COUNT‡	Read or Write count in bytes
	11-8		Not used
	7-4	OP‡	Operation Type 000 - Read; interrupts 001 - Write; interrupts 002 - Write file mark 003 - Erase 004 - Skip record 005 - Backspace record 006 - Skip file 007 - Backspace file 010 - Rewind 011 - Reset 012 - Read backward; interrupts 013 - Read backward; lockup 014 - Read; lockup 015 - Write; lockup
	3-0		Not used
REQTB+4	23-0	—	BSS 1 — Used by time critical processing in $ITAPE

†Furnished by $ITAPE ‡Furnished by user program ‡Generated by $ITAPE; other bits come from controller and are posted by $ITAPE.

A207D

$JTAPE Request Table

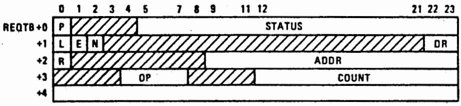


Word	Bits	Symbol	Meaning
REQTB	23-0	STATUS†	Bit 23, Drive not ready Bit 22, Busy Bit 21, Hard Error Bit 20, Write Protect Bit 19, Beginning of Tape (BOT) Bit 18, End of Tape (EOT) Bit 17, Too Late Bit 16, File mark Bit 15, Transfer required Bit 14, Not used Bit 13, Rewinding Bits 12 and 11, Byte boundary on a read = 00, Last data word full, operation complete = 01, Last word has 1 byte left justified = 10, Last word has 2 bytes left justified = 11, Last word full, operation complete Bit 10, Short count Bit 9, Reject Bit 8, Corrected parity Bit 7, 1600 ID Bit 6, Operation complete Bit 5, Device address out of range (see REQTB+2)‡ Bit 4, Incorrect operation type (see REQTB+3)‡ Bit 3, Number of words transmitted is wrong‡ Bit 2, Lost interrupt Bit 1, Not used Bit 0 = 1, Status posted, operation complete‡ = 0, Operation incomplete‡
REQTB+1	23-22	DR‡	Drive index: 00, 01, 10, or 11
	21-0		Not used
REQTB+2	23-9	ADDR‡	Memory address of tape data buffer
	8-1		Not used
	0	R‡	Bit 0 = 1, Normal return only when operation complete = 0, Return when operation queued or initiated
REQTB+3	23-10	COUNT‡	Read or Write count in bytes
	9-8		Not used
	7-4	OP‡	Operation Type 000 = Read 001 = Write 002 = File mark 003 = Erase 004 = Skip record 005 = Backspace record 006 = Skip file 007 = Backspace file 010 = Rewind 011 = Reset
	3-0	—	Not used
REQTB+4	23-0	—	BSS 1 — Used by time critical processing in $JTAPE

†Furnished by $JTAPE ‡Furnished by user program ‡Generated by $JTAPE; other bits come from controller and are posted by $JTAPE.

A364A

$TPE Request Table



Word	Bits	Symbol	Meaning
REQTB	23-0	STATUS†	Bit 23, Drive not ready Bit 19, Beginning of Tape (BOT) Bit 22, Busy (error condition) Bit 18, End of Tape (EOT) Bit 21, Parity error Bit 17, Data Lost (too late) Bit 20, Write Protect Bit 16, File mark Bits 15-13, Not used Bits 12 and 11, Byte boundary on a read - 00, Last data word full, operation complete - 01, Last word has 1 byte left justified - 10, Last word has 2 bytes left justified - 11, Last word has 3 bytes left justified Bit 10, Short count Bit 9, Even Parity Bit 8, Incorrect operation type (see REQTB+3)‡ Bit 7, Number of words transmitted is wrong‡ Bit 6, Lost interrupt Bit 5, Head Out of Range (see REQTB+2)‡ Bits 4-1, Not used
		P†	Bit 0 - 1, Status posted, operation complete‡ - 0, Operation incomplete‡
REQTB+1	23-22 21-3 2 1 0	DR‡ — N E L	Drive index: 00, 01, 10, or 11. Not used Density: 1 - Low, 0 - High Parity Select: 1 - Even, 0 - Odd Lock-up: 1 - Lockup, 0 - Interrupt per Word
REQTB+2	23-9 8-1 0	ADDR‡ — R‡	Memory address of tape data buffer Not used Bit 0 - 1, Normal return only when operation complete - 0, Return when operation queued or initiated
REQTB+3	23-12 11-8 7-4	COUNT‡ — OP‡	Read or Write count in bytes Not used Operation Type 000 - Read 005 - Backspace record 001 - Write 006 - Skip file 002 - Write file mark 007 - Backspace file 003 - Erase 010 - Rewind 004 - Skip record 011 - Reset
	3-0	—	Not used
REQTB+4	23-0	—	BSS 1 — Used by time critical processing in $TPE

† Furnished by $TPE ‡ Furnished by user program ‡ Generated by $TPE; other bits come from controller and are posted by $TPE.

A441C

IDOS NOTES:

- To bypass Autoboot (AD32 and AD33-A)

  1. Set AUTO/MANUAL switch to MANUAL
  2. Depress BOOT switch
  3. Set console switch 0 up
  4. Set AUTO/MANUAL switch to AUTO
  5. When cursor appears, return switch 0 to its down position

- To bypass MOD II set (AD32 and AD33-A) at boot time:

  1. Set AUTO/MANUAL switch to MANUAL
  2. Depress BOOT switch
  3. Set all console switches down
  4. Set AUTO/MANUAL switch to AUTO
  5. When cursor appears, return switches to normal boot position

IDOS DISK STRUCTURE

All values shown for sector Address are in octal.

TYPE/USE	BEGIN SECTOR	END SECTOR
CONTROL DATA CORP. - 8270 2.5 meg DIABLO -(8230) 2.5 meg		
$DIR SPAN	0	037
BOOT & POST BOOT	0	04
IDOS DIRECTORY	05	034
SECTOR ALLOCATION TABLE	037	037
CONTROL DATA CORP. - (8270) 10 meg		
$DIR SPAN	0	077
BOOT & POST BOOT	0	04
IDOS DIRECTORY	05	073
SECTOR ALLOC TABLE	074	077
INFORMATION STORAGE SYSTEMS (8240)40 meg		
$DIR SPAN	0	0237
BOOT & POST BOOT	0	004
IDOS DIRECTORY	05	0212
SECTOR ALLOC TAB 6	0220	0237

CONTROL DATA CORP. (8260) 80 Meg

*SECTOR ADDRESS IS IDOS SECTOR NUMBER RELATIVE TO THE BEGINNING OF THE IDOS AREA ON THE PACK.

$DIR SPAN	0	0337
BOOT & POST BOOT	0	04
IDOS DIRECTORY	05	0271
SECTOR ALLOC TAB	0306	0337

THE DIRECTORY, BOOT & POST BOOT, AND SECTOR ALLOC. TABLES ARE INITIALIZED WITH THE IDOS PROCESSOR PACKIN - SEE IDOS UTILITIES MANUAL.

*BE AWARE:

IDOS REL AD31 and the previous IDOS releases AD30 and AD29 (E4 and F0 IDOS had a very similar Disc Structure. The primary difference is that the Boot Sector was at Sector 0 thru 03 leaving one additional Directory Sector. Disc Packs that used sector 04 for directory entries (Directories are built top down i.e. Full Directories) are not usable by IDOS AD32 or AD33.

COMMUNICATIONS REGION FORMAT - RELEASE D - AD33 COMPATIBLE

00000 - 00027	LDTAB	Each position represents a logical device number. bits 0-7 device ID number bits 13-21 channel and unit number
00030 - 00137	$ERROR	24 3-word error cells not used by IDOS
00140 - 00157	$USARA	16 words reserved for inter-program communication never disturbed by utilities
00160 -	$AREA	DCN 12*256+81*256+1 is default values for screen
00161 -		DCA " 12" lines
00162 -		DCA " 81" columns
00163 -		DCA " 1" number of screens
00164 -	SLOMEM	highest location in video RAM + 1 (0100000)
00165 - 00167		remainder of $AREA
00170 -	$CCK	DCN -6 tenth of second clock
00171 -	$CLOCK	cumulative clock
00172 -	$DATE	number of days since 1/1/76
00173 -	$TIME	number of 1/60 seconds since midnight
00174 -	EXTMEM	size of extended memory in words
00175 -	CPUTYP	0=IV/70, 1=IV/90I, 2=IV/90II, 3=IV/90III
00176 -	SYSNS	number of 1K screens (system), also named SYSSP
00177 -	SYSNLP	number of logical pages
00200 -	PROGNS	number of 1K screens (program specific)
00201 -	$MODE	0=single, 3=multiple
00202 -	$ACERR	accumulated error count from processor to processo
00203 -	$RCODE	return code from a processor
00204 -	$YSTCK	pointer to last control file SYSIN terminated
00205 -	$YSNP1	1st half of name of last processor run
00206 -	$YSNP2	2nd half
00207 -	$YSMAM	0=NPMAM not configured, 1=NPMAM conf, -1=don't kno
00210 -	$BOTID	CPU identification (NP/80 port number)
00211 -	$UCOMM	CHECK utility variable
00212 -	$VCOMM	"
00213 -	$WCOMM	"
00214 -	$XCOMM	"
00215 -	$YCOMM	"
00216 -	$ZCOMM	"
00217 -	$KBTYP	0=std 7200 ASCII A/N, 1=keypunch (029)
00220 -	$LASTP	last printer used (from LPOUT)
00221 -	$BSTOP	flag to stop $BATCH to enter alternate control fi
00222 -	$LUPR	0=don't convert low case to up for print, not 0=d
00223 -	$LOCKF	locked flag: 0=don't retry if asset locked, not 0=
00224 -	$BERR	automatic error control (AEC) stream flag
00225 -	$BSTK	AEC stack index
00226 -	$BPGN1	AEC name
00227 -	$BPGN2	" "
00230 -	$BETYP	0=run AEC after any program ends, 1=only if $RCOD
00231 -	$LRCOD	last non-zero $RCODE set
00232 -	$LRNP1	name of program with last $RCODE set
00233 -	$LRNP2	"
00234 - 00253		reserved for future enhanced forms control
00254 - 00265	$C500D	device control table for 8121 unit 00
00266 -	CPUID	CPU port number
00267 -	LPOUT	0=8135/2=8121/3=8131/4=NOPRNT/5=linepr/7=SPOOL
00270 -	CINPUT	0=Screen 0/1=unbufferred card rdr/-1= buffered ca
00271 -	LUCONV	0=don't convert keyboard 0 lower to upper/nonzero
00272 -	MEMORY	top of memory + 1
00273 -	$CARDM	card end of file marker (initialized to "// ")
00274 - 00307	FTABI	input file table
00310 - 00327	FTABO	output file table
00330 - 00356		I/O stack stuff for nesting control files 10 deep
00357 -	$CREND	BSS 0

IDOS Disc Directory, Allocation Table, Boot

- $BATCH/MONITR

Comparison	E3 (& After)	E2 (& Prior)
Name	$BATCH	MONITR
Location on Disc	A standard catalogued IDOS file	Resides in fixed location on disc
Communications Region	Not part of $BATCH, a standard catalogued IDOS file name $COMM	Part of MONITR
Printer Drivers	All printer drivers memory resident with $BATCH	Printer overlays only, LPOUT printer driver memry resident

- The Number of Disc Directory Entries

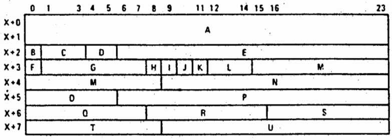
Disc Type	Directory Sectors		No. of Directory Entries	
	E3 (& After)	E2 (& Prior)	E3 (& After)	E2 (& Prior)
8320	05-034	020-34	767	351
8240	05-0212	020-0212	4287	3871
8260	05-0271	020-0271	5791	5375

NOTE: SUBTRACT 32 DECIMAL ENTRIES IF USING AD32 FOR ALL DISC TYPES

- The Execution of BOOT

Comparison	E3 (& After)	E2 (& Prior)
Sector 0	Contains initial boot program	Contains bootstrap
BOOT	1. Reads post boot program into memory 2. Zeros memory 3. Look up the location of $BATCH in directory. 4. Executes $BATCH	1. Zeros memory 2. Reads MONITR into memory. 3. Executes MONITR

IDOS Directory Format

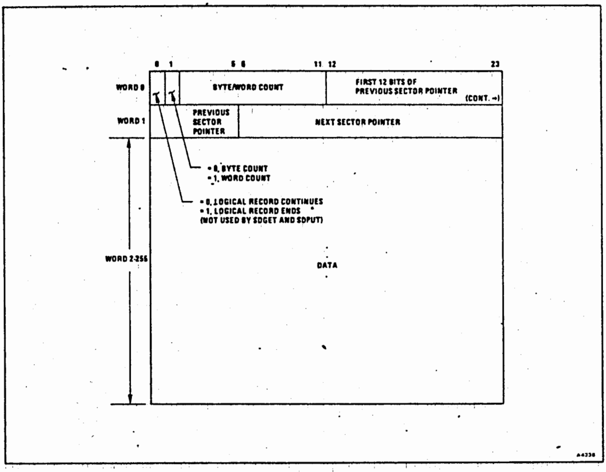


Word	Bits	Symbol	Meaning
X+0, X+1	0-23, 0-23	A	Six byte ASCII name with parity bits 0.
X+2	0	B	1 = Locked out to CPU's whose ID does not match. 0 = Allow access.
	1-3	C	Reserved
	4-5	D	Unused
	6-23	E	Ending sector address
X+3	0	F	1 = Protected 0 = Automatic deletion rule applies.
	1-7	G	Flag bytes (see DIRDMP)
	8	H	1 = Chained file 0 = Contiguous file
	9	I	1 = Load file 0 = Data file
	10	J	1 = Relocatable 0 = Not relocatable
	11	K	1 = ISAM 0 = Not ISAM
	12-14	L	CPU ID
	15-23	M	Sector count - 1 (high order bits)
X+4	0-8	M	Sector count - 1 (low order bits)
	9-23	N	Load point for a memory load.
X+5	0-5	O	Not used
	6-23	P	Starting sector address
X+6	0-7	Q	% space used during initial load. (ISAM only)
	8-15	R	Number of Index sectors/cylinder.
	16-23	S	Key length in bytes
X+7	0-8	T	Blocking factor (ISAM)
	9-23	U	Record length in bytes

A477D

Note: X+6 and X+7 are used by ISAM and RBS; X+6 is used by DISAM.

Sector Format of a Chained File



Error Bits in System Subroutines

	Meaning	File not opened	File already open	Null file	$FOPEN	$FCLOS	Incomplete Lost Record	File Relocatable	NP/80 Assignment List Full	End of File	File not chained	Linkage error	No more room on disc	Disc error	Allocator Locked Out	Directory Full	File protected	File not in Directory	File locked Out
	Bit	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Routine	$DGET									X		X		X					
	$DGETI										X	X							
	$DPUT												X	X	X				
	$DPUTF			X									X	X	X				
	$DPUTI												X	X	X				
	$DRUPD													X		X	X	X	X
	$DRFND								X					X				X	X
	$FCLOS	X		X		X							X	X	X	X	X	X	X
	$FOPEN		X		X			X	X		X		X	X	X		X	X	X
	$MDGET									X		X		X					
	$MDGTI										X								
	$MDPTI												X	X	X				
	$MDPTF			X									X	X	X				
	$MDPUT												X	X	X				
	GETREC	X					X			X	X	X		X					
	PUTREC	X											X	X	X				
	$ALLOC												X	X	X				
	$ALLC1												X	X	X				

A439D

Table 2 gives the meaning carried by each bit used by the system subroutines listed above.

Table 2. Error Bit Interpretations.

Bit #	Meaning
6	File not opened.
7	File already opened.
8	Null file.
9	$FOPEN indication.
10	$FCLOS indication.
11	Incomplete lost record.
12	File relocatable.
13	NP/80 assignment list full.*
14	End of file.
15	File not chained.
16	Linkage error.
17	No more room on disc.
18	Disc error.
19	Allocator locked out.*
20	Directory full.
21	File protected.
22	File not in directory.
23	File locked out.*
	(* New for AD33)

Clearing a Printer Halt Under IDOS - Rel "D" Level or Before

1. AUTO to MANUAL, RESET then STEP.
2. LOAD 0231 into RP
3. LOAD 04 into MEM
4. LOAD 0344 into RP
5. LOAD 0344 into MEM
6. LOAD 0336 into RP
7. LOAD 0336 into MEM
8. LOAD 72000446 into TIR
9. MANUAL to AUTO

Clearing a Printer Halt Under IDOS - Rel "E" Level and After

1. AUTO to MANUAL, RESET then STEP
2. LOAD 72000116 into TIR
3. MANUAL to AUTO

*** NOTE: LPOUT should never be manually set to seven (7=SPOOL)

$IDISC Request Table

Word	Bit	Symbol	Meaning
REQTAB+0 (status)	0	A	Request has been resolved and status is current
	1-5	B	Reserved
	6	C	Interrupt timer expired
	7-11	D	Reserved
	12	E	Incorrect number of words transferred
	13	F	Invalid request table
	14-16	G	Reserved
	17	H	Seek incomplete
	18	I	Head range error (nonexistent sector address requested)
	19	J	Header error
	20	K	Too late to transfer data
	21	L	CRC error
	22	M	Disc controller busy
	23	N	Drive not ready
REQTAB+1	0	O	1 = seek only 0 = seek, then read/write
	1	P	1 = read/write headers only 0 = normal read/write
	2-21	Q	Reserved (user should set to 0's)
	22-23	R	Physical drive number (must be between 0-3)
REQTAB+2	0	S	1 = return after resolution 0 = return after initiation
	1	T	Reserved (user should set to 0)
	2-8	U	Sector count minus 1
	9-23	V	Starting address of buffer
REQTAB+3	0	W	1 = write to disc 0 = read from disc
	1	X	1 = trigger interrupt on level 7 on resolution
	2	Y	1 = do not attempt retries 0 = retry as per ($DERC)
	3-5	Z	Reserved (user should set to 0's)
	6-23	a	Sector address (range = 0-06157)
REQTAB+4	0-23	b	Reserved for time-critical queuing

$JDISC Request Table

Word	Bit	Symbol	Meaning
REQTAB+0 (status)	0	A	Request has been resolved and status is posted
	1-4	B	Reserved
	5	C	$JDISC is busy at another interrupt level
	6	D	Lost interrupt (not applicable to R92AMI)
	7	E	Logical device number specified is not assigned in LDTAB
	8	F	Illegal device type (logical number specified is not a disc number; physical device type specified is not OK)
	9	G	NP/80 too busy (8261 only)
	10	H	Spurious ready interrupt since last request for drive (potential pack change), with REQTAB+3 bit 3 = 1 (8271 only)
	11	I	Losing rotations (R92AMI, 8241 only)
	12	J	Incorrect number of words transferred (8231, 8241, 8271 only)
	13	K	Invalid request table (sector address too high)
	14	L	Drive is set for read only (8241, 8271 only)
	15	M	Software error (8241) or system has 8230 controller and user attempts to use 8271 driver (8271 only)
	16	N	File unsafe (8241)
	17	O	Seek incomplete (8231, 8241, 8271 only)
			(8261) Bits 14-17 contain the sequence number of the failing IOB in the NP/80
	18	P	Head range error (8231, 8241, 8271) or Series IVinterface problem (8261)
	19	Q	Header error
	20	R	Too late to transfer (8231, 8241, 8271 only) or attempt to write to read-only virtual disc (8261)
	21	S	CRC error
	22	T	Disc controller busy (8231, 8241, 8271) or conflict detected in sign-on process when usingNPMAM (8261)
	23	U	Drive not ready or nonexistent (with bit 7, unassigned device, set)
REQTAB+1	0	V	1 = seek only 0 = seek then read/write
	1	W	1 = read/write headers only 0 = normal read/write (8231, 8241, 8271 only)
	2	X	1 = short read (R92AMI, 8241 only)
	3	Y	1 = stop short reads (R92AMI, 8241 only)
	4	Z	1 = allow access to any cylinder (8241 diagnostic purposes only)
	5	a	1 = bits 6-8 and 20-23 are physical drive description
	6-8	b	Physical device type if bit 5 = 1: 000 = 8231 001 = 8241 010 = 8261 011 = 8271
	9-16	c	Short read count minus 1 (R92AMI, 8241 only)
	17-18	d	Reserved (user should set to 0's)
	19-23	e	Device number: logical if bit 5 = 0; physical if bit 5 = 1
REQTAB+2	0	f	1 = return after resolution 0 = return after initiation
	1	g	Window number status: 0 = set by $JDISC at REQTAB+4 1 = preset by user at REQTAB+4 (extended memory processor only)
	2-8	h	Sector count minus 1
	9-23	i	Starting address of buffer
REQTAB+3	0	j	1 = write to disc 0 = read from disc
	1	k	1 = trigger a level 7 interrupt upon resolution (not applicable to R92AMI)
	2	l	1 = do not attempt retries 0 = retry errors as per $DERC (not applicable to R92AMI or 8261)
	3	m	Reject request and set status bit 10 (REQTAB+0) (8271 only)
	4-5	n	Reserved (user should set to 0)
	6-23	o	Sector address
REQTAB+4	0	p	Reserved
	1-8	q	Window address (extended memory processor only)
	9-23	r	Reserved for time-critical queuing and NP/80 use
REQTAB+5	0-23	s	Required only for R92AMI 8241 when short reads or header reads are to be performed. Not needed in other applications

SUMMARY OF COMMAND KEY FUNCTIONS

Key	Unshifted	Shifted	Controlled
TAB	Move cursor ahead to first colon; if none go home.	Move cursor backward to first colon; if none, go home.	Ignored
NEW LINE	Display new line symbol and advance cursor to first position of next line.	Move cursor to first position of current line.	Ignored
ENTER	Ignored	Display EOM symbol and then transfer data between start MI symbol and EOM symbol to 360/370.	Start digital clock display.
ERASE HOME	Move cursor to home position.	Erase screen and move cursor to home position.	Ignored
DEL ←	Backspace cursor one position.	Erase current character and move all characters between cursor and end of line (or new line symbol) backspace one position.	Blank current line; the cursor does not move.
INSRT →	Move cursor ahead one position.	Move all characters between cursor and end of line (or new line symbol) ahead one position.	Ignored
ROLL ↑	Move cursor up one line.	Move screen up one line with wraparound (top line moves to last line).	Delete current line, roll up all lines below, and blank bottom line.
ROLL ↓	Move cursor down one line.	Move screen down one line with wraparound (bottom line moves to first line).	Insert a blank line at current line by moving all lines down one line. Bottom line is lost.
TOTAL	Ignored	Same as unshifted.†	Same as unshifted.†
PRINT	Displays EOM symbol and then print data between home position and EOM symbol.	Same as unshifted.†	Same as unshifted.†
ERASE EOL	Erase all characters from cursor to end of current line.	Same as unshifted.†	Same as unshifted.†
ERASE EOS	Erase all characters from cursor to end of screen.	Same as unshifted.†	Same as unshifted.†
LOWER CASE	First time, switch keyboard to lower case. Second time, switch keyboard to upper case.	Same as unshifted.†	Same as unshifted.†
REST	Unlock keyboard.	Same as unshifted.†	Same as unshifted.†
START	Display start MI symbol at cursor location and advance cursor one position.	Same as unshifted.†	Same as unshifted.†
ATTN	Ignored	Same as unshifted.†	Same as unshifted.†
PASSW	Enter password to initiate supervisory mode.	Same as unshifted.†	Same as unshifted.†
SEE	Copy specified screen to first screen.	Same as unshifted.†	Same as unshifted.†
SHOW	Copy first screen to specified screen.	Same as unshifted.†	Same as unshifted.†
MONIT	All keystrokes at either keyboard appear on both screens.	Same as unshifted.†	Same as unshifted.†
REPT	When this key is used with any other key (except SHIFT or CTRL), the corresponding character or command is repeated nine times per second. Thus, repeat space moves the cursor across the screen at a rate of nine characters positions per second.		

† The SHIFT and CTRL keys have no effect on these keys.

A209A

Note: If START MI is already on the screen, erase between the start symbol and the cursor. Place the cursor to the right of the old START MI symbol.

2260 Simulator Video Gaps

Configuration word #2 in both the REMOTE and LOCAL Simulators reflects the video gaps on primary cables. Gapping is subject to the following rules:

1. There can be no vacant primary cable positions (i.e., if there are three primary cables, use cable positions 1-2-3 and NOT 1-2-4).

2. On systems with more than 12 primary cables, gaps cannot be specified for cables 13-16.

3. Gaps must be for the highest video areas on a primary cable and must be contiguous.

Video Size	48-Character	80-Character
Full (960) (40 x 24) (80 x 12)	Gaps not allowed	01-gap 2nd area
Half (480) (40 x 12) (80 x 6)	01-gap 2nd area	01-gap 4th area 10-gap 3rd and 4th areas 11-gap 2nd, 3rd and 4th areas
Quarter (240) (40 x 6)	01-gap 4th area 10-gap 3rd and 4th areas 11-gap 2nd, 3rd, and 4th areas	Not applicable

NOTE: Only the gapping bit patterns for the hardware configurations shown above are legal. Other combinations will give undefined results.

2260 Simulator Options

Supervisory Option

This option refers to the terminals on the system as being labeled from A to Z; therefore, only 26 videos can be supported. Note also that only video zero (labeled A) can enter supervisory mode. With this option, the following keys have the indicated functions:

PASSW	Depress this key and then enter the password (12345678) to enter supervisory mode. Pressing the key a second time end supervisory mode.
SEE	After pressing this key, the supervisor presses CTRL with the letter assigned to a video to see its contents.
SHOW	This key displays the contents of the supervisor's video on any other video. The video is indicated by pressing the CTRL and the letter key assigned to that video.
MONIT	This key operates the same as the SHOW and SEE keys, except that any key pressed on either keyboard will be displayed o both videos.

Clock Option

To activate the clock display, position the cursor at the screen location where the clock value is to be displayed. Depress the CTRL and EOM keys. The time is set by entering the value from the keyboard. The clock is deactivated by pressing the CTRL and EOM keys again.

Adding Machine Option

The adding machine option is activated by the TOTAL key and deactivated b the ATTN key. This option is only available on the 48-character Remote Simulator.

Card/Tape Media Package Option

The Media Package allows the user to do card-to-tape, card-to-print, and tape-to-print operations. Note: This cannot be done concurrently with the 2260 Remote Simulator because of timing conflicts between the I/O devices.

The Media Package Option is activated by the CTRL and INSERT keys. The option display will appear on the screen and the user may either utilize the default options or supply replacements. Use the TAB key to position the cursor on the next option and shifted TAB to go back to the previous option. Any options specified incorrectly are replaced by "???". When the options are correctly entered, depress the CTRL and INSERT keys to begin processing.

The options are:

Input	tape = TP card = CD
Output	tape = TP card = CD
Record Size	1 to maximum buffer size
Rewind	yes or no (tape only)
Blocked	1 to maximum buffer size
Number of Files	1 to 99 (tape only)
Carriage Control	yes or no for print tapes
Skip Files	number to be skipped (tape-to-print)
Card Deck End	the two characters used to specify the last card
Julian Date	used for tape and label writes
Lower Case	if data is to be printed in upper and lower case

Placing the 2260 Simulator on the 8250 Diskette

The normal 2260 Local or Remote system generation procedures are followed except that the Card/Tape Media Package Option cannot be supported and should not be generated. After system generation, the following procedur creates a distribution diskette:

1. Boot from the latest DKOS Master Pack, load a new diskette and execut the following:

```
// FLCOPY
/I=FMONTR,O=FMONTR,MONITOR,CLEAR,SIZE=(24, 48, 72 or 96).
//
```

   This step will place the DKOS monitor on the diskette, create an empt directory and reserve space for "DUMP" (the checkpoint file).

2. Execute the control file DKTGEN to place the DKOS utilities on the diskette.

3. Put the 2260 Master Pack containing the newly generated simulator on drive 1 and execute FLCOPY to transfer the simulator to the diskette. First, examine the load map to determine if 0300 words are available at the top of RAM. If so, the /ADD CHECKPOINT option is recommended to allow memory checkpoints to be taken. Also, the /BOOT parameter i recommended to auto-load the simulator after running two diagnostic routines when the diskette is bootstrapped.

```
// FLCOPY
/INPUT=A2260 (LOCAL) or B2260 (REMOTE) @1.
/OUTPUT=A2260 or B2260.
/ADD CHECKPOINT.  (If RAM is available)
/BOOT=A2260 or B2260.  (Optional)
//
```

   A minimum of three distribution diskettes should be prepared to be used as follows:

Copy 1	Daily use.
Copy 2	Immediate backup (used in case copy 1 is damaged).
Copy 3	Secondary backup (used only to create another copy in case copy 1 is damaged)

Placing the 2260 Simulator on Punched Cards

The following procedures will provide a deck of cards containing the 2260 Simulator object program.

1. Convert the absolute program to octal card format:

```
// DCCD
/INPUT=A2260 (LOCAL) or B2260 (REMOTE)
/OUTPUT=TEMP.
//
```

2. Create a card image tape. This program must be executed from a card deck or control file. The output is 80-byte unblocked records.

```
// COPYMD
/SOURCE=TEMP.
     (blank card)
```

At this point, key STE01 on keyboard 0, put switch 0 in the up position, and the disc/tape transfer will begin.

```
//
```

The output tape must be punched to 80-column cards on a mainframe using any tape/card utility. Place a 5-card loader in front of the deck and the simulator can be booted through a card reader.

Note: The first card punched must be discarded.

3. Patches can be applied to the simulator by inserting patch cards as explained under CDDC in the utilities section. For example, the config words can be changed with the following card format:

```
,TEM XXXXX + AAAAAAA WWWWWWWW XXXXXXXX YYYYYYY ZZZZZZZZ - AAAAAAA,

 ID  SEQ     ADDR    CW1      CW2   CW3   CW4     LOAD
```

The first configuration word is 13 words past the load address. The load address is dependent on the configuration:

Example: If the load address is 07540, then the first configuration word is 07553 and ADDR would be 0007553. This card would normally replace the last card.

2260 REMOTE Simulator - Hardware Configurations

Minimum requirements:

- CPU with 12K bytes of RAM on 7001; 24K on 4300, 4500
- Disc or card reader or diskette
- Async controller (8411)
- Keyboard

Maximum configurations:

RAM	CLK & SUP	MEDIA	FULL	HALF	QUARTER
12	No	No	4	8	16
18	Yes	No	8	16	32
24	Yes	No	8	16	32
24	No	Yes	8	16	32
48	Yes	No	32	32	32
48	No	Yes	32	32	32
72	Yes	Yes	32	32	32
96	Yes	Yes	32	32	32

Notes:

1. If a printer is added, the maximum number of screens possible drops b one.

2. If special gapping is used, subtract one from the maximum number of screens possible for each gap.

F5* DIAGNOSTIC KEY DISPLAY - REMOTE VERSION

1st Line: Octal Memory Dump

```
        00000                                00000000   00000000   00000000   00000000
L = Location in memory where dump starts,    1st word   2nd word   3rd word   4th word
    obtained from console switchs 11-23         L          L+1        L+2        L+3
```

2nd Line: Modem and Communications Display

```
        +0                  +2                          +5                          +7                       +9
        000                 000                         000                         000                      000

Asynch octal status     Response sent to 360 in oct 1   Code or Condition of Message   Last unexpected character   Code or position of
(STATM)                 (RESP)                          from 360 decimal (RCOMDT)      received from 360 octal     bad character deci-
                                                                                       (CHARE)                     mal (SEQUN)

1xx = Data Set Ready    201 = SOH                       001 = ACK received                                         001 = 2848 address
x4x = Clear to Send     202 = STX                       002 = READ FULL                                                  (bad or multi-
x2x = Carrier Detect    003 = ETX                       003 = Line Addr WRITE                                            drop)
x1x = Parity Error      204 = EOT                       004 = ERASE                                                002 = 2260 address
xx4 = Rate Error        006 = ACK                       005 = POLL                                                       (1xx or 3xx)
xx2 = XMIT Ready        225 = NAK                       006 = WRITE                                                003 = command (POLL,
xx1 = Char. Received    000 = Nothing sent yet          007 = Text received okay                                         etc.) ** or
000 = No Interrupt yet                                                                                                   expecting STX,
                                                        008 = Text received (err)                                        ACK, etc.
                                                        009 = Search for EOM                                             (parity may be
                                                              after NL                                                    bad)
                                                        00: = NAK received
                                                        00; = ERASE done (Send ACK)        ** 060 = READ FULL
                                                        00  = No response to 360              120 = Line Addr WRITE
                                                        000 = Nothing received yet            140 = ERASE
                                                                                              240 = POLL
                                                                                              300 = WRITE
```

* Each depression of the F5 key regenerates this display. Use repeat key to regenerate about 9-10 times per second.

NOTE: The second line is garbage until communications have been established.

Steps to Generate a REMOTE 2260 Simulator

1. Determine:

   a. 2848 Address. Use the following procedure to convert the customer's EBCDIC control unit address to an ASCII address used by Four-Phase. Note that bits are counted left to right from 0 through 7. Note also that the only valid EBCDIC addresses are those where bits 0 and 2 are the same.

      (1) Form the bit pattern of the 2848 EBCDIC address. For an address of x"A0", the word is 1010 0000.

      (2) Eliminate bit 2. Hex x"A0" would become 100 0000.

      (3) Add a parity bit in front of bit 0 to obtain even parity. The example of hex x"A0" would become 1100 0000.

      (4) Convert this value to octal. Thus x"A0" would become 0300. This is the value which should be entered into the configuration word(s) below.

      Note: The 2260 REMOTE Simulator only supports one 2848 address. Since an IBM 2848 supports only 16 screens and Four Phase can support 32, if multiple 2848s are being replaced, the customer must alter the polling and/or selecting lists in the communications software to incorporate the screens on the second 2848 as though they were on the first 2848.

   b. Number and type of videos and cable gapping information.

   c. Options desired by the customer. These include 029 keyboards, clock and supervisor functions, the card/tape media package, and the adding machine option (only available on 48-character systems.)

2. Use SIMED to patch the configuration words in the P161xx (P169xx for the card/tape media package) module where xx is the current version.

Word 1 bits:

0	Reserved, must be 0
1	ONLY USED IF THERE IS A PRINTER IN THE SYSTEM 0 = even number of videos 1 o odd number of videos
2	0 = no printer 1 = printer in system
3,4,5	Reserved, must be 0
6-8	001 = 960 char video (full) 010 = 480 char video (half) 100 = 240 char video (quarter)
9-14	Number of videos on system excluding printer or gaps
15	0 = no special gapping 1 = special gapping
16-23	Remote 2848 address (get value from above table)

Word 2 Bits: ONLY USED IF CONFIG WORD 1 BIT 15 IS 1 (Special Gapping)

0-1	Primary cable 0 gaps	12-13	Primary cable 6 gaps
2-3	Primary cable 1 gaps	14-15	Primary cable 7 gaps
4-5	Primary cable 2 gaps	16-17	Primary cable 8 gaps
6-7	Primary cable 3 gaps	18-19	Primary cable 9 gaps
8-9	Primary cable 4 gaps	20-21	Primary cable 10 gaps
10-11	Primary cable 5 gaps	22-23	Primary cable 11 gaps

Word 3 Bits:

0	0 = use bits 15-23 for buffer size 1 = use config word #1 screen size for printer buffer size
1-14	Reserved, must be 0
15-23	Print buffer size in words. May not exceed 320 (0500).

Word 4 Bits:

0-7	ONLY USED IF CONFIG WORD 1 BIT 15 IS 1 (Special Gapping) Total # tables to initialize, = to # tubes in config word 1. plus gaps in config word 2.
8-23	Reserved, must be 0

3. Run the following control files in the order specified:

// UNIVCR	Must be run even if printer not in system.
// STDKBR	Standard keyboard (7201)
or	
// 029KBR	029 data entry keyboard (7204)
// NCLSPR	No clock and supervisor functions
or	
// CLSPR	Include clock and supervisor functions (requires 18K)
// RRCDTP	No card/tape media package
or	
// ARCDTP	Include card/tape media package
// RADERR	No adding machine option or 81-character system
or	
// ADDERR	Include adding machine option (48-char system only)
// CFREMx	Where x is from this table of supported versions

x	CHAR	RAM	LOAD
CFREMA	40	12	03360
CFREMB	80	12	03660
CFREMC	40	18/24	07420
CFREMD	80	18/24	07540
CFREME	40	48	06660
CFREMF	80	48	07540
CFREMG	40	72/96	06660
CFREMH	80	72/96	07540

4. The output of the CFREMx step is the remote simulator (B2260). See the separate descriptions in this section for instructions on putting B2260 on cards or diskette.

5. The simulator may be tested as follows:

   If switch 0 is up, the simulator will halt after loading to allow for changes. The configuration words are located eight words past the halt address. Note: If the test CPU does not have an async board, the 2848 address can be changed to a 0 to avoid a cut-word hang. If the test CPU does not have a printer, change bit 2 of configuration word 1 to a zero to avoid a cut-word hang.

Initialization Constants (Config Words)

There are four (4) CONFIG words for each version of the Simulator (LOCAL and REMOTE) which reside in the Initialization source modules:

Media Package Included		Media Package Excluded	
Local:	P47-9X	Local:	P47-1Y
Remote:	P169XY	Remote:	P161YZ

The CONFIG words also reside in the absolute load modules B2260 (REMOTE) and A2260 (LOCAL).

There are several methods by which the CONFIG can be altered to suit a user's requirements:

1. Source code alteration.
2. Absolute load module alteration.
3. Octal card deck "patch" cards.

Source Code Alteration Method

Use either SIMED or SNEDIT to perform the alteration process. Select the appropriate source module for LOCAL or REMOTE and with or without the Card Tape Media Package. Enter Edit Mode and locate the four (4) DCN values immediately following

NON-MEDIA LOCAL OR REMOTE	MEDIA LOCAL OR REMOTE
the BAL B2848 instruction located eleven (11) words past the START label in both the P47-1Y and P161YZ modules.	the BAL CB2848 instruction located three (3) word past the :BEGIN label in P47-9Y and one (1) word past the :BEGIN label in P169YZ.

Enter the appropriate bit patterns for the CONFIG words and exit SIMED in the normal manner. Reassemble the altered source module using the DOS program // ASM and re-execute a complete SYSGEN.

Absolute Load Module Alteration Method

Place console switch 0 up before entering // B2260 (Remote) or // A2260 (Local) and pressing the EOM key and the system will halt at RP+2 past the HLT $ instruction. Note the contents of the RP register. Select MEM and STEP through the program until the BAL B2848 (Non-media LOCAL/REMOTE) or the BAL CB2848 (Media LOCAL/REMOTE) is encountered. It can be recognized by the octal 66 in the first six bits. The next four locations contain the CONFIG words. Enter the appropriate constant in each location with the console switch keys, pressing LOAD and STEP after each entry. After entering the CONFIG word constants, select TIR and enter a BRA RP-1 where RP equals the value of the "initial halt". Select AUTO and the program will begin execution by clearing all screens and displaying a blinking cursor in the HOME position.

Octal Card Deck "Patch" Cards Alteration Method

Octal card decks are created with the DOS utility DCCD. This utility creates a card image file on disc. The file is then copied to tape using the DOS utility COPYMD. The tape must then be punched into 80-column cards on an IBM 360/370.

```
// DCCD
/I=A2260 (LOCAL) or B2260 (REMOTE)
/O=TEMP.
//
// COPYMD (creates a card image tape)
/ SOURCE = TEMP.
/ TAPE = TAPE.
    (blank card)
```

At this point, key STE01 on keyboard 0, put switch 0 in the up position, and the disc/tape transfer will begin.

```
//
```

b

Note: The output tape must be punched to 80-column cards on a main-frame using any tape/card utility. Place a 5 card loader in front of the deck and it can be booted through a card reader.

The following points should be noted about the format of the octal card deck:

1. Each card begins with the three letters which are the first three letters of the input file name. If input is for LOCAL Simulator, the first three letters would be 'A22'. For the REMOTE Simulator, they would be 'B22'.
2. Columns 4-8 are the card deck sequence numbers starting with zero and incremented by one for each successive card. The sequence numbers ar in OCTAL.
3. Columns 9-72 contain contiguous eight digit octal representations of instructions, data, and/or conrol statements. These eight digit representations may be one of four types:
    a. True data and/or instructions (72001577).
    b. Origin statements, first digit is a "+" or "&".
    c. Check-sum values, first digit 'P' through 'W'.
    d. Transfer values, first digit '-' followed by an address.
4. Columns 73-78 of the next to last card contains the check-sum value with zeroes padded through the card.

2260 LOCAL Simulator - Hardware Configurations

Minimum requirements:

CPU with 12K bytes of RAM
Disc or card reader or diskette
Channel adapter (7071 or 7072)

Maximum configurations:

RAM	CLK & SUP	MEDIA	FULL	HALF	QUARTER
12	No	No	4	8	16
18	Yes	No	8	16	32
24	Yes	No	8	16	32
24	No	Yes	8	16	32
48	Yes	No	32	32	32
48	No	Yes	32	32	32
72	Yes	Yes	32	32	32
96	Yes	Yes	32	32	32

Notes:

1. If a printer is added, the maximum number of screens possible drops by one.
2. If special gapping is used, subtract one from the maximum number of screens possible for each gap.

Steps to Generate a LOCAL 2260 Simulator

1. Determine:

   a. 2848 address (Note: The local simulator allows 1 or 2 addresses)

   b. Number and type of videos and any cable gapping information.

   c. Options desired by the customer. These include 029 keyboards, clock and supervisor functions, and the card/tape media package.

2. Use SIMED to patch the four configuration words in the P47-1x (P47-9x for the card/tape media package) module where x is the current version

Word 1 Bits:

0	0 = one 2848 address 1 = two 2848 addresses
1	Reserved, must be 0
2	0 = no printer 1 = printer in system
3,4,5	Reserved, must be 0
6-8	001 = 960 character video (full) 010 = 480 character video (half) 100 = 240 character video (quarter)
9-14	Number of videos on system
15	0 = no special gapping 1 = special gapping
16-23	Low 2848 hex address (IBM EBCDIC without conversion)

Word 2 Bits: ONLY USED IF CONFIG WORD 1 BIT 15 IS 1 (Special Gapping)
Note: Each gap reduces by one the number of videos supported. (S
"2260 Simulator Gaps" in Section AE - 2260 Remote)

0-1	Primary cable 0 gaps	12-13	Primary cable 6 gaps
2-3	Primary cable 1 gaps	14-15	Primary cable 7 gaps
4-5	Primary cable 2 gaps	16-17	Primary cable 8 gaps
6-7	Primary cable 3 gaps	18-19	Primary cable 9 gaps
8-9	Primary cable 4 gaps	20-21	Primary cable 10 gaps
10-11	Primary cable 5 gaps	22-23	Primary cable 11 gaps

Word 3 Bits: ONLY USED IF CONFIG WORD 1 BIT 16 is 1 (Printer in System)

0	0 = use bits 15-23 for buffer size 1 = use config word 1 screen size for print buffer size
1-14	Reserved, must be 0
15-23	Print buffer size in words. May not exceed 320 (0500).

Word 4 Bits:

0-7	ONLY USED IF CONFIG WORD 1 BIT 15 IS 1 (Special Gapping) Total # tables to initialize, = to # tubes in config word 1, plus # gaps in config word 2.
8-15	ONLY USED IF CONFIG WORD 1 BIT 0 IS 1 (Two 2848 Addresses) number of tubes on the low 2848 address.
16-23	ONLY USED IF CONFIG WORD 1 BIT 0 IS 1 (Two 2848 Addresses) Address in hex of the high 2848 addresses (x"B0" = 1011 0000

3. Run the following control files in the order specified:

// UNIVAC	Must be run to define printer (even if not in system)
// STDKBD	Standard keyboard (7201)
or	
// 029KBD	029 data entry keyboard (7204)
// RSPRCL	No clock and supervisor functions
or	
// ASPRCL	Include clock and supervisor functions (requires 18K)
// RLCDTP	No card/tape media package
or	
// ALCDTP	Include card/tape media package (requires 24K)
// CFLOCx	Where x is from this table of supported versions

x	CHAR	RAM	LOAD
CFLOCA	40	12	03660
CFLOCB	80	12	03660
CFLOCC	40	18/24	07540
CFLOCD	80	18/24	07420
CFLOCE	40	48	06660
CFLOCF	80	48	07540
CFLOCG	40	72/96	06660
CFLOCH	80	72/96	07540

4. The output of the CFLOCx step is the local simulator (A2260). See the instructions under 2260 Remote (Section AE) to put A2260 on cards or diskette.

5. The simulator may be tested as follows:

   a. If switch 0 is up, the simulator will halt after loading to allow for changes. The configuration words are located 8 words past the halt address. If no printer is on the test CPU, set config word 1, bit 2 to 0 to avoid a cut-word hang.

   b. If no local channel is on the test CPU, when a cut-word hang occurs, press RESET and STEP, decrement RP by 1, and NOP the I/O instruction there. Then LOAD a branch to that location into TIR. This cut-word hang will occur twice.

2260 LOCAL Simulator - Channel Debugging

When channel failures occur, there are several locations to note to aid in problem isolation. The steps to be taken when a failure occurs are:

```
***********************************************
*    DO NOT HIT SYSTEM RESET AT ANY TIME      *
***********************************************
```

Step 1. Stop all keyboard activity on the system. This will preserve the state of the system when the failure occurred.

Step 2. See if the F5 Diagnostic Dump works. If the F5 dump does not work on one keyboard, try another.

Step 3. Place the system in MANUAL.

Step 4. Write down the contents of all the registers.

Step 5. Read out the contents of the following locations:

LOCATION	VERSION	MEANING
01017*	40 Char.	New command status
01022*	40 Char.	Control word
01023*	40 Char.	Last status received

* For an 80 character system, add 0600

Step 6. Obtain the current status of the channel by placing 67701004* in TIR and depress STEP. Read out location 1024* which contains the status word.

Step 7. Obtain the diagnostic status (4 words) of the Channel Adapter.

a. LOAD 40000000 into location 1023*.

b. LOAD TIR with 67701004* and press STEP.

c. Load TIR with 67701006* and press STEP.

d. Read out and note the contents of the following locations:

LOCATION	VERSION	MEANING
01024*	40 Char.	Diagnostic word 0
01025*	40 Char.	Diagnostic word 1
01026*	40 Char.	Diagnostic word 2
01027*	40 Char.	Diagnostic word 3

* For an 80 character system, add 0600.

Step 8. Obtain the contents of the following locations:

LOCATION	VERSION	MEANING
01015*	40 Char.	Location of the user table.
01020*	40 Char.	Write Flag. If =0, means write operation.
01021*	40 Char.	Read Flag. If =0, means read operation.
01046*	40 Char.	Word location of buffer where next transmission will start.
01037*	40 Char.	Last data word read from 7071.
01014*	40 Char.	2848 Address
01035*	40 Char.	If a Halt, contents indicates where from.
053	BOTH	General I/O instruction pointing tc a 2-word pair.

Step 9. If a KEYBOARD is LOCKED, obtain the listed User Table information for that keyboard.

Step 10. If possible, obtain the 360/370 System Status information by execution of SEREP (System Environment Recording, Editing, and Printing). SEREP is part of the 360/370 Diagnostics Package which provides for printing 360/370 system status information.

Step 11. If execution of SEREP is not possible, obtain the contents of the following hexidecimal locations if the 360/370 is hung in a hard wait.

Hex location 3B and 40-47.

CHANNEL APAPTOR DIAGNOSTIC STATUS WORDS

Bit	Signal			
	Word 0	Word 1	Word 2	Word 3
0	BOOT Address Equal	Interrupt	Interrupt Request 1 & 2	Data Interrupt
1	Upper ≠ and Lower ≠	Upper Less or Equal	Printer Address Equal	Upper Less
2	---	---	Control Unit Request	Clocked Read Buffer Full
3	---	BOOT Address Ok	Channel Request	Write Buffer Full
4	IV/70 Buffer Full	Read Buffer Full	Device Busy	Allow Write
5	Convert	Status Request	Control Unit Busy	Request Next Word
6	Bus Out Parity Ok	Service Request	Adapter Busy	Continue
7	Not Enable Bus In	Printer Intervention Required	Interface Busy	End Op
8	DB1-0	Address Ok	Proceed	Hold/Select Out Gated
9	DB1-1	Test I/O	Map	Allow Bus
10	DB1-2	No Op	360 System Reset	Stop
11	DB1-3	Sense	Load Line Address Register	Allow Request In
12	DB1-4	Gate Address to Address Register	Attention	Attention Accepted
13	DB1-5	Printer Busy	Done	Under Flow
14	DB1-6	Intervention Required	Byte Counter Lo-A	First Byte
15	DB1-7	Gate Bus In-A	Byte Counter Lo-B	Read
16	Internal Bus Out-0	Gate Bus In-B	Byte Counter Lo-C	Write
17	Internal Bus Out-1	Bus Out Check	Byte Counter Lo-D	Stack
18	Internal Bus Out-2	Command Reject	Byte Position 0	Bus and Status
19	Internal Bus Out-3	Control Unit End	Byte Position 1	Read Special
20	Internal Bus Out-4	Command Chain	Byte Position 2	Printer Request Device End
21	Internal Bus Out-5	Channel End	Load Byte 0	EOM Character IV/70
22	Internal Bus Out-6	Device End	Load Byte 1	NL Character From IV/70
23	Internal Bus Out-7	Unit Check	Load Byte 2	NL Character From Channel

F5* Diagnostic Key Display - Local Version

1st Line: Octal Memory Dump

```
        00000               00000000  00000000  00000000  00000000


L=Location in memory where    1st word  2nd word  3rd word  4th word
  dump starts, obtained          L        L+1       L+2       L+3
  from console switches
11-23.
```

Each depression of the F5 key regenerates this display. Use the REPT key to regenerate the display about 9-10 times per second.

TABLE OF CONTENTS

		PAGE
SECTION I	SYSTEM CONFIGURATION	
1	Terminal Support	AG15-2
2	Simulator Configuration Words	AG15-2
3	A76 Configurator	AG15-3
4	Execution Time Reconfiguration	AG15-5
5	MINIGEN	AG15-6
6	Store and Forward	AG15-8
7	TBLEDT	AG15-9
SECTION II	NTP/150	
1	SB3270 Interface	AG15-10
2	Entry Codes	AG15-10
3	Detail Codes	AG15-11
4	NTP/150 Subroutines	AG15-13
5	SBRSET	AG15-14
6	Software Action Codes	AG15-15
SECTION III	DEBUGGING	
1	Escape Key Functions	AG15-16
2	Display of Attribute Characters	AG15-17
3	Display of Line Control Characters	AG15-18
4	TRC327 Execution	AG15-19
5	TRC327 Output	AG15-20
6	Taking Memory Dumps	AG15-22
7	DMP327 Execution	AG15-22
8	DMP327 Output	AG15-23
SECTION IV	BISYNC and 3270 PROTOCOL	
1	3270 Control Characters	AG15-27
2	Bisync Data Link Control	AG15-28
3	3270 Message Formats	AG15-29
4	Local and Remote Command Codes	AG15-30
5	Write and Copy Control Characters	AG15-30
6	Buffer Control Orders	AG15-31
7	Attribute Bytes	AG15-31
8	Attention ID Byte	AG15-32
9	Remote Status/Sense Bytes	AG15-33
10	Local Status and Sense Bytes	AG15-35
SECTION V	COMMUNICATION CONTROLLERS	
1	8436-2 Status and Control	AG15-36
2	7073 Status, Control, and IOID's	AG15-37
3	7074 Status and Control (WIDGET)	AG15-39
SECTION VI	MISCELLANEOUS TOPICS	
1	Bypassing Communication Controller Halt	AG15-40
2	Adjustment of Dual Intensity	AG15-40
3	Marketing Considerations	AG15-40

SECTION I SYSTEM CONFIGURATION

1.1 TERMINAL SUPPORT

memory size	24K			48K			72K			96K		
screen size	480	960	1920	480	960	1920	480	960	1920	480	960	1920
REMOTE:												
no printers	5	3	1	32	19	10	32	32	18	32	32	24*
8121's	3	1	1	30	18	9	32	32	17	32	32	25*
line printers	3	2	1	31	18	10	32	32	18	32	32	26*
both printers	2	1	1	29	17	9	32	32	17	32	32	25*
EMOTE Store and Forward:												
no printers	0	-	0	18	-	6	32	-	14	32	-	22
8121's	0	-	0	15	-	5	32	-	13	32	-	21
line printers	0	-	0	16	-	5	32	-	13	32	-	21
both printers	0	-	0	15	-	4	32	-	13	32	-	21
LOCAL:												
no printers	6	3	2	32	19	10	32	32	18	32	32	26*
8121's	3	2	1	30	18	9	32	32	17	32	32	26*
line printers	4	2	1	31	19	10	32	32	18	32	32	26*
both printers	2	1	1	30	18	9	32	32	17	32	32	26*
LOCAL Store and Forward:												
no printers	0	-	0	18	-	6	32	-	14	32	-	22
8121's	0	-	0	15	-	5	32	-	13	32	-	21
line printers	0	-	0	17	-	6	32	-	14	32	-	21
both printers	0	-	0	15	-	5	32	-	13	32	-	21

* - Note the limitation of 24 1920 VDU's

1.2 SIMULATOR CONFIGURATION WORDS

Terminal Polling Byte from Host System:

x1..	Bit 0 determined by bits 2 - 7.
..1.	Selection for control unit only.
...x xxxx	Device address entered into A76.

Configuration Word:

x..	1 = LOCAL, 0 = REMOTE.
.1.	NTP/150.
..1	Store and Forward.
... 1..	Line printers configured.
... .1.	8121 printers configured.
... ..x x..	Communication code used:
0 0	EBCDIC.
0 1	Invalid.
1 0	ASCII-A.
1 1	ASCII-B.
... ... xxx xxx	Number of terminals configured.

1.3 NTP100/150 CONFIGURATOR (A76)

```
// A76
/INPUT = xxxxxx@y Input file (optional, default is DEFAULT@0)
/OUTPUT = xxxxxx Output file (optional, default is CFGFIL)
/AUTOMATIC = xx.
//
```

If /A is specified, then the program operates as though SHIFTED DOWN ARROW were pressed repeatedly.

A76 creates and executes control files "C77G-C", which assembles the conversion table source file "P77-TA", and "C77G-E", which loads the 3270 simulator.

Keyboard Input to A76:

A76 page 1:

```
+---------------------------------------------------+
+                                                   +
+                                                   +
+ LINE LENGTH OF THIS SYSTEM 80;                    +
+ INPUT FILE DEFAULT@0;  OUTPUT FILE CFGFIL         +
+                                                   +
+                                                   +
+                                                   +
+                                                   +
+ PRESS SHIFTED DOWN ARROW TO ADVANCE TO NEXT PAGE +
+---------------------------------------------------+
```

LINE LENGTH OF SYSTEM	Enter 40 or 80.
INPUT FILE	Enter input configuration file and drive.
OUTPUT FILE	Enter output configuration file.
	(CFG327 keyboard entries cont.)

A76 page 2:

```
+---------------------------------------------------+
+ APPLICATION NTP100; REMOTE; COBOL @ FIELD? N/A;   +
+ MEMORY SIZE IS 72K BYTES; SCREEN SIZE IS 80x24;   +
+ TRANSMISSION CODE(EBCDIC/ASCIIA/ASCIIB)? EBCDIC;  +
+ LINE SPEED IS 4800 BPS; CONTROL UNIT ADDRESS 40;  +
+ MAXIMUM MESSAGE LENGTH (WITH ORDER BYTES) 2000;   +
+ DEBUG? N; STRING EDITOR DEBUG? N/A                +
+ LOG LENGTH N/A;  ; CHECKPOINT? N; CKPT DEV N/A;   +
+ TAB TO COLON? N;  STORE AND FORWARD? N;           +
+                                                   +
+                                                   +
+ PRESS SHIFTED RIGHT ARROW TO ACCEPT A FIELD       +
+---------------------------------------------------+
```

APPLICATION	Enter 100 or 150.
REMOTE	Enter REMOTE or LOCAL.
COBOL @ FIELD	Enter Y or N.
MEMORY SIZE	Enter 24, 48, 72, or 96.
SCREEN SIZE	Enter 80x24, 80x12, or 40x12.
TRANSMISSION CODE	Enter EBCDIC, ASCIIA, or ASCIIB.
LINE SPEED	Enter 24, 48, 72, or 96.
CONTROL UNIT ADDRESS	Enter local control unit address or bisync control unit polling address or SNA physical unit address.
MAXIMUM MESSAGE LENGTH	Enter length of longest message expected.
DEBUG	Enter Y or N.
STRING EDITOR DEBUG	Enter Y or N.
LOG LENGTH	Enter 0 to use all available memory. Compute log area size to allow for NTP/150 overlays.
CHECKPOINT	Enter Y or N.
CKPT DEV	Enter 8230, 8240, 8250, Or 8260.
TAB TO COLON	Enter Y or N.
STORE AND FORWARD	Enter Y or N.
LINE DISCIPLINE	Enter BSC or SDLC. (N/A for LOCAL)
NRZI (Non Return to Zero Inverted)	Enter Y or N to match the specification in the NCP 3704 or 3705 sysgen.

A76 page 3:

```
+-----------------------------------------------------+
+ KEYBOARD TYPE 0 IS PRINTER ONLY;  TYPE - = DONE     +
+ KEYBOARD TYPE 1 IS 7226;                            +
+ LOWER CASE? N; NUMERIC LOCK? Y;                     +
+ KEYBOARD TYPE 2 IS 7200;                            +
+ LOWER CASE? Y; NUMERIC LOCK? N;                     +
+ TERMINAL  1 USES KEYBOARD TYPE 1 (0,1,2,-);         +
+ IN SCREEN POSITION 01;     WITH POLL ADDRESS 40;    +
+ AUDIBLE ALARM? Y;   INITIAL INTERCEPT FLAG? N/A;    +
+ PRINTER TYPE 81XX  (XX=NONE, 21=8121, LP=OTHER);    +
+          AT HARDWARE ADDRESS N/A;                   +
+                                                     +
+ PRESS SHIFTED DOWN ARROW TO ADVANCE TO NEXT PAGE    +
+-----------------------------------------------------+
```

KEYBOARD TYPE 1	Enter 7200, 7203, 7226, 7227, 7228, or U0-U9.
LOWER CASE	Enter Y or N.
NUMERIC LOCK	Enter Y or N.
KEYBOARD TYPE 2	(same as for keyboard type 1)

Repeat the following entries for each terminal:

TERMINAL n USES KEYBOARD TYPE	Enter 0 for printer 1 for keyboard type 1 2 for keyboard type 2 - for no more terminals.
SCREEN POSITION	Enter 01 - 32 (01 - 24 LOCAL).
POLL ADDRESS	Enter local device address (LOCAL) . BSC device polling address (BSC) SNA logical unit address (SDLC)
AUDIBLE ALARM	Enter Y or N.
INITIAL INTERCEPT	Enter Y or N.
PRINTER TYPE	Enter 8121 or 81LP.
AT HARDWARE ADDRESS	Enter octal 000 - 037 (8121) 030 - 031 (81LP)

A76 page 4:

```
+-----------------------------------------------------+
+ IS LOWER CASE USED BY PRINTERS? N/A;                +
+ WHAT IS THE 8121 DEFAULT FOR LINES/PAGE? N/A;       +
+ WILL SB3270 OPEN FILES? N/A;                        +
+ NUMBER OF DISC RETRIES? N/A;                        +
+                                                     +
+ LOAD MODULE NAME? SIM327;                           +
+                                                     +
+                                                     +
+ PRESS SHIFTED UP ARROW TO REVIEW CONFIGURATION      +
+    OR PRESS SHIFT DOWN ARROW TO FINISH              +
+-----------------------------------------------------+
```

LOWER CASE BY PRINTERS	Enter Y or N.
DEFAULT LINES/PAGE	Enter 01 - 99.
SB3270 OPEN FILES	Enter Y or N.
DISC RETRIES	Enter 0 - 9.
LOAD MODULE NAME	Enter any valid IDOS file name.

I.4 EXECUTION TIME RECONFIGURATION

With console key 0 up :

1. Select RAM, the control unit address will be displayed in the console lights right adjusted in following formats.

   Local - the most significant 4 bits of the hex address (50 = 5).
   Remote - the least significant 5 bits of hex address (5F = 1F).

2. Keeping RAM selected depress the step switch and release. A word representing the description of the first terminal will be displayed.

Terminal Descriptor Word

0xx xxx	Video display area assignment.
... ... 1..	Initial cobol intercept flag.
... ... .xx xxx	Position in polling list.
... ... 1..	Terminal has keyboard.
... ...x	Keyboard type:
0	Type 1.
1	Type 2.
... ... 1..	Audible alarm.
... ... .1.	Line printer.
... ...1	8121 printer.
... ... xxx xxx	Printer hardware address.

After the terminal description is entered, keeping MEM selected toggle STEP. This will cause the next terminal description to be displayed. All console lights on idicates the end of the terminal description list.

3. To change the poll delay select RA with the console switches. The poll delay is displayed as a positive octal value representing tenths of a second. This delay must not exceed 3 seconds. Set the console switches to the new value and press LOAD. Be sure switch 0 is down for this step.

4. Set register select switches to TIR and return to AUTO. If switch 0 is up, another halt will occur.

5. The second halt allows the initialization of communications to be bypassed. Move AUTO/MAUAL to MANUAL, select MEM, set a NOP (eg. 06700000) in the console keys, and press LOAD. Reset the register select switchs to TIR and return to AUTO mode.

I.5 MINIGEN

The MINIGEN program permits modification to an existing NTP/100 or 150 configuration without performing a new configuration procedure.

MINIGEN allows the user to modify:
- Control Unit Address
- 8121 Character Printer default page size
- Terminal Characteristics

MINIGEN can create any number of MINIGEN parameter load modules (MPLM) with different configurations. Each MPLM is stored as a different program file. Execution of an MPLM file updates the configuration table in a 3270 absolute load module.

Restrictions:

- MINIGEN must not be used on releases prior to G0.
- MINIGEN must not add more terminals to the original configuration.
- Before altering a configuration, store the original system configuration as an MPLM.
- Alter default page size for 8121 character printers only if the original configuration included at least one 8121.
- Do not configure a system with all terminals deleted.
- Video gapping may be altered unless it increases the absolute load address of the Simulator.
- If a local 3270 control unit address is changed, all terminal addresses must be changed.

```
// MINGNx      x = 3 for IDOS or 5 for DKOS
```

Keyboard Controls

CTRL E	Exit from MINIGEN
SHIFTED ↑	Restart MINIGEN at first display. All previous changes are saved
SHIFTED ←	Skip back one field
SHIFTED →	Skip forward one field
SHIFTED ↓	Store the current page and skip to the next page
RESET	Unlock the keyboard after an erroneous entry and position cursor within a field

(MINIGEN keyboard entries cont.)

Page 1 Entries - File Specifications

```
/I=filename-a,/O=filename-6
/U=filename-c
//
```

/INPUT file can only be on MPLM file. It cannot be modified directly. Using the same name for both /I and /O file has the effect of modifying the /I file. To examine, but not modify an existing MPLM, just specify . The update file must be the NTP/100 or NTP/150 load module file -- ex. A77C. It is used as both the source file and destination file to receive the modified configuration table. When both /I and /O files are specified, MINIGEN uses /I as the base configuration and alter the NTP load module.

Page 2 Display - File Verification

```
File Name
    XXXXXX    EXISTING MPLM
    XXXXXX    MPLM TO BE CREATED
    XXXXXX    NTP100 or NTP150 LOAD MODULE TO UPDATE
    GO TO NEXT PAGE (Y/N)
```

X's appear where file names were not provided.

Page 3 Display - CU_Address and Page Length

Page three allows the user to alter the control unit address and/or the default lines per page for 8121 printers.

Page 4 Display - Terminal Descriptors

Page four displays the terminal descriptor list. This display is repeated one time for each terminal in the configuration.

Format:

```
TERMINAL XX USES KEYBOARD TYPE X (0, 1, 2) IN SCREEN
POSITION XX WITH POLL ADDRESS XX AUDIBLE ALARM (Y,N)
X INITIAL COBOL INTR X PRINTER TYPE 81XX (NO, 21=8121)
LP=81LP) AT HARDWARE ADDRESS 0XX.
```

Page 5 Display - Program Termination

Format:

```
END OF LIST
HIT U TO UPDATE DISC AND EXIT
HIT CTRL E TO EXIT AND NOT UPDATE
HIT SHFT UP ARROW TO RESTART
```

To construct a Store and Forward DF3270 disc file, enter:

```
    // AF3270
    /OUTPUT FILE = DF3270 @ drive 0.
    /VIDEO 'TYPE = 480 BYTES/SCREEN
                           or
                1920 BYTES/SCREEN
    /TERMINALS CONFIGURED = # of terminals.
    /FORMAT STORAGE = # of formats.
    /SCREEN IMAGE STORAGE = # of screens
 //
```

The Store and Forward routines access the simulator by means of the NTP/150 Subroutines and therefore execute in the background. Even though it uses NTP/150 subroutines, it is configured as an NTP/100 system.

The Store and Forward function keys:

- CTRL G - Store the format on the screen into the DF3270 file and catalog the name in the DF3270 directory.
- CTRL H - Delete the format name from the DF3270 directory and delete the image from the DF3270 file.
- CTRL R - Display the first completed form image in the terminal's queue.
- CTRL N - Delete the first completed form image in the terminals queue and display the next image.
- CTRL F - Find the format name in DF3270 directory and display the format on the video screen.
- CTRL S - Store the completed form image on the video screen into the terminal's queue.

When executing A76 to configure NTP/100 Store and Forward System, enter the keyboard type = U5.

The simulator must be in debug mode (ESC key-P - PASSWORD) to create or delete format images.

The internal 3270 keyboard code tables must be modified so that these keystrokes invoke Store and Forward functions. This is done by the utility program TBLEDT.

Listing of T77AC2

```
Col
1

016253   CTRL-N to Entry Code 1
022254   CTRL-R to Entry Code 2
023255   CTRL-S to Entry Code 3
006256   CTRL-F to Entry Code 4
007257   CTRL-G to Entry Code 5
010260   CTRL-H to Entry Code 6
```

```
// TBLEDT
/INPUT TABLE = T77ATP
/OUTPUT TABLE - T77LU5
/CHANGE = T77AC2
// TBLEDT
/INPUT TABLE = T77ATQ
/OUTPUT TABLE = T77LU5
/CHANGE = T77AC2
// TBLEDT
/INPUT TABLE = T77ATR
/OUTPUT TABLE = T77XU5
/CHANGE = T77AC2.
//
```

When executing A76 to configure NTP/100 Store and Forward System, enter the keyboard type = U5.

The simulator must be in debug mode (ESC key-P - PASSWORD) to create or delete format images.

II.1 SB3270 INTERFACE

```
LINKAGE SECTION.
01 USER-SCREEN.
   02 USER-LINE OCCURS 24 TIMES.
      03 SCREEN-LINE PICTURE X(80).
      03 FILLER      PICTURE X(16).

01 USER-CURSOR.
   02 USER-ROWID     PICTURE S9(6) COMPUTATIONAL.
   02 USER-COLID     PICTURE S9(6) COMPUTATIONAL.

01 USER-SUBSCRIPT    PICTURE S9(6) COMPUTATIONAL.

01 ENTRY-CODE        PICTURE S9(6) COMPUTATIONAL.

01 DETAIL-CODE       PICTURE S9(6) COMPUTATIONAL.



PROCEDURE DIVISION USING
     USER-SCREEN,
     USER-CURSOR,
     USER-SUBSCRIPT,
     ENTRY-CODE,
     DETAIL-CODE.
```

II.2 ENTRY CODES

Entry Code	Condition
1	A transmission from the host computer contained an extra escape character (ESC,X'27') as part of a Write or Erase/Write data stream.
2	A transmission from the host computer is intended for a terminal that is conditioned to interrupt all transmission in this terminal, and execute SB3270 as a result of the intercept.
3	An operator entered a key that is defined (During system configuration) as a request for SB3270 execution.
4	An operator entered a key that would normally cause a transmission to the host, at a terminal that is conditioned to intercept all transmissions and to cause execution of SB3270.
5	A terminal that SB3270 previously requested control of is now available.
6	An idle condition exists where SB3270 could be executed, but none of the other conditions are pending.
7	A print operation initiated by SB3270 has terminated; the simulator returns the termination status of the operation to the user program in Detail-Code.

DETAIL CODES

Entry Codes 1 and 2, and SBCOPY:

```
Bit                              Explanation

... ... ... ... xxx x.. ... ...    Specifies operation:
                000 1              1 - Write
                001 0              2 - Read Buffer
                001 1              3 - Erase Unprotected
                010 0              4 - Read Status
                010 1              5 - Erase Write
                011 0              6 - Read Modified
                011 1              7 - Copy
                100 0              8 - System reset  (LOCAL ONLY)
... ... ... ... ... ... xx. ...    Defines the printout format:
                        00         0 - Unformatted
                        01         1 - 40 Column Print Format
                        10         2 - 64 Column Print Format
                        11         3 - 80 Column Print Format
... ... ... ... ... ... ..1 ...    Start printout
... ... ... ... ... ... ... 1..    Sound audible alarm
... ... ... ... ... ... ... .1.    Restore keyboard operation
... ... ... ... ... ... ... ..1    Reset all modified data tags (MDT).
... ... ... ... ... ... ... .xx    For copy command only:
                             00    0 - Attributes only
                             01    1 - Attributes and unprotected only
                             10    2 - Attributes and protected only
                             11    3 - Entire screen
```

Notes:

1. The subroutine "SBBITS" may be used to determine the value of specified bit combinations.

2. When SB3270 is entered, the operations indicated by the DETAIL-CODE will already have been performed except:
   a. A print operation is not started, even if bit 20 is set.
   b. The keyboard remains locked, even if bit 22 is set.

Entry Code 4:

Detail Code Value	Key Pressed or Equated Key Sequence	Aid Transmitted When Passed to SBXMIT
1	ENTER	X'7D'
2	CLEAR	X'6D'
3	TEST REQ	X'F0'
4	PA1	X'6C'
5	PA2	X'6E'
6	PA3	X'6B'
7	PF1	X'F1'
8	PF2	X'F2'
9	PF3	X'F3'
10	PF4	X'F4'
11	PF5	X'F5'
12	PF6	X'F6'
13	PF7	X'F7'
14	PF8	X'F8'
15	PF9	X'F9'
16	PF10	X'7A'
17	PF11	X'7B'
18	PF12	X'7C'

NOTES:

1. The above are the Attention Identification Codes (AID) associated with the specified keys; however, the same eighteen detail codes may be specified for any other eighteen key combinations during Simulator Generation.

2. If the DETAIL-CODE is changed by the subroutine, the transmitted AID character will correspond to the modified DETAIL-CODE.

3. If a 3270 access key is redefined as a subroutine access key, entry code 3 is used when the key is pressed. The detail code will be whatever was assigned (see the NTP 100/150 System Generation and Debugging Guide.

Entry Code 7:

Code	Meaning
1	Intervention Required
2	Normal device end
3	Device end and intervention required
4	Other (device busy, unit specify, transmission check, command reject, data check, control check, or operation check).

II.4 NTP/150 SUBROUTINES

Terminal Control Request Routines:

SBXMIT	Requests a screen to be transmitted to host.
SBREQ	Requests control over a specified terminal.
SBINTO	Turns off the Subroutine Intercept Flag for the currently controlled terminal.
SBINT1	Turns on the Subroutine Intercept Flag for the currently controlled terminal.
SBULCK	Unlocks keyboard on currently controlled terminal.
SBFLSH	Flushes keystroke queue on currently controlled terminal.
SBDISC	Disconnects a terminal from communications lines.
SBBUSY	Sets a terminal "busy" to the host.
SBRLSE	Releases a terminal set busy by SBBUSY.

System Function Routines:

SBCOPY	Allows a copy from one terminal to another.
SBBEEP	Sounds alarm on currently controlled terminal.

Screen Erase Routines:

SBEEOF	Erase to end of field.
SBEINP	Erase all unprotected fields.
SBEPRT	Erase all protected fields.
SBEALL	Erase the full screen.

Cursor Movement Routines:

SBSOF	Put cursor at attribute of current field.
SBTAB	Tab cursor to next entry field.
SBBTAB	Back-tab cursor to previous entry field.
SBTABA	Put cursor at next field - protected or unprotected.
SBCURA	Move cursor forwards or backwards a specified number of character positions.
SBHOME	Put cursor at home position.
SBNL	Move cursor to the first input field on the next line.

Bit, Character, and Data Manipulation Routines:

SBBITS	Extract bits from 3270 commands, controls, attribute bytes, etc. for examination.
SB6BIT	Change 3270 commands, controls, attribute bytes, etc. from six-bit structure to COBOL computational code.
SBPACK	Strip nulls from a field.
SBMOVE	Move full screens or large data blocks to and from program buffers, etc.

Current Attribute Subroutines:

SBMDTO	Reset Modified Data Tag
SBMDT1	Set Modified Data Tag
SBPRTO	Reset Protected Tag
SBPRT1	Set Protected Tag
SBFLDO	Reset Field End Processing Tag (Reserved bit in IBM)
SBFLD1	Set Field End Processing Tag (Reserved bit in IBM)
SBDARK	Set Invisible Intensity
SBNORM	Set Normal Intensity
SBHIGH	Set Bright Intensity

II.5 SBRSET - Perform Simulator Reset
(3270 local Systems - only)

In 3270 local systems, the IBM channel can issue a command to the 3271 Control Unit to perform a system reset. The local NTP/150 interface has been designed to intercept the command and notify SB3270 that a "system reset" command was issued, but the simulator has not performed it. ENTRY-CODE 1 has been modified to include the "system reset" command in its DETAIL-CODE as follows:

Bits	12	13	14	15			
	0	0	0	1	=	1	Write Command
	0	0	1	0	=	2	Read Buffer Command
	0	0	1	1	=	3	Erase All Unprotected Command
	0	1	0	0	=	4	Read Status Command (Remote)
	0	1	0	1	=	5	Erase Write Command
	0	1	1	0	=	6	Read Modified Command
	0	1	1	1	=	7	Copy Command (Remote Only)
	1	0	0	0	=	8	System Reset Command (Local Only)

Even though there was no "ESC" order in the data stream, the "system reset" command will cause entry to SB3270 at ENTRY CODE 1.

At the receipt of a "system reset" command, an IBM 3271 control unit will

- Disconnect any terminal connected to the IBM communications channel.
- Clear all terminal buffers to nulls.
- Reset and unlock keyboards.
- Set all terminal AID's to keyboard or printer null.
- Set all cursors to the "home" position.

SBRSET is the NTP/150 subroutine that performs all the functions of a system reset command.

When SB3270 is called with ENTRY-CODE = 1 and
with DETAIL-CODE indicating system reset, the user may ignore the command and continue processing or use SBRSET to reset the simulator.

Calling sequence:

```
    Call "SBRSET" for COBOL applications.
          or
         BRM  SBRSET    for ASM applications
         DCN  0
```

II.6 SOFTWARE ACTION CODES

	000	020	040	060	100	120	140	160	200	220	240	260	300	320	340
000			SP	0	@	P	'	P	NOP	DUP	PFC	C12			
001		[	!	1	A	W	a	q	UP	FM	ESC	C13			
002			"	2	B	R	b	r	LFT	PA1	HOM	C14			
003	¢		#	3	C	S	c	s	RGT	PA2	LCK	C15			
004			$	4	D	T	d	t	DWN	PA3	tb:	C16			
005			%	5	E	U	e	u	ENT	PF1	C01	C17			
006				6	F	V	u	v	RST	PF2	C02	C18			
007		]	'	7	G	W	g	w	CLR	PF3	C03	2NL			
010			(	8	H	X	h	x	TAB	PF4	C04				
011			)	9	I	Y	i	v	BTB	PF5	C05				
012			*	:	J	Z	j	z	NL	PF6	C06				
013			+	;	K		k		INS	PF7	C07				
014			'	<	L		l	¦	DEL	PF8	C08				
015			-	=	M	¦	m		EIN	PF9	C09				
016	¬		.	>	N		n		EEF	PFA	C10				
017			/	?	O		o		TRO	PFB	C11				

To find a SOFTWARE ACTION CODE, find the keyboard graphic or acronym. Add the associated value in the verticle column to the left, to the value in the horizontal row at the top. For example, to find the code for COBOL access key C07, add the vertical colimn value, 013, to the horizontal row value, 240. The sum, 253, is the SOFTWARE ACTION CODE for COBOL ACCESS CODE 07.

LEGEND OF ACRONYMS

NOP - NO OPERATION
UP - CURSOR UP
LFT _ CURSOR LEFT
RGT - CURSOR RIGHT
DWN - CURSOR DOWN
ENT - ENTER. E01
RST - RESET, ATTENTION
CLR - CLEAR
TAB - TAB OR SKIP
BTB - BACK TAB
NL - NEW LINE, CURSOR RETURN
INS - INSERT
DEL - DELETE
2NL - 2260 NEW LINE RPQ

EIN - ERASE INPUT
EEF - ERASE EOF
TRQ - TEST REQUEST
DUP - DUPLICATE
FM - FIELD MARK
PA1-PA3 - PROGRAM ACCESS KEYS 1 THRU 2.
PF1-PFC - PROGRAM FUNCTION KEYS 1 THRU 12.
ESC - ESCAPE
HOM - HOME
LCK - LOCK
tb: - TAB-TO-COLON OPTION
C01-C18 - COBOL ACCESS 01 THRU 18.

SECTION III DEBUGGING FACILITIES

III.1 ESCAPE KEY FUNCTIONS

Key	Function	Password	Basic	Debug	S&F	Printer
1	Toggle CG	FOURFAZE	X			
B	Exit to DOS	FOURFAZE		X		
C	Take checkpoint	FOURFAZE		X		
D	Dynamic Dump	FOURFAZE		X		
I*	Pass string to 3270	FOURFAZE		X		
J*	Turn on input line trace	FOURFAZE		X		
K*	Turn on output line trace	FOURFAZE		X		
L	Set number of lines/page	none				X
M	Set address of Dynamic Dump Dump	FOURFAZE		X		
O*	Output string to 360/370	FOURFAZE		X		
P	Store password	none	X			
Q*	Disable 360/370 output if = 0	FOURFAZE		X		
T	Execute keystrokes & time level 7	FOURFAZE		X		
U	Stop timer and display it	none		X		
V	Store value string in address	FOURFAZE		X		
W	Set printer spacing	none				X
X*	Toggle time out	FOURFAZE		X		
Z**	Print screen	FOURFAZE				X
ESC	Store attribute character	FOURFAZE		X		

* - Remote only.
\+ - For details about S&F, see the 3270 Operator's Manual
** - Must be an unbuffered character printer

III.2 SCREEN DISPLAY OF ATTRIBUTE CHARACTERS

Octal	Display	Key	Field Attribute				
300	@	(=)	UNPROT	ALPHA	DISPLAY	NO-@FLD	NO-MDT
301	A	(A)	UNPROT	ALPHA	DISPLAY	NO-@FLD	MDT
302	B	(B)	UNPROT	ALPHA	DISPLAY	@FIELD	NO-MDT
303	C	(C)	UNPROT	ALPHA	DISPLAY	@FIELD	MDT
304	D	(D)	UNPROT	ALPHA	DISPLAY	NO-@FLD	NO-MDT
305	E	(E)	UNPROT	ALPHA	DISPLAY	NO-@FLD	MDT
306	F	(F)	UNPROT	ALPHA	DISPLAY	@FIELD	NO-MDT
307	G	(G)	UNPROT	ALPHA	DISPLAY	@FIELD	MDT
310	H	(H)	UNPROT	ALPHA	BRIGHT	NO-@FLD	NO-MDT
311	I	(I)	UNPROT	ALPHA	BRIGHT	NO-@FLD	MDT
312	J	(J)	UNPROT	ALPHA	BRIGHT	@FIELD	NO-MDT
313	K	(K)	UNPROT	ALPHA	BRIGHT	@FIELD	MDT
314	L	(L)	UNPROT	ALPHA	DARK	NO-@FLD	NO-MDT
315	M	(M)	UNPROT	ALPHA	DARK	NO-@FLD	MDT
316	N	(N)	UNPROT	ALPHA	DARK	@FIELD	NO-MDT
317	O	(O)	UNPROT	ALPHA	DARK	@FIELD	MDT
320	P	(P)	UNPROT	NUM	DISPLAY	NO-@FLD	NO-MDT
321	Q	(Q)	UNPROT	NUM	DISPLAY	NO-@FLD	MDT
322	R	(R)	UNPROT	NUM	DISPLAY	@FIELD	NO-MDT
323	S	(S)	UNPROT	NUM	DISPLAY	@FIELD	MDT
324	T	(T)	UNPROT	NUM	DISPLAY	NO-@FLD	NO-MDT
325	U	(U)	UNPROT	NUM	DISPLAY	NO-@FLD	MDT
326	V	(V)	UNPROT	NUM	DISPLAY	@FIELD	NO-MDT
327	W	(W)	UNPROT	NUM	DISPLAY	@FIELD	MDT
330	X	(X)	UNPROT	NUM	BRIGHT	NO-@FLD	NO-MDT
331	Y	(Y)	UNPROT	NUM	BRIGHT	NO-@FLD	MDT
332	Z	(Z)	UNPROT	NUM	BRIGHT	@FIELD	NO-MDT
333	divide	divide	UNPROT	NUM	BRIGHT	@FIELD	MDT
324	cross	cross	UNPROT	NUM	DARK	NO-@FLD	NO-MDT
335	bar	(-)	UNPROT	NUM	DARK	NO-@FLD	MDT
336	arrow	(divide)	UNPROT	NUM	DARK	@FIELD	NO-MDT
337	score	(cross)	UNPROT	NUM	DARK	@FIELD	MDT
340	`	(0)	UNPROT	NUM	DARK	NO-@FLD	NO-MDT
341	a	a	PROT	ALPHA	DISPLAY	NO-@FLD	MDT
342	b	b	PROT	ALPHA	DISPLAY	@FIELD	NO-MDT
343	c	c	PROT	ALPHA	DISPLAY	@FIELD	MDT
344	d	d	PROT	ALPHA	DISPLAY	NO-@FLD	NO-MDT
345	e	e	PROT	ALPHA	DISPLAY	NO-@FLD	MDT
346	f	f	PROT	ALPHA	DISPLAY	@FIELD	NO-MDT
347	g	g	PROT	ALPHA	DISPLAY	@FIELD	MDT
350	h	h	PROT	ALPHA	BRIGHT	NO-@FLD	NO-MDT
351	i	i	PROT	ALPHA	BRIGHT	NO-@FLD	MDT
352	j	j	PROT	ALPHA	BRIGHT	@FIELD	NO-MDT
353	k	k	PROT	ALPHA	BRIGHT	@FIELD	MDT
354	l	l	PROT	ALPHA	DARK	NO-@FLD	NO-MDT
355	m	m	PROT	ALPHA	DARK	NO-@FLD	MDT
356	n	n	PROT	ALPHA	DARK	@FIELD	NO-MDT
357	o	o	PROT	ALPHA	DARK	@FIELD	MDT

(Screen display of attribute characters cont.)

Octal	Display	Key	Field Attribute				
360	p	p	PROT	NUM	DISPLAY	NO-@FLD	NO-MDT
361	q	q	PROT	NUM	DISPLAY	NO-@FLD	MDT
362	r	r	PROT	NUM	DISPLAY	@FIELD	NO-MDT
363	s	s	PROT	NUM	DISPLAY	@FIELD	MDT
364	t	t	PROT	NUM	DISPLAY	NO-@FLD	NO-MDT
365	u	u	PROT	NUM	DISPLAY	NO-@FLD	MDT
366	v	v	PROT	NUM	DISPLAY	@FIELD	NO-MDT
367	w	w	PROT	NUM	DISPLAY	@FIELD	MDT
370	x	x	PROT	NUM	BRIGHT	NO-@FLD	NO-MDT
371	y	y	PROT	NUM	BRIGHT	NO-@FLD	MDT
372	z	z	PROT	NUM	BRIGHT	@FIELD	NO-MDT
373	left-brace	"divide"	PROT	NUM	BRIGHT	@FIELD	MDT
374	bar	"cross"	PROT	NUM	DARK	NO-@FLD	NO-MDT
375	right-brace	"#"	PROT	NUM	DARK	NO-@FLD	MDT
376	tilde	"up-arrow"	PROT	NUM	DARK	@FIELD	NO-MDT
377	%	"back-arrow"	PROT	NUM	DARK	@FIELD	MDT

( ) = Character is entered with shift depressed
" " = Character is entered with ctrl depressed

III.3 DISPLAY of LINE CONTROL CHARACTERS

	Mnemonic		Mnemonic		Mnemoni
.	Unknown	\	ITB	←	PT
^	SYN	|	DLE	|	SBA
≠	ENQ	|△	ACK0	\	EAU
▲	EOT	|/	ACK1	—	IC
b	STX	|@	RVI	✓	RA
△	SOH	|,	WACK	■	SUB
¢	ETX	ʌ ʌ "" ≠	General poll	○	ESC
]	ETB	ʌ ʌ ≠	Select dvc 0	|	SF
-	NAK	ʌ ʌ A A ≠	Select dvc 1		

III.4 TRC327

TRC327 is used to display and/or print formatted output of the debug log from a checkpoint file.

```
// TRC327
/INPUT = filename @ drive
/OUTPUT = filename @ drive
//
```

If no input file name is given, the default file name of CKPT@0 is used if running on 8230 or DUMP@0 is used if running on an 8250.

TRC327 uses terminal 0 for all displays and keyboard commands.

KEY	FUNCTION
1	The display is moved up one line.
H	The display is moved up four lines
F	The display is moved up 24 lines
E	The program will exit to IDOS or DKOS.
R	Restart the display
C	The 3270 line trace counters are displayed
S	Start or stop a full system trace display
P	Print the trace on the SYSOUT printer. While printing, depressing the P key will terminate printing and cause the program to wait for another control key.

Error Counters:

```
COUNTER   VALUE    COUNTER   VALUE    COUNTER   VALUE    COUNTER   VALUE
  !?!    aaaaaaaa    PTO    bbbbbbbb    DBL    cccccccc    LST    dddddddd
  N16    eeeeeeee    T-O    ffffffff    BCC    gggggggg    RTY    hhhhhhhh
  UE     iiiiiiii    US     jjjjjjjj    OC     kkkkkkkk    CR     llllllll
  DMY    mmmmmmmm    VER    nnnnnnnn
```

aaaaaaaa	Invalid interrupt.
bbbbbbbb	Pad timeout.
cccccccc	8121 printer error or IR condition.
dddddddd	Lost data due to too few buffers.
eeeeeeee	Error detected in 8436 controller.
ffffffff	Time-outs waiting for host response.
gggggggg	BCC errors.
hhhhhhhh	Retries by host. Opposite of BCC.
iiiiiiii	Unit exception status generated by local simulator.
jjjjjjjj	Unit specify sense generated.
kkkkkkkk	Operation check sense generated.
llllllll	Command reject sense generated.
mmmmmmmm	
nnnnnnnn	Version identification of simulator.

Trace Log Entries:

```
              8436-2 STATUS
STT ssssssss dddddddddddddddddddd
    ssssssss = 8436-2 status word in octal.
    dd ... d = Description of status bits that are on.

              8436-2 CONTROL
CNT cccccccc dddddddddddddddddddd
    cccccccc = 8436-2 control word in octal.
    dd ... d = Description of control bits that are on.

              8436-2 DATA IN OPERATION
HOST xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx dddddddddddddddddd
    xx = 1 byte of data received from the host system in hex.
    dd .. d = Description of BSC termination character if any.

              8436-2 DATA OUT OPERATION
TERM xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx dddddddddddddddddd
    Format the same as HOST above.

              7073 STATUS
STT ssssssss dddddddddddddddddddd
    ssssssss = The 7073 status word in octal.
    dddddddd = Description of any status bits that are on.
```

(TRC327 log entries cont.)

```
                 7073 CONTROL
CNT cccccccc dddddddddddddddddddd
    cccccccc = The 7073 control word in octal.
    dddddddd = description of any control bits that are on.

                 7074 STATUS    (WIDGET)
WST ssssssss dddddddddddddddddddd
    ssssssss = The 7074 status word in octal.
    dddddddd = Description of any status bits that are on.

                 7074 CONTROL
WCT cccccccc dddddddddddddddddddd
    cccccccc = 7074 control word in octal.
    dddddddd = Description of any control bits that are on.

ADR aaaaaaaa
    aaaaaaaa = The address associated with the next trace entry.  For
               7073 status and control entries it is the return address
               of the status or control subroutine that will create the
               next trace log entry.

                 8121 PRINTER CONTROL
CUT cccccccc
    cccccccc = 8121 control command in octal.

                 8121 DATA OUT
DAT dddddddd
    dddddddd = Data word sent to the 8121 in octal.

                 NTP150 CALLS
CBL eeeeeeee ddddddd sssssss mm:ss.tt
    eeeeeeee = Entry-code in octal.
    dddddddd = Detail-code in octal.
    ssssssss = User-subscript in octal.
    mm:ss.tt = Time in minutes, seconds, and tenths.

                 TIME
TIM mm:ss.tt                          *TIMEOUT*
    mm:ss.tt = Time as above since the simulator was started.
    *TIMEOUT* = Remote only,  the previous TIM entry was more than 3
                minutes previous.

                 TASK CONTROL
TSK aaaaaaaa mm:ss.tt
    aaaaaaaa = Address of the calling routine.
    mm:ss.tt = Time as above of call.

                 GENERAL PURPOSE MARK
MRK xxxxxxxx
    xxxxxxxx = Various value used for debugging purposes.  Varies from
               release to release.
```

III.6 TAKING MEMORY and CHECKPOINT DUMPS

Generate the 3270 Simulator with the debug feature in order to take memory and checkpoint dumps.

Memory Dump to Line Printer:

Place AUTO/MANUAL to MANUAL
Hit SYSTEM RESET and STEP
Load TIR with 46700000
Place AUTO/MANUAL to AUTO

Checkpoint Dump to 8230 or 8250:

In in debug mode, ESC C will take a checkpoint dump; OR

Place AUTO/MANUAL to MANUAL
Hit SYSTEM RESET and STEP
Load TIR with 71100001
Place AUTO/MANUAL to AUTO

III.7 DMP327 EXECUTION

```
// DMP327
```

/INPUT = name@i	Name of checkpoint file
/TOP = LINETRACE	Memory dump will terminate at the beginning of the line trace area, if there is one. Otherwise, it will terminate at the top of memory.
/TOP = MEMORY	The memory dump will include the line trace area.
/TOP = xxxxxx	The memory dump will terminate at address xxxxxx minus one.
/LOWER CASE	Lower case letters may be printed because the printer has upper and lower case.
/UPPER CASE	Convert lower case to upper before printing (Initial default).
/NO LOWER CASE	Do not print lower case, substitute blanks for all lower case letters.
/FORMAT = OCTAL	Memory dump in OCTAL and ASCII.
= HEX	Memory dump in hexadecimal and ASCII.
= BOTH	Memory dump in octal, hexadecimal, and ASCII.

SINGLE WORDS IN RAM (REMOTE)

CU	A six bit index to the table of control unit addresses.CU should be in the range 0-31, which results in a control unit address in the range 40-5F.
MEOT	The master EOT flag. If negative EOT will be sent in response to a general poll. If positive a text response will be sent.
CX1	The index to the user table for the screen currently active. If zero, no screens are active.

SINGLE WORDS IN RAM (LOCAL)

CU	The control unit address. Binary 'xxxx0000'.
INITWD	The initialization control word used by the address recognition logic in the channel adapter.
INIT	The AID code in the leftmost byte, and the 2 byte cursor position in the rightmost bytes. Used in the last READ MODIFIED or READ BUFFER operation.
CNTLWD	The last control word issued to the channel adapter (7073) by a control type I/O instruction.
STATUS	Contains the last status word from the channel adapter.
CMBUSY	This flag is set non-zero when a new command interrupt occurs to indicate that interrupt level 1 is busy and is reset to zero when ending status is posted to the channel.
NEWCMD	The command code portion of the status, right justified and zero-filled from the last new command interrupt.
DEV#	The address of the last device for which New Command type status was received.
CHAIN	A flag; when negative the current command is chained from the previous command. (Bit 0 of CHAIN is bit 12 of the last New Command type status word.)
RESDBC	The number of bytes transferred during the last operation, computed from the residual byte count of the last Continue/ End status.
DISARM	A flag; when DISARM is non-zero, level-1 is should notbe rearmed.
CX1	The index to the user table of the terminal currently connected to the host for communications.

(DMP327 output cont.)

REGISTERS

The registers at the time the checkpoint was taken.

TASK CONTROL BLOCKS

Task Control Blocks are printed vertically.

LINE	LABEL	DESCRIPTION
1	TCBLNK	Address of next TCB.
2	TCBPRI	Dispatching priority.

TCB Name	Priority	Use
CMTCB	0	communications task.
KBTCB	100	keyboard task.
CWTCB	100	communications wait task.
TOTCB	200	remote communications time-out task. cations
AIDTCB	500	attention ID (Local) task.
IRTCB	500	printer intervention required task.
UTCB	100/150	one task for every terminal, line printer and character printer

3	TCBX1	Address of the associated user table.
4	TCBX2	Address of the instruction to be executed when the task becomes active.
5	TCBCTR	The address of the task's event counter. If negative the task is not ready to execute. The currently executing task is not in the Task Control Block Queue, so it is not displayed.
6		The value of the task's event counter.

COMMUNICATIONS BUFFERS

The queue of pending communications buffers is often empty. The first buffer in the free queue is the last buffer processed. The free queue operates as a stack, hence the second buffer in the queue is probably not the next-to-last buffer processed. There may be buffers at the end of the free queue which have never been used. The free queue and the pending queue should not both be empty.

The first word of each comm buffer points to the next comm buffer. Zero indicates the end of the queue. For local, the remaining 255 words each contain 3 bytes of data. For remote, the remaining 16 words of the buffer each contain only 1 byte of data, so only the data byte of each word is printed. The data in the buffer includes commands, orders, attributes, true data, and for remote the bisync data link control characters.

COMMUNICATIONS SCREEN

The communications screen area is used to construct video screens. When data is written from the host the destination screen is copied into the communications screen area. The commands, orders, and data from the host are interpreted onto the comm screen resulting in a combination of original screen data and the new data written by the host. The comm screen is then copied to the destination screen.

USER TABLES

The terminal address has different formats for remote and local. For local there is a unique two-digit hex device number assigned to each terminal. The numbers for the terminals are sequential, and the first one must end in zero. (This is also the control unit address.) No gaps are allowed. For remote, two two-digit hex numbers are associated with each terminal. The first is the control unit address, and should be the same for the first 16 terminals. (If there are more than 16 terminals, the control unit address for the second 16 should be one more than the first one.) The second number is the device address.

Note that FWA is the first word of the user table, while K3RET1 and K3RET2 precede the user table.

K3RED1	Object address from keyboard IOID table
K3RET2	Contains a BRM K3INT to process keyboard interrupts
FWA	Address of first word of user screen
LWA	Address of last word of screen + 1
PT023	The return address after waits by the printer code
LPBEND	Lineprinter buffer ending address + 1 while printing. If 0 then printing is not in progress
CLINK	Link in cursor blink list (a timer list)
CURC	Clock for cursor blink. If 0 then cursor is turned off and the keyboard is disabled
PSROT	Cursor address byte pointer (SROT, SR1T, SR2T)
DEST	Word address of cursor
ID	(Double Word) Format name for Store & Forward. 8121 code uses first word for a return address
FSROT	Byte address of attribute governing current cursor position. (SROT, SR1T, SR2T)
FDEST	Word address of attribute
EMARK	Event counter for printer termination task and EM flag for printers
USERNO	Index of this user in polling list
CURA	True data under cursor (visible for .4 sec)
CURB	Current cursor (visible for .1 sec)

(DMP327 output cont.)

LNSPPG	Lines per page for 8121s and 8131s (DMP327 user table cont.)
FCUR	Attribute governing current field (address in FSROT & FDEST)
COLID	Current column number of cursor
KDATA	BRM to keyboard data routine, dependent on current keyboard operation mode
FDSAVE	Second to last character sent to 8121
KPW	The password in hashed form
AID	The current attention ID character. If the sign bit is set then the AID character has not been processed
KVIDI	Address of first video intensity byte
KLAST	Last raw keystroke processed
KINDEX	Keybord index, minus shows keyboard alarm
QUECNT	Count of keystrokes in queue for user
COBOL4	COBOL scratch
VATTR	Set for formatted screens (screens with 1 or more attributes; 0 otherwise)
INDSCSL	Flag for system selected indicator
INDCSA	Flag for system available indicator
INDCIM	Flag for insert mode indicator
INDCII	Flag for input inhibited indicator
PRBA14	Line length for current print operation. A function of CMDCC in the user table
PRGBF	Non-zero if current field of screen is printable; zero if it's invisible
PRCONV	An indexed load instruction from the internal-to-printer code conversion table (LDA TOINPR,X3); to allow upper and lower case
PRBA12	Printer byte counter (line length)
PRFL19	Count of the number of non-null characters in current print line (0 indicates all nulls). The sign bit is set to indicate printing in progress
PQCNT	Printer queue blocks available for this printer ??
CMDCC	Command control character (e.g. CCC or WCC from 370)
ERRBSS	Status/Sense bytes for the terminal
KTYPE	Contains an indexed load from the keystroke to internal code conversion table specified at configuration (LDA T1NAIN,X3)
LDBG20	Counter of ESC keys hit before the first non-ESC.
PDDAT	Printer driver scratch (New line indicator for 8121s)
PDRET	Printer driver scratch (Return address saved here)
PDLF	Printer driver scratch (Event counter for line printer task. Active line counter for 8121s; initialized with LNSPPG of the user table)

IV.1 3270 CONTROL CHARACTER SUMMARY

Hex	Name	Type	Description
01	SOH	LINE CTL	Start of header.
02	STX	LINE CTL	Start of text.
03	ETX	LINE CTL	End of text.
05	PT	BUFFER ORDER	Program tab.
0C	FF	PRINT CTL	Form feed.
10	DLE	LINE CTL	Data link escape. See below.
11	SBA	BUFFER ORDER	Set buffer address.
12	EUA	BUFFER ORDER	Erase unprotected to address.
13	IC	INSERT CURSOR	Insert cursor.
15	NL	PRINT CTL	New line.
19	EM	PRINT CTL	End of media.
1D	SF	BUFFER ORDER	Start field.
26	ETB	LINE CTL	End of transmission block.
27	ESC	BUFFER ORDER	Escape.
2D	ENQ	LINE CTL	Enquiry, bid for line.
32	SYN	LINE CTL	Synchronization character.
37	EOT	LINE CTL	End of transmission.
3C	RA	BUFFER ORDER	Repeat to address.
55	LPAD	LINE CTL	Leading pad.
6F	EAU	CMD	Erase all unprotected.
F1	WRITE	CMD	Write.
F2	RDBUF	CMD	Read buffer.
F5	E/WRT	CMD	Erase write.
F6	RDMOD	CMD	Read modified.
F7	COPY	CMD	Copy.
FF	TPAD	LINE CTL	Trailing pad.
		TWO CHARACTER SEQUENCES	
10 37	DISC	LINE CTL	Disconnect. (switched line)
10 61	ACK1	LINE CTL	Acknowledgement of odd blocks.
10 6B	WACK	LINE CTL	Acknowledgement with wait.
10 70	ACK0	LINE CTL	Acknowledgement of even blocks.
10 7C	RVI	LINE CTL	Acknowledgement with reverse interrupt.
02 2D	TTD	LINE CTL	Temporary text delay.

PAD	PAD brackets the beginning and end of each transmission.
SYN	Two or mor SYN characters must precede a STX to allow the receiving station to establish synchronization. SYN characters may be embedded within a tranmission to help the receiver maintain synchronization. They are discarded by the receiver.
ACK0 ACK1	ACK0 and ACK1 are positive acknowledgements to even and odd text blocks respectively.
NAK	NAK is transmitted by the simulator in response to a text transmission that contains an ENQ or lacks a terminating ETX or ETB or has an incorrect BCC. The simulator responds to a NAK by retransmitting the last text block.
ENQ	The simulator transmits an ENQ to request a retransmission of the last non-text message usually after 3 second timeout. ENQ is transmitted to the simulator as the last character of a polling or selection sequence. When ENQ is imbedded in a text message the simulator responds with NAK.
WACK	WACK is transmitted by the simulator in place of ACK0 or ACK1 in response to an selection sequence or command if the selected device (printer) is busy.
RVI	RVI is transmitted by the simulator in response to selection when a status/sense message is pending. When the simulator receives RVI it responds with EOT and resets all pending status/sense information.
STX	STX marks the beginning of text messages both transmitted and received and starts the accumulation of the BCC. STX is imbedded within both status/sense and test request messages.
SOH	SOH marks the beginning of both status/sense and test request messages sent by the simulator.
ETB	The simulator terminates intermediate transmission blocks with ETB rather than ETX. ETB's received are treated as ETX's.
ETX	ETX mark the end of a sequence or one or more text transmission blocks for both send and receive. It terminates BCC accumulation and indicates that the next 2 characters are the BCC.
EOT	EOT is transmitted in response to an ACK when the simulator has sent its last message. It is transmitted in response to all RVI's. When EOT is received it terminates selection and resets the simulator to control mode.
TTD	TTD is sent by the host to delay transmission from the simulator. The simulator always responds with NAK.

IV.3 3270 DATA FORMATS

Polling and Selection;

```
CUADDR-CUADDR-DEVADDR-DEVADDR-ENQ      (Specific poll)
  1      1      1       1      1

CUADDR-CUADDR-"7F7F"-ENQ               (General poll)

CUADDR'-CUADDR'-DEVADDR-DEVADDR-ENQ     (Selection seq)
  1       1       1       1       1
                                       CUADDR' = CUADDR + X"20"
```

Input Message:

```
SYN-STX-CUADDR-DEVADDR-AID-(cursor addr)-(data)-ETB/ETX-BCC
 n   1    1       1     1          2        n       1     2
```

Output Message:

```
SYN-STX-ESC-(command)-WCC-(data)-ETX-BCC
 n   1   1      1      1    n     1    2
```

Status/Sense Message:

```
SYN-SOH-"%R"-STX-CUADDR-DEVADDR-(status)-(sense)-ETX
 n   1   2    1     1      1       1        1      1
```

IV.4 LOCAL and REMOTE COMMAND CODES

Command	Remote EBCDIC	Remote ASCII	Local EBCDIC	
Write	F1	31	01	(HEX)
Erase/Write	F5	35	05	
Read Buffer	F2	32	02	
Read Modified	F6	36	06	
Copy	F7	37	N/A	
Select	N/A	N/A	0B	
Erase All Unprotected	6F	3F	0F	
No Operation	N/A	N/A	03	
Sense	N/A	N/A	04	

IV.5 WRITE CONTROL CHARACTER (WCC)

x1..	Determined by contents of bits 2 - 7.
..xx	Define the printout format:
00	The NL order in data stream determines line length.
01	40 character print line
10	64 character print line
11	80 character print line
.... 1...	initiates a printout operation at completion of the write operation.
.... .1..	Sounds the audible alarm at the end of the operation.
.... ..1.	restores operation of the keyboard, resets the INPUT INHIBITED indicator and the AID byte.
....1	All MDT bits in the device's buffer are reset before any data is written or orders are executed.

COPY CONTROL CHARACTER (CCC)

x1..	Bits 0 and 1 are determined by the contents of bits 2 - 7.
..xx	Define the printout format
00	The NL order in data stream determines line length
01	40 character print line
10	64 character print line
11	80 character print line
.... 1...	Initiates a printout operation at the device after buffer transfer are completed.
.... .1..	Sounds the audible alarm at the "to" device after buffer transfers are completed.
.... ..xx	Define the type of data to be copied
00	Copy only attribute characters
01	Copy only attribute characters and unprotected fields (plus nulls). Nulls are transferred for the alphanumeric characters not copied from the protected fields.
10	All attribute characters and protected fields (plus nulls) are copied. Nulls replace the alphanumeric characters in the unprotected fields.
11	The entire storage buffer is copied.

IV.6 BUFFER CONTROL ORDERS

Order	Byte 1		Byte 2	Byte 3	Byte 4
	EBCDIC	ASCII			
Start Field (SF)	1D	1D	Attribute character	n/a	n/a
Set Buffer Address (SBA)	11	11	Address byte 1	Address byte 2	n/a
Insert Cursor (IC)	13	13	n/a	n/a	n/a
Program Tab (PT)	05	09	n/a	n/a	n/a
Repeat To Address (RA)	3C	14	Address byte 1	Address byte 2	Character to be repeated
Erase Unprotected To Address (EUA)	12	12	Address byte 1	Address byte 2	n/a

IV.7 ATTRIBUTE BYTE

x1..	Bits 0 and 1 are determined by the contents of bits 2 - 7.
..1.	Field protected.
...1	Field numeric.
.... xx..	Display intensity:
00	Normal.
01	Normal (Light pen detectable for IBM).
10	Bright.
11	Dark.
.... ..1.	NTP/150 @FIELD. (Not used by IBM)
....1	Modified data tag (MDT).

IV.8 ATTENTION IDENTIFICATION BYTE

EBCDIC		ASCII		
HEX	OCTAL	HEX	OCTAL	MEANING
60	030	2D	055	Keyboard null.
E8	350	59	161	Printer null.
7D	175	27	047	Enter key.
F1	361	31	061	PF1 key.
F2	362	32	062	PF2 key.
F3	363	33	063	PF3 key.
F4	364	34	064	PF4 key.
F5	365	35	065	PF5 key.
F6	366	36	066	PF6 key.
F7	367	37	067	PF7 key.
F8	370	38	070	PF8 key
F9	371	39	071	PF9 key.
7A	172	3A	072	PF10 key
7B	173	23	043	PF11 key.
7C	174	40	100	PK12 key.

The following AID's result in short reads after general polls

E6	346	57	127	ID card reader (not used by Four Phase).
7E	176	3D	075	Light pen (not used by Four Phase).
6C	154	25	045	PA1 key.
6E	156	3E	076	PA2 key.
6B	153	2C	054	PA3 key.
6D	155	5F	137	Clear key.
F0	360	30	060	Test request key.

IV.9 REMOTE STATUS and SENSE BYTES

Bits	Byte 0:
x1..	Dettermined by contents of bits 2 - 7.
..xx	Reserved.
.... 1...	Device Busy (DB. The addressed device is executing an operation or a previous command or Specific poll detected a busy condition. The device is busy when executing an EAU command, a print operation, or certain keyboard operations (Erase Input, Backtab, and Clear). Set with Operation Check when a Copy command is received that specifies a busy "from" device. Set with Unit Specify when a command is addressed to a busy device.
.... .1..	Unit Specify (US). Set with DB if a command is addressed to a busy device.
.... ..1.	Device End (DE). The addressed device has become available ready, or not busy. This bit is not considered pending status by a Selection Addressing sequence.

If a Selection Addressing sequence detects that the addressed device has pending status and also detects one of the above status changes that warrants a Device End, then the Device End bit is set and preserved along with the other pending status, and an RVI response is made.

....0	Not used.

(Remote status/sense bytes cont.)

Bits	Byte 1:
x1..	Determined by the value of bits 2 - 7.
..1.	Command Reject (CR). Receipt of an invalid 3270 command (or Copy command if this feature is not installed).
...1	Intervention Required (IR). Set if: A Copy command contains a "from" address that specifies an unavailable device. A command attempted to start a printer but found it not ready. The printout is suppressed. A Selection Addressing sequence or a Specific Poll sequence is received for a device that is unavailable or that became not ready during a printout. A General Poll sequence does not respond to the unavailable/not ready indication and proceeds to determine the state of the next device. A command is received for a device that has been logged as unavailable or not ready.
.... 1...	Equipment Check (EC). Indicates that a printer character generator error occurred or the printer became mechanically disabled.
.... .1...	Data Check (DC). Indicates the detection of a partiy or Cursor check.
.... ..1.	Control Check (CC). a device failed to respond to communications or failed to complete an operation within a specified time period.
....1	Operation Check (OC). Receipt of an illegal buffer address or of an incomplete order sequence on a Write or Erase/Write command. The device did not receive a CCC or a "from" address on a Copy command. Receipt of an invalid command sequence. (ESC is not received in the second data character position of the sequence.)

IV.10 LOCAL STATUS and SENSE BYTES

Status byte:

Bit	Meaning
1...	ATTENTION (ATTN). An AID generating key has been entered.
.1..	STATUS MODIFIER (SM). Set with BUSY if the 7073 can not accept a command from the channel.
..1.	CONTROL UNIT END (CUE). The control unit is no longer busy.
...1	BUSY (see SM above).
.... 1...	CHANNEL END (CE). The control unit has terminated the data transfer portion of an I/O command.
.... .1..	DEVICE END (DE). The control unit and device have completed a cmd.
.... ..1.	UNIT CHECK (UC). An error has occurred that is further defined in the sense byte.
....1	Unit Exception (UE). Non-standard but valid completion of a command. Not used by Four Phase.

Sense Byte:

Bit	Meaning
1...	Command Reject (CR). The channel has received an invalid command.
.1..	Intervention Required (IR). The addressed printer is not ready (out of paper, off line, etc.).
..1.	Bus Out Check (BOC) Parity error on the channel's bus-out lines.
...1	Equiptment Check (EC) A parity error has been detected on data transferred to the channel. Set by the System IV/70 load-address-register control word (bit 8).
.... x...	Not Used
.... .1..	Unit Specify (US) The sense bits are the result of a System IV/70 processing unit-detected error.
.... ..1.	Not Used
....1	Operation Check (OC) The Channel Adapter has received a valid command that cannot be executed. Same as in the remote simulator.

V.1 8436 CONTROL AND STATUS

8436-2 Status Word

... ... .1.	Ring indicator
... .1	Transmitted data has been lost.
... ... 1..	In transmit mode output data is needed.
... ... .1.	The data set is ready.
... ...1	Clear to send.
... ... 1..	Sync character received.
... ...1	Receiving carrier.
... ... 1..	Input ready.
... ... .c. .cc ..c	Interrupt when bit changes.
... ... t.. ... t..	Interrupt when bit becomes true.

8436-2 Control Word

... ... .1. ...	Set data terminal ready.
... ...1 ...	Reset data terminal ready.
... ... 1..	Reset transmitter.
... ... .1.	Request to send.
... ...1	Reset receiver.

7073 IOID Addresses:

ADDR	TYPE OF INTERRUPT
054	New command
055	Continue operation
056	End of operation
057	Data transfer required (input or output)

Initialization Control:

```
010 0.. ... ... ... ... ... ...   Identifies initialization control.
... ... ... ..x ... ... ... ...   Screen size:   0 = 480,  1 = 1920
... ... ... ... xxx xxx xx. ...   Upper control unit address range.
... ... ... ... ... ... ..x xxx   Lower control unit address range (xxxx0000).
```

Load Byte Control Word One:

```
000 0.. ... ... 0.. ... ... ...   Identifies load byte control word one.
... ... xx. ... ... ... ... ...   Wrapped modified field control:
        00                        xx = 00 - no change
        01                           = 01 - no change
        10                           = 10 - reset wrapped modified field
        11                           = 11 - set wrapped modified field
... ... ..x x.. ... ... ... ...   Modified data tag control:
          0 0                     xx = 00 - no change
          0 1                        = 01 - no change
          1 0                        = 10 - reset modified data tag
          1 1                        = 11 - set modified data tag
... ... ... .xx ... ... ... ...   Starting byte position:
             00                   xx = 00 - left byte
             01                      = 01 - middle byte
             10                      = 10 - right byte
             11                      = 11 - invalid
... ... ... ... .xx xxx xxx xxx   Byte count for data transfer.  When
                                  exhausted causes a continue interrupt.
```

Load Byte Control Word Two:

```
000 0.. ... ... 1.. ... ... ...   Identifies load byte control word two.
... ... ... ... .xx xxx xxx xxx   Relative screen address of data.  Used to
                                  compute SBA's on read modified.
```

Load Bit Control Word:

```
000 1.. ... ... ... ... ... ...   Identifies load bit control word.
... .1. ... ... ... ... ... ...   Reset chained command latch.
... ..1 ... ... ... ... ... ...   Disable EBCDIC to ASCII conversion.
... ... 1.. ... ... ... ... ...   Enables next 3 bits for printer control.
... ... .1. ... ... ... ... ...   Generates an asynchonous DEVICE END inter-
                                  rupt from the addressed printer.
... ... ..1 ... ... ... ... ...   Sets busy device status for the addressed
                                  printer.
... ... ... 1.. ... ... ... ...   Sets intervention required status for the
                                  addressed printer.
... ... ... ... ... .xx xxx xxx   Device address.
... ... ... ... ... ... .pp ppp   Printer address.
```

(7073 status and control cont.)

Load Address Register Control:

```
001 0.. ... ... ... ... ... ...   Identifies load address register control
... ..1 ... ... ... ... ... ...   Set operation check sense bit.
... ... .1. ... ... ... ... ...   Set unit specify sense bit.
... ... ..1 ... ... ... ... ...   Set equiptment check sense bit.
... ... ... 1.. ... ... ... ...   Set intervention required sense bit.
... ... ... .1. ... ... ... ...   Set attention status bit.
... ... ... ..1 ... ... ... ...   Set channel end status bit.
... ... ... ... 1.. ... ... ...   Set device end status bit.
... ... ... ... .1. ... ... ...   Set unit exception status bit.
... ... ... ... ... 1.. ... ...   Asynchronous control.
... ... ... ... ... .xx xxx xxx   Device address.
```

New Command Status:

```
1.. ... ... ... ... ... ... ...   Data required.
.1. ... ... ... ... ... ... ...   Attention or device end accepted.
... 1.. ... ... ... ... ... ...   System reset has been signalled by channel.
... .1. ... ... ... ... ... ...   Printer control word accepted.
... ..1 ... ... ... ... ... ...   Channel end has been sent to the channel.
... ... 00. ... ... ... ... ...   Identifies new command status word.
... ... ... ... 1.. ... ... ...   Current cmd is chained from previous cmd.
... ... ... ... .xx x.. ... ...   Command code:
                 00 0             000 = INVALID
                 00 1             001 = WRITE
                 01 0             010 = ERASE/WRITE
                 01 1             011 = ERASE ALL UNPROTECTED
                 10 0             100 = INVALID
                 10 1             101 = READ BUFFER
                 11 0             110 = SELECT
                 11 1             111 = READ MODIFIED
... ... ... ... ... .xx xxx xxx   Device address
```

Continue/End Status:

```
... .1. ... ... ... ... ... ...   Printer control word accepted.
... ..1 ... ... ... ... ... ...   Channel end sent to channel.
... ... xx. ... ... ... ... ...   Type of interrupt:
        00                        00 = NEW COMMAND  (see above)
        01                        01 = CONTINUE OPERATION
        10                        10 = OPERATION ENDED BY CHANNEL
        11                        11 = OPERATION ENDED BY 7073
... ... ..1 ... ... ... ... ...   Attribute detected during read.
... ... ... 1.. ... ... ... ...   Modified data tag detected during read.
... ... ... .xx ... ... ... ...   Ending byte position:
             00                   00 = Last word full.
             01                   01 = last word has 1 byte left.
             10                   10 = Invalid
             11                   11 = Last word has 2 bytes left.
... ... ... ... .xx xxx xxx xxx   Residual byte count.
```

7074-2 Control Word:

```
1.. ... ... ... ... .xx xxx xxx   Literal load  xxxxxxxx = value to load
.1. ... ... ... ... ... ... ...   BCC mode flag.
..1 ... ... ... ... ... ... ...   Protected field flag.
... xxx x.. ... ... ... ... ...   Ending character codes:
    010 0                         End on unprotected attribute.
    100 0                         End on attribute with MDT.
    110 0                         End on any attribute.
    001 0                         End on any null.
    000 1                         End on any control character.
... ... .x. ... ... ... ... ...   Set MDT status bit.
... ... ..1 ... ... ... ... ...   Load literal for protected non-attributes.
... ... ... 1.. ... ... ... ...   Load literal for unprotected non attributes.
... ... ... .1. ... ... ... ...   Load literal for all characters.
... ... ... ..x x.. ... ... ...   Processing flags:
              0 1                 Reset MDT on all attributes.
              1 0                 Do not load control characters.
              1 1                 Do not load attributes or nulls
... ... ... ... .1. ... ... ...   Leave data right adjusted in shift register.
... ... ... ... ..1 ... ... ...   Beginning alignment override.
... ... ... ... ... xx. ... ...   Beginning alignment:
                    00
                    01
                    10
                    11
... ... ... ... ... ..x xxx xxx   Byte length count for operation.
```

7074-2 Status Word:

```
.1. ... 1.. ... ... ... ... ...   Attribute found.
..1 ... ... ... ... ... ... ...   Protected flag.
... ... .1. ... ... ... ... ...   MDT flag.
... ... ... ... ... xx. ... ...   Ending alignment:
                    00            Left byte.
                    01            Middle byte.
                    10            Right byte.
                    11            Invalid.
... ... ... ... ... ..x xxx xxx   Byte count residue.
```

VI.1 BYPASSING COMMUNICATIONS HALT

1. Switch 0 up.
2. Key in simulator name.
3. Clear first halt.
4. When second halt occurs, BRA to RP+1 into TIR (720xxxxx).
5. Hit any PA/PF ENTER key to hang on a cut word. Re-enter the BRA to RP+1 as above in order to continue.

VI.2 ADJUSTMENT of DUAL INTENSITY

The following procedure can be used to adjust the internal video pots that control 3270 dual intensity.

1. Execute a DEBUG version of the 3270 simulator.

2. Depress CLEAR and RESET until the screen is blank.

3. Type ESC P FOURFAZE. Capital letters must be entered, if needed use the SHIFT key depending upon the type of keyboard and SYSGEN options. Only a cursor should appear on the screen. If the P appears, the I01 jumper for unique numeric island codes is not installed. If FOURFAZE appears, a capital P was not entered after the ESC key.

4. Type ESC A. Four P's should appear on the right of the video in place of the 3270 the indicator lights. If not, step 3 was not correct. During the following steps, the P's will change to other characters.

5. Using the capital W and REPT key, place W's across the first three rows of the video.

6. Press the HOME key. Type ESC ESC H. "H" will appear and move the cursor one column to the right .

7. Press CURSOR RETURN. Type ESC ESC E. "E" will appear and move the cursor to the right.

8. Press CURSOR RETURN. Type ESC ESC L. "L" will appear and move the cursor to the right.

9. Type ESC A. The four indicators will disappear. The top row of W's should be at high intensity, the second row should be at normal intensity and the third row should be invisible.

   "W" facilitates adjustment of all three video pots. The attributes "H" "E" "L" are easy to remember.

VI.3 MARKETING CONSIDERATIONS FOR 3270 LOCAL

1) The 3270 local or remote should not be sold as an operator's console.

2) Installation of our 3270 local on the same block multiplexor channel as IBM 3330 disc drive is likely to cause serious degradation problems.

3) When the Tab-to-Colon option is desired, specify it on the SOF. The C7 and C9 chips in the Channel Adaptor must be modified.

MASTER DIRECTORY SECTOR- Release AH04

+0	0		addr of the config sector	
+1	0		0	
+2	0		02435	
+3	MAINT code word = 0			
+4	"FVS" (ASCII)			
+5	first sector address of the format value set (FVS) directory			*1
+6	"JOB" (ASCII)			
+7	First sector address of the job directory			*2
+010	"ALC" (ASCII)			
+011	sector address of the allocation sector			*3
+012	binary zeroes			
+013	release level in ASCII			
+014	0	# sectors-1	1st sector addr of overlay 1	*4
. . .	0	# sectors-1	1st sector addr of overlay n	*4
+063	0	# sectors-1	1st sector addr of overlay 40	*4
+064 . . .	reserved			
+070	C		# sectors in DATA47 area	*5
+071 . . . +0377	0	Allocation table of 199 words to represent cylinders 1 to 199 on an 8230 disc. The first 31 words of this table are not used as DATA47 starts at sector 01000 (cylinder 32). This table contains zeroes while DATA is running and the bits are kept in the allocation sector. Word +0131 represents the first 020 sectors of the DATA47 area. Word +0132 has the second 020 sectors, word +0133 the third 020 sectors, ... Bits 8-23 represent the relative sectors in the cylinder (bit 8 indicates relative sector 0 and bit 23 indicates relative sector 017).		

Footnote #1: 03441 on many packs

Footnote #2: 03442 on many packs

Footnote #3: 02440 on many packs

Footnote #4: bit 0-8 reserved
9-11 number of sectors -1
11-23 first sector address of the overlay

Footnote #5: bit 0 (C) is the CRASH flag which indicates whether or not DATA was exited properly: 0=yes, normal exit and allocation table is correct; 1=no, abnormal exit and allocation table is zeroed.

STANDARD 4-WORD SECTOR HEADER

Word +0	FORWARD POINTER	NNNNNNNNNNDDSSSSSSSSSSSS
	NNNNNNNNNN	number of words used in the data sector; for other than data sectors, all zeroes.
	DDSSSSSSSSSSSS	physical drive number and sector address ofthe next sector in this chain; if zeroes, then this is the last sector in the chain.
Word +1	BACKWARD POINTER	0000000000DDSSSSSSSSSSSS
	DDSSSSSSSSSSSS	physical drive number and sector address of the previous sector in this chain; if all zeroes, then this is the first sector in the chain.
Word +2	HERE POINTER	0000000000DDSSSSSSSSSSSS
	DDSSSSSSSSSSSS	physical drive number and sector address of this sector
Word +3	CODEWORD	CCCC*DISP*DDSSSSSSSSSSSS

If this sector is a format/value set sector, this word contains the name of that format or value set. Otherwise,

CCCC 4-bit code indicating the type of sector

0000	master directory
0100	job directory
0010	batch directory
0101	maintenance sector
0011	format/value set directory (leftmost four bits of the format/value set number)
1000	data sector

DISP The six most significant bits of the displacement (inwords) into the job directory sector (for batches) or batch directory sector (for data) of the entry to which this chain belongs. These bits are zero for all but batch directory and data sectors.

DDSSSSSSSSSSSS The physical drive number and sector address of the entry described under *DISP*. These bits are zero for all but batch directory and data sectors.

The code word is zero for the master directory and negative only for data sectors.

CONFIG SECTOR- 2nd SECTOR IN MASTER DIRECTORY CHAIN

+0 . .	standard 4-word header		
+4	time of last change (t-86400) in seconds		
+5	day of last change		
+6 . . .	32 words for the first word address (FWA) of each screen		
+046	32 words for the first word address (FWA) of each user table		
+0106	screen size (0140/0300/0600/01400)		
+0107	master password		
+0110	debug password (Mode D)		
+0111	reserved password		
+0112	supervisor password		
+0113	size of alternate tape buffer (in words)		
+0114	first word address (FWA) of alternate tape buffer (0=none)		
+0115	release level in ASCII		
+0116	configuration word (SYSFIG)		*1
+0117	bits 0-17 are unused	number of terminals	
+0120	LPOUT		
+0121	FWA of the 256-word disc read-back buffer (0=none)		
+0122	blocking factor for $DIRECT		
+0123	configuration word (SYSFG1)		*2
+0124 . . .	not used		
+0154 . .	22 words of the first word address (FWA) of system blocks		
+0202 . . .	29 words of the FWA of between-screen areas		
+0237	configuration option word (COPT)		*3
+0240 . +0377	not used		

CONFIGURATION SECTION NOTES

Footnote *1: Configuration Word (SYSFIG)

bit	description
0	video: 0=48, 1=81
1	ram size: 0=24K, 1=48K
2	7008 system: 0=no, 1=yes
3	ram-to-disc dump capability: 0=no, 1=yes
4	keyboard: 0=source, 1=keypunch (029)
5	printer: 0=no, 1=yes
6	magnetic tape: 0=no, 1=yes
7	tape density: 0=9-track (800 bpi), 7-track(556 bpi) 1=9-track (1600 bpi), 7-track (800 bpi)
8	tape parity: 0=even, 1=odd
9	disc read-back: 0=no, 1=yes
10	audible alarm: 0=no, 1=yes
11	separate paging buffer: 0=no, 1=yes
12	bisync: 0=no, 1=yes
13	12-channel printer carriage tape: 0=no, 1=yes
14	bisync extended retry: 0=no, 1=yes
15	bisync memory log: 0=no, 1=yes
16	omit left zero key in key verify mode: 0=no, 1=yes
17	not used
1820	number of tape decks
2123	number of disc drives

Footnote *2: Configuration Word (SYSFG1)

bit	description
0	find mode operator statistics: 0=no, 1=yes
1	detail operator statistics by batch: 0=no, 1=yes
2	save format source with object code: 0=no, 1=yes
3	bisync printer: 0=SYSOUT, 1=bisync direct printer
4-23	not used

Footnote *3: Configuration Word (COPT) - 0=yes, 1=no

bit	description
0	multiply
1	check digit
2	modulo 7 check digit
3	modulo 10 check digit
4	modulo 11 check digit
5	hexadecimal
6	generate check digit
7	DYNAMO
8	overpunch
9	keypunch (029) style keyboard
10	right justify
11	printer
12	ram dump
13	reserved for internal use
14	right justify in verify mode
15-23	not used

JOB DIRECTORY SECTOR

+0 . .	standard 4-word header		
+4 . .	job name (9 ASCII characters left justified with blank fill)		*1
+7	see footnote #2	batch directory disc address	*2
+010	reserved	record size (bytes)	#3
+011	password (0=none)		
+012 .	format number (PROG1)	format number (PROG2)	*4
. .	format number (PROG3)	format number (PROG4)	
. .	format number (PROG5)	format number (PROG6)	
+015 .	value set number 1 (V1)	value set number 2 (V2)	*4
. .	value set number 3 (V3)	value set number 4 (V4)	
. .	value set number 5 (V5)	value set number 6 (V6)	
. .	value set number 7 (V7)	value set number 8 (V8)	
+021	value set number 9 (V9)	0	
+022 . .	job name (9 ASCII characters left justified with blank fill)		
+025			

Note: A job directory entry is a fixed length of 14 words and a sector contains a maximum of 18 entries.

Footnote #1: If the first word of the jobname entry is zero, then this is an empty slot which is available for describing the next job created.

Footnote #2:
bit 0 SELECT
1 reserved
2 BUSY
3-9 reserved
10-23 batch directory disc address

Footnote #3:
bit 0-13 reserved
14-23 record size in characters (bytes)

Footnote #4: These are packed decimal entries (3 digits of 4 bits each) to indicate the format/value set number in 12 bits. Unused entries are zeroes. A maximum of 6 formats and 9 value sets may be assigned to a particular job.

BATCH DIRECTORY SECTOR

+0 . .	standard 4-word header			
+4	batch number right justified			*1
+5	see footnote #2	drive=0	first sector address of data	*2
+6	0	drive	last sector address of data	
+7	L	operator id of last entry operator		*3
+010	L	operator id of last operator in key verify		*3
+011	see footnote #4			*4
+012	day when this batch was last accessed (3 ASCII digits)			
+013	transaction code (3 ASCII characters)			
+014 . .	accumulator 1 4 words to represent 12 ASCII digits			
+020 . .	accumulator 2 4 words to represent 12 ASCII digits			
+024 . .	accumulator 3 4 words to represent 12 ASCII digits			
+030 . .	accumulator 4 4 words to represent 12 ASCII digits			
+034 . .	accumulator 5 4 words to represent 12 ASCII digits			
+040	batch number right justified			

BATCH DIRECTORY SECTOR Continued

te: A batch directory entry is a fixed length of 28 words and a sector contains a maximum of 9 entries. When a batch is purged, the contents of relative word 1 is saved in relative word 010 before word 1 is cleared.

Footnote #1: The six digits of the batch number are in packed decimal format with 4 bits used for each digit. The batch number is right justified within the word (000000-999999).

Footnote #2:	bit 0-1	reserved	
	2	COMP	complete: 0=no, 1=yes
	3-4	00=FIND 01=ENTRY 10=VERIFY 11=VERIFY COMPLETE	
	5	WWR	was written: 0=no, 1=yes
	6	WRD	was read: 0=no, 1=yes
	7	WRF	was reformatted: 0=no, 1=yes
	8	CBRF	created by reformat: 0=no, 1=yes
	9	DIS	display request: 0=no, 1=yes
	10-11		drive=0
	12-23		first sector address of data chain
ootnote #3:	bit 0	L	last operation: 0=no, 1=yes
	123		operator identification
Footnote #4:	bit 0	RPUR	request to purge: 0=yes, 1=no
	1	RWR	request to write: 0=yes, 1=no
	2	RRD	request to read: 0=yes, 1=no
	3	RRF	request to rerformat: 0=yes, 1=no
	4	LOCK	batch locked from access: 0=yes, 1=no
	5	INPR	batch active (in-process): 0=yes, 1=no
	6-23		time since midnight (t-86400 seconds)

Any of the above flags being set to zero (yes) will "lockout" the batch.

DATA SECTOR

+0	see footnote #1	drive	forward pointer	*1
+1	0	drive	backward pointer	
+2	0	drive	here pointer	
+3	see footnote #2	drive	sector address of batch dir.	*2
+4	time batch was created (t-86400 seconds)			
+5	record header			*3
+6	ASCII characters of the record			
.	3 characters per record			
.	any trailing 1 or 2 bytes are not defined			
.				
.				
.	record header			
.	another data record			

Note: All records in all batches of any one job are the same length which is determined by job size. The last sector of the chain must always have room for one additional record. The last word of the last sector in the batch contains the binary value of BINREC (record count).

A data sector has a 5-word header for the sector instead of the standard 4-word header.

The combination of the record header and the record data is a minimum of two words and a maximum of 251 words.

Footnote #1:

bit 0-7	0=full; 5=empty; otherwise, number of words in use
8-9	always 0
10-11	drive number of the next data sector
12-23	sector address of the next sector of data

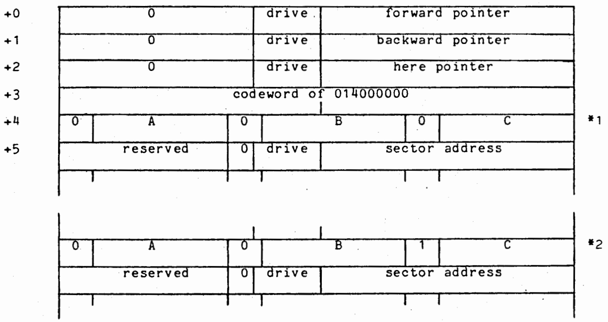
Footnote #2:

bit 0	always 1
1-9	displacement of the batch directory entry in the sector designated in bits 12-23 divided by 4
10-11	drive number of the batch directory sector
12-23	sector address of the batch directory entry

Footnote #3: A copy of the record header is kept in RECHDR,X1 of the user table while the record is displayed.

bit 0	always 1
1	reserved
2	field type override (PROG CTRL off)
3	validation override (VALID key hit)
4	unintelligible (? key hit)
5-6	reserved
7	has been key verified
8	correction (record altered since entry)
9-20	reserved
21-23	program level -1 (0-5)

FORMAT/VALUE SET DIRECTORY



Note: Each entry is a two-word entry consisting of the format or value set name and the drive/sector address where the format or value set object code is located. ABC indicates the 3-digit ASCII identifier for the format or value set. (A, B, and C must be greater than or equal to 060 and less than or equal to 071.) All zeroes in word 1 of the entry indicates that the slot is empty and available for the next format or value set that is created. A maximum of 126 entries are contained within one sector.

Footnote #1: Format directory entry

Footnote #2: Value set directory entry

FORMAT CODE SECTOR

+0	0	drive	forward pointer	
+1	0	drive	backward pointer	
+2	0	drive	here pointer	
+3	format name (3 ASCII digits)			
+4	1	first block of format object code		*1
.				
.				
.				
+042				
+043	0	first block of format source code		*2
.	0	0	0	
.				
.				
+0373				
+0374	implied screen size in bytes			*3
+0375	implied record size ($SIZE) in bytes			*3
+0376	day number in ASCII when the format was compiled			*3
+0377	time (t-86400 seconds) when the sector was written			*3

Note: The sector address in the format/value set directory points to the first sector of this chain. This chain will exist on multiple drives with the first sector always on drive 0.

Footnote #1: Format object code always begins in the first sector of the format code sectors. Object code is stored on the disc in blocks of 31 words so that calling sequences do not span system blocks when the format is paged into memory. Object code blocks are indicated by bit 0 of the first word of each object code block being set to 1. If a calling sequence will span two blocks, the block is filled with NOP instructions (006700000) to pad the block to 31 words and the calling sequence is put into the next block. A total of 8 blocks are contained on a sector.

Footnote #2: Format source code follows all the blocks of format object code. Source code is indicated by bit 0 of all words of source code being set to 0.

Footnote #3: These values are kept in each sector.

VALUE SET SECTOR

+0		drive	forward pointer	
+1		drive	backward pointer	
+2		drive	here pointer	
+3	value set name (3 ASCII digits)			*1
+4			NW	*2
+5	string of ASCII characters			
+6				
+7				
.				
.				
.				
.				
NW+4	zero word signaling the end of the value set items			

Footnote #1: Bit 16 of the value set name is set to 1.

Footnote #2: Each item in the value set begins on a word boundary. NW is the number of words/item (1 less than or equal to N less than or equal to 250).

First Sector of Overlay

+0	overlay number in octal
+1	BRA entry 1
+2	BRA entry 2
. . . . . .	
+n	BRA entry n
n+1	balance of first sector overlay code

Second thru Fourth Overlay Sectors

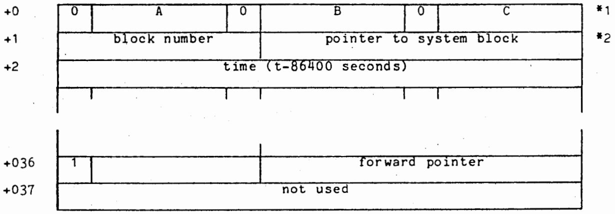
+0 . . . . . +0377	second thru fourth sectors of overlay code

Note: Overlays are 1-4 sectors in length (0400-02000 words) and load at absolute location 07400. All of the sectors for an overlay are within the same cylinder on the disc.

When entered, register X1 points to the User Table. Subroutines GETSYS and PUTSYS must be used to read and write (respectively) sectors on the disc. ALLOCS is called to allocate a new sector. Q8RET is called when SYSBUF is not being used to allow the preallocator to catch up. SYSBUF immediately follows the overlay area at absolute location 011400.

DIRECTORY OF IN-RAM FORMAT PAGES

PAGDIR points to the first system block of the page directory.



Note: FORPTR,X1 points to the last word in the entry. When a keystroke is processed or LINK to a new block, the current system time is store in the last word. When no job is assigned to a terminal, FORPTR,X1 =0.

Footnote #1: ABC represents the 3 digit ASCII name of the format. If thisword contains zeroes, then there is no entry.

Footnote #2: Block # contains the page number of the format (for example, 0, 1, 2 ... <512).

DISC SPACE ALLOCATION

Calculation of sectors available for data on a one drive system:

Total sectors available = 06200 or 3200 (decimal)		3200	
Required sectors for DOS		- 512	
DATA47 total sectors available on drive 0		2688	
Sectors required for overlays		- 124	
Sectors required for the master directory and config sector		- 2	
Sectors available for other directories, formats, value sets, and data		2562	
Job directory sectors (minimum 1)	________		*1
Batch directory sectors (minimum 1/job)	________		*2
Format/value set directory sectors (minimum 1)	________		*3
Total sectors for directories		- ________	
Total sectors available for object code and data		________	
Format object code sectors (minimum 1/format)	________		*4
Value set object code sectors (minimum 1/value set)	________		
Total sectors for object code		- ________	
Total sectors available for data records		-________	*5

Notes:

1. Job directory entries are a fixed length of 14 words and each sector has a maximum of 18 entries.
2. Batch directory entries are a fixed length of 24 words and each sector has a maximum of 9 entries.
3. Format/value seet directory entries are a fixed length of 2 words and each sector has a maximum of 126 entries.
4. The number of sectors for format code varies according to the complexity of the individual format. Each sector will contain a maximum of 8 system blocks of format object code.
5. The number of data records that may be contained in a sector varies based on the number of characters per record. The following table may be used to determine the number of characters/number of records per sector ratio:

373-750 = 1
247-372 = 2
184-246 = 3
147-183 = 4
121-146 = 5
103-120 = 6

91-102 = 7
79-90 = 8
73-78 = 9
64-72 = 10
58-63 = 11
56-57 = 12

49-55 = 13
46-48 = 14
43-45 = 15
40-42 = 16
37-39 = 17
34-36 = 18

31-33 = 20
28-30 = 22
25-27 = 25
22-24 = 27
19-21 = 31
16-18 = 32

13-15 = 41
10-12 = 5
7-9 = 62
4-6 = 83
1-3 = 125

0	interrupt locations	
060	screen areas, user tables, system blocks read-back disc buffer, tape buffer	
07000	PUSER (normal keystroke interrupt IOID table)	
07040	PUSERL (hardware lost keystroke interrupt IOID table)	
07100	PIOID5 (level 5 IOID table for 8121 printers)	
07120	KSWRAP - level 3 code (S56+3)	
07260	LEVEL4- level 4 code (S56+4)	
07400	4-sector overlay area	
011400	SYSBUF (system sector buffer)	
012000	M1 (S56+5 code)	
013610	LEVEL7 (S56+7 code)	*
015176	LINK (S56+9 code)	*
015426	GETSYS (S56+8 code)	*
016202	$DISCI (S56+2 code)	*
016642	AOLMPY (S56+C code)	*
016662	CONFIG options; system blocks	*
020000		

* This location may change depending upon the system configuration.

IN-RAM JOB DIRECTORY

JOBRAM PZE address of the first system block of the directory

+0	0	pointers to format entries in the Active Format/Value Set Directory
+6	0	pointers to value set entries in the Active Format/Value Set Directory
+017	0	pointers to format entries in the Active Format/Value Set Directory
+025	0	pointers to value set entries in the Active Format/Value Set Directory
+035	1	pointer to next system block
+036		not used
+040		

Note: The total entry for a job is 15 words in length and 1 or 2 entries are contained in one system block. If the first word of the entry is zero, then no entry exists. Any other words set to zero indicates that the format or value set does not exist. Each job must have at least one program level assigned.

IN-RAM FORMAT/VALUE SET DIRECTORY

VSRAM PZE address of the first system block of the directory

<table>
<tr><td>+0</td><td>0</td><td colspan="2">A</td><td colspan="2">0</td><td colspan="2">B</td><td>T</td><td>C</td><td>*1</td></tr>
<tr><td>+1</td><td colspan="2">1st block #/sector</td><td colspan="2">0</td><td colspan="2">drive</td><td colspan="3">sector address</td><td>*2</td></tr>
<tr><td>+2</td><td></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td></td><td></td><td></td></tr>
<tr><td>.
.
.
.</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="3"></td><td></td></tr>
<tr><td>+036</td><td>1</td><td colspan="8">pointer to next system block</td><td></td></tr>
<tr><td>+037</td><td colspan="9">not used</td><td></td></tr>
</table>

Note: One or more of these entries are needed to represent an entire format or value set. If a format spans n (n<65) sectors, then n entries are needed. Each entry represents up to 8 blocks (i.e., one sector).

The MODE key deletes unneeded entries before creating new entries.

The maximum number of blocks per sector is 8 (a systems constant) and is used to build these entries and set the value in the second word of the entry (1st block number) for this sector for additional entries after the first.

All formats being actively used by terminals will have entries in thisdirectory. Entries that are part of multiple sector formats will appear in ascending order by "1st block number" (0, 8, 16, etc.).

Footnote #1: ABC indicates the 3-digit ASCII name of the format/value set. T indicates the type of entry: 0=format, 1=value set. The words in the JOBRAM entry point to these entries.

Footnote #2:

bit 0-8	1st block number for sector
9	always 0
10-11	drive number
12-23	sector address

USER TABLE

Note: Each user table is 96 (0140) words in length and resides below location 07100 in memory. Each word in PUSER (the level 3 keyboard IOID table) contains a BRM to a user table.

Skeleton Definitions

```
     +0     PZE    0         keystroke handling routine
     +1     IO     IOPAIR    take in a keystroke
     +2     SKN*   IOPAIR+1  skips if buffer is full
     +3     BRD    $3        else, debreak level 3
     +4     BRM    KSWRAP    branch to handle wrap-around
0003 KSEXIT EQU    3         postion of BRD in table
0004 KSBRM  EQU    4         position of BRM KSWRAP in table
0005 CURC   EQU    5         counts level 4 until cursor change needed
0006 CUR    EQU    6         off-screen copy of blinking word
0007 SCR    EQU    7         execute to place character on screen
0010 PSROT  EQU    8         PZE SROT(1st word of word pair PSROT-DEST)
0011 DEST   EQU    9         location on screen(2nd word of PSROT word pair)
0012 MODE   EQU    10        operation type (see *1)
0013 OPPTR  EQU    11        pointer to operator directory
0014 JOBDES EQU    12        pointer to job descriptor table
0015 FORPTR EQU    13        pointer to current format
0016 FSROT  EQU    14        pointer to start of current field
0017 FDEST  EQU    15        first word of current field
0020 REQ8   EQU    16        request for level 8
0021 RECLOC EQU    17        location of current record in sector
0022 WIDTH  EQU    18        negative width of field
0023 MODPTR EQU    19        pointer to in-front modifier
0014 QUE8RA EQU    20        return address from BRM QUE8
0025 ACCUM  EQU    21        accumulator 0 (4 words)
0031 ACCUM1 EQU    ACCUM+4   accumulator 1 (4 words)
0035        EQU    ACCUM+8   accumulator 2 (4 words)
0041        EQU    ACCUM+12  accumulator 3 (4 words)
0045        EQU    ACCUM+16  accumulator 4 (4 words)
0051        EQU    ACCUM+20  accumulator 5 (4 words)
0055 STATS  EQU    45        operator statistics table
0056 STATS1 EQU    46        second word of operator statistics table
0057 STATS2 EQU    47        third word of operator statistics table
0060 BINCOL EQU    48        binary column counter
0061 LASSEC EQU    49        address of last sector in batch
0062 SAVPTR EQU    50        save pointer for backspacing
0063 MODEXT EQU    51        mode extension flag word (see *2)
0064 BATCH  EQU    52        batch number
0065 VRECB  EQU    53        pointer to verify record buffer (see *3)
0066 RECHDR EQU    54        current record header
0067 RECSIZ EQU    55        record size
0070 SECTOR EQU    56        current sector number
0071 NEXTCR EQU    57        return word from BRM NEXTC
0072 COLUMN EQU    58        FWA status line; LWA+1 message line
0073 FLASH  EQU    59        MVE length level 4 flasher; bit 0=toss ks
0074 FX2    EQU    60        X2=location in format code (via BAL)
0075 FX3    EQU    61        X3=negative number of columns left in field
0076 BINREC EQU    62        current binary record number
0077 MAXREC EQU    63        maximum record number for this batch
            FORCE  0
0100 IOPAIR EQU    64        DCN 01401 (cut word of level 3 data-in)
0101 KSIN   EQU    65        level 3 current buffer pointer for next ks
            FORCE  0
0102 KSCUR  EQU    66        level 7 current buffer pointer for next ks
0103 KSFWA  EQU    67        pointer to first word of keystroke buffer
0104 KSBUF  EQU    68        beginning of keystroke buffer
0024 KSLEN  EQU    20        length of keystroke queue in words
0126 KSLOST EQU    KSBUF+KSLEN-2 place in ks queue to put lost ks message
            FORCE  0
0130 ALARM  EQU    KSBUF+KSLEN   flags end of ks buffer; audible alarm word
0131 MODE2  EQU    ALARM+1   cut word for I/O control for beep (see *4)
0132 STATS3 EQU    90        fourth word of operator statistics table
0133 OV36BL EQU    91        address of system block for MODE T
            FORCE  0
0134 WPA    EQU    92        scratch for overlay and display use only
0135 WPA1   EQU    93        second word of double word pair
0136 FWA    EQU    94        address of first word of screen area
0137 LWA    EQU    95        addr of last word +1 of screen area (msg line)
```

USER TABLE FOOTNOTES

Footnote #1: MODE

bit 0	1=normal mode from keyboard, 0=feed-thru mode from level 8
1	program control: 1=off, 0=on
2	release key in process: 1=yes, 0=no
3	verify active: 1=yes, 0=no
4	waiting for minus key in left zero field: 1=yes, 0=no
5	waiting at end of field (must release/left zero): 1=yes, 0=no
6	pass 1 (disc-to-screen): 1=yes, 0=no
7	Automatic Skip/Duplicate (ASD): 1=on, 0= off
8	special mode: 1=yes, 0=no
9	verify mode: 1=yes, 0=no
10	find mode: 1=yes, 0=no
11	skip/release inhibit (must fill): 1=yes, 0=no
12	character input inhibited: 1=yes, 0=no
13	record: 1=new, 0=existing
14	Auxiliary Duplicate (AUX DUP): 1=inhibited, 0=allowed
15	index: 1=inhibited, 0=allowed
16	pseudo new record: 1=yes, 0=no
17	job assigned: 1=no, 0=yes
18	must enter: 1=yes, 0=no
19	perform validation checks: 1=yes, 0=no
20	right justify before modifier encountered: 1=yes, 0=no
21	waiting for zero after MULT key followed by minus: 1=yes, 0=no
22	MULT key active: 1=yes, 0=no
23	left zero field: 1=yes, 0=no

Footnote #2: MODEXT

bit 0	record down in key verify (bits 1-23=0)
1	mode key (bit 0=1)
2	record up (bit 0=1)
3	record delete
4	record insert (bit 0=1); verify reconstruct (bit 7=1)
5	HOME key and PROG keys
6	pass 2 for record insert
7	verify reconstruct
8	waiting for key-in after MODE key in key verify mode
9	waiting for SKIP after end-of-tape encountered
10-17	reserved
18	screen print in process
19-23	current line number from top-of-screen (00-23)

Footnote #3: Verify record buffer (VRECB)

bit 0	record insert: 1=yes, 0=no
1	field correction: 1=yes, 0=no
2	character correction: 1=yes, 0=no
3-8	reserved
9-23	first word address of the verify record buffer

Bit 0 is turned off in overlay 1 entry 9.

If VRECB is all zeroes, then this is not key verify mode.

Footnote #4: MODE2

bit 0	cursor state flip-flop (used by level 4)	
112	reserved	
13-15	=03 (channel 3)	bits 13-23 are the cut word for alarm beeping
1621	terminal number in binary	
22-23	=02 (control I/O)	

DUMPING RAM TO DISC OR TAPE

1. AUTO to MANUAL
2. RESET
3. STEP
4. LOAD 71100001 into TIR
5. MANUAL to AUTO
   The contents of memory are copied to sectors 06200-06257.
6. a. Use DTUX to copy the sectors to tape

```
// DTUX
/I=D@0,O=tape option,L=06200,H=06257.
.
enter message
//
```

   or

   b. Print the memory dump

      1) Ensure that the line printer is ready and on-line

      2) // P8146

      3) // DUMPER The contents of memory as dumped to the disc are printed. CLEAR HALT TO RETURN TO DOS is displayed when the printing is completed.

Note: The disc must have sectors 06200-06257 formatted by the DOS Diagnostic pack. The DOS FORMAT program formats only sectors 0 thru 06177. If this is not done, the dump will NOT be successful.

FREQUENT QUESTIONS ABOUT CONFIG

The offscreen buffer is an area of memory which is dedicated for storing oversized reformat command strings or for building large data blocks which are being prepared for output to tape. The size of the offscreen buffer is a maximum of 2013 characters, depending on the number and size of the screens configured for the system and the other features required. Entering 0000 instructs the system to default to the screen size. The offscreen buffer is no longer automatically generated for 7008 systems, but must be designated each time CONFIG is run. This buffer is located in RAM immediately below 07000 and just after the user tables. This area of RAM is permanently assigned to this buffer. The buffer takes space normally used for extra user tables.

2. LPOUT is checked when DATA is initialized to see if it matches the type of printer specified in CONFIG. If not, DATA will not come up. However, DATA will initialize with NOPRNT (LPOUT equal to 4), regardless of the printer type specified in CONFIG and printing will take place if the attached printer is the printer selected under CONFIG. If the attached printer is different from that selected under CONFIG, the system will hang on a cutword.

CONFIG creates a single sector file (S56+C), if it does not already exist, which contains the source image of the latest CONFIG plus the date and time when CONFIG was last changed.

4. The user can change the master, supervisor, and debug password in CONFIG. Changing any password to CTRL @@@@@@@@ (binary zeroes) will signify that no password is requested.

5. The data read-back option provides a read-after-write check when data is being written from a buffer to the disc. This option can greatly impact system performance because it uses more system blocks and takes more time. The option is designed only for isolated installations with 1-4 terminals which do not have a backup media OR for installations that suspect they are having hardware problems with memory, disc, or controllers.

6. The second disc buffer is dedicated to paging in formats and should be specified when the system wil be handling a lot of verify or find mode activity. Use of the buffer should improve system performance when these activities are occurring. Use of this option may reduce maximum terminal support.

7. The code for all of the options of CONFIG (like check digits, keyboard types, etc.) are selected from a library file called R56+6. The load step for DATA includes /L=R56+L which is empty and EOPs to R56+6. If a check digit or keytop modification needs to be made, put the altered code in the relocatable file R56+L so that LOADOV can incorporate the changes when DATA is created.

MEMORY LAYOUT

72-K		96-K
0	Interrupt Locations	0
060	Screens, User Tables, etc.	060
01660	Format Vector Table	01660
02060	Screens, User Tables, Disc Request Queues, Sector Buffers, System Blocks, etc.*	02060
	Zone Pointer Table	
	Overlay Directory	
$DATA		$DATA
	Optional Resident	
	Fixed Resident	
047200	IOID5 (8121 Print Output)**	067200
047300	PUSER (Keyboard Input)	067300
047400	Overlay Area	067400
052000	Transfer Program Area or 12 Sector buffers for level 7	072000
057777		077777

*All memory not allocated to screens, user tables or executable code is organized into the following:

Sector Buffers	-	0400 (256) words number set at CONFIG time
System Blocks	-	0100 (64) words
Mini Blocks	-	020 (16) words
Micro Blocks	-	04 (04) words

**If there are no 8121 printers, this area will be used for system blocks.

ACTIVE JOB DIRECTORY IN RAM

JOBRAM	Pointer to first system block of directory
JOBDES,X1	In user table - points to specific JOBRAM entry

+0	Length of entry (words)			
+1	Jobname (9 ASCII characters)			
+2	left adjusted - blank fill			
+3				
	8	2	14	
+4	Length of batch directory entry	0	Disc sector address of job directory entry	Always in Zones 0 to 3
+5	Count of active terminals 0 = none			
+6 .	0		Points to Format entry in active Format/Value set Directory in RAM, one entry for each format	NFOR (in User Table) is the number of formats in the job
+6+NFOR	0		Points to Value Set entry in active Format/Value set Directory in RAM, one for each value set	NVAL (in User Table) is the number of value sets in the job
+63	1		Pointer to next system block (0=last block)	Last word in block
	1	8	15	

Length is 6+NFOR+NVAL : minimum = 7; maximum = 36.
Therefore, 1-9 entries in block.

If active terminal count is not zero, job cannot be deleted.

When count is = 0, entry can be deleted. However, deletion will wait until system block would overflow with the addition of another job.

Unused words are all binary zero. When length = 0, the next word > 0 is a valid length.

ACTIVE FORMAT/VALUE SET DIRECTORY IN RAM

FVSRAM   Pointer to first system block of directory

	1	7	1	7	1	7	
	0	A	0	B	0	C	
+1	page number of 1st page in this sector		0	sector address of this format object code sector			Format Entry (several entries possible for same format.)
	9		1	14			

	0	D	0	E	1	F	
+1	0	number of words in each Value Set item	0	sector address of the 1st sector of the Value Set			Value Set Entry (one for each active value set)
	1	8	1	14			

	0	D		0	E	1	F	
+1	1	00	number of words in each Value Set item	address of mini-block in RAM containing the Value Set				Value Set entry (when it is in a dedicated mini-block)
	1	2	6	15				

1	0	Pointer to next system block (0=last block)	63rd Word (64th word not used)
1	8	15	

001 < ABC < 999   three ASCII digits of format name
001 < DEF < 999   three ASCII digits of value set name

Entries are created when the job directory entry in RAM is built. Several jobs may share entries. A format entry is needed for each sector of format code. Four pages can reside in each sector. The "1st page number" will count as 0,4,8,12, etc., for each format and may appear in any random order in the directory.

Entries are on an even boundary.

An empty or deleted entry has binary zeroes in the first word.

PAGE DIRECTORY IN MEMORY (PAGDIR)

PAGDIR	Pointer to first micro block of directory
FORPTR,X1	In User Table; points to time word of appropriate page directory micro block

<table>
<tr><td>+0</td><td>USE COUNT
8</td><td colspan="2">Forward Pointer
16</td></tr>
<tr><td>+1</td><td>A</td><td>B</td><td>C</td></tr>
<tr><td>+2</td><td>0 < page # < 511
9</td><td colspan="2">PSB: address of system block
15</td></tr>
<tr><td>+3</td><td colspan="3">Time last referenced</td></tr>
</table>

An entry for each page of format code. Binary zeros for the name = no entry.

NOTE: The page directory consists of a linked list of micro blocks, one micro block for each page in memory. The list is kept in sequence by format and page number.

PSB: non-zero = address of system block containing page of code
zero = entry is a page-in request (no block assignment)

Time: non-zero = time this page was last referenced = t-86400 where t is seconds since midnight
zero = page-in request (same as PSB=0)

ABC is the format name in ASCII: 001 to 999.

IXRRAM

Displ.	Use
0	ASCII INDSET number (with bit 8 set) or zero
1	Sector address of the highest index level.
2	Bits 0-11 Starting column number of the key field in the data batch records. Bits 12-23 Key length in bytes.
3-4	"Who-is-using" bits. Bits 0-23 of word 3 followed by bits 0-7 of word 4 correspond, in that order, to the 32 possible terminals. For example, bit 1 of word 4 corresponds to terminal 25. If a "using" bit is on, the terminal it corresponds to has a record selected in the INDSET given by word 0. Words 5 and 6 specify which record. More than one "using" bit can be on, meaning more than one terminal has the record selected. A terminal may have the record selected, but it might be in a waiting state (with the "WAITING FOR TERM.XXX TO RELEASE INDSET iii" message on his screen) if another terminal currently has exclusive access to the record (see word 7).
5	If any bit in words 3-4 is on, this word has the sector address of the sector containing the selected record.
6	If any bit in words 3-4 is on, this word has the displacement in the sector of the first (header) word of the selected record.
7	User table address of exclusive accessing terminal, or zero if no terminal has exclusive access. When non-zero, the corresponding "using" bit will also be on.

Note: IXRRAM is contained in a chain of system blocks; each system block may contain a maximum of seven entries. The pointer to the next system block is in word +070 which has bit 0 = 1; the last block in the chain has 040000000 in that word. (Words 071-077 always contain binary zeroes.)

USER TABLE SKELETON DEFINITIONS

```
*       DCN      01401     IO instruction in PUSER Table points here.
KSCUR   EQU      1         Current position in KS queue.
KSRES   EQU      2         FWA of keystroke queue (UT ADDR + 0140).
FWA     EQU      3         First word of screen
LWA     EQU      4         Last word address + 1 of data portion of screen
MFWA    EQU      LWA       First word address of message line
SLWA    EQU      LWA       Last word + 1 of screen
CURC    EQU      5         Counts Level 4 until cursor change needed
CUR     EQU      6         Off screen copy of blinking word
SCR     EQU      7         Execute to place character on screen
PSROT   EQU      010       Character location of cursor (PZE SROT)
DEST    EQU      011       Location on screen of cursor
MODE    EQU      012       24 flags (see below)
OPPTR   EQU      013       Sector address of OPERATORS entry
JOBDES  EQU      014       Pointer to JOBRAM entry
FORPTR  EQU      015       Pointer to PAGDIR entry
FSROT   EQU      016       Character location of start of current field (PZE SROT)
FDEST   EQU      017       1st word of current field
REQ8    EQU      020       Overlay request for Level 8
RECLOC  EQU      021       Relative loc of current record in sector
WIDTH   EQU      022       -(Width of field)
MODPTR  EQU      023       Scratch word
LCR     EQU      024       Execute to get character from existing record
ACCP1   EQU      025       Pointer to first 4 accumulators. 0-3.
ACCP2   EQU      026       4-7
ACCP3   EQU      027       8-11
ACCP4   EQU      030       12-15
ACCP5   EQU      031       16-19
ACCP6   EQU      032       20-23
NFOR    EQU      033       Number of formats
NVAL    EQU      034       Number of value sets
SIZE    EQU      035       Number of words in data area of screen
MSIZE   EQU      036       - Size of screen in words
MSIZEB  EQU      037       - Size of screen in bytes
KBTYPE  EQU      040       Keyboard descriptor word from config
LASTKS  EQU      041       Last three keystrokes processed
TVUSER  EQU      042       Trail verify user pointer
TERMN   EQU      043       Terminal index 0-31, terminal printer number
BATCHN  EQU      044       ASCII batch ID - 6 characters - even boundary
*       EQU      045       continued
BINDOC  EQU      046       Binary document No.
TERMAS  EQU      047       Terminal number in ASCII right adjusted, blank filled
BATCH   EQU      050       Sector address of batch dir entry or batch index entry
LINKBS  EQU      051       Return address from LINKVS.
BINREC  EQU      052       Binary record number
MAXREC  EQU      053       Largest binary record number in the batch
NEXTUR  EQU      054       Unconditional keystroke return address
STATS   EQU      055       Operator statistics table
STATS1  EQU      056       2nd word of op. stats. table.
STATS2  EQU      057       3rd word of op. stats. table.
BINCOL  EQU      060       Binary column counter.
ALARM   EQU      061       Audible alarm instruction (IO KBTYPE or NOP)
SAVPTR  EQU      062       Save pointer for backspacing.
MODEXT  EQU      063       Mode extension flag word.
DISCRA  EQU      064       Return address when disc I/O complete (level 7)
VRECB   EQU      065       Non-zero when in verify mode.
RECHDR  EQU      066       Current record header word
RECSIZ  EQU      067       Binary record size
SECTOR  EQU      070       Current relative sector address
NEXTCR  EQU      071       Return address from BRM NEXTC
COLUMN  EQU      072       Pointer to column count on screen (FWA of status line)
ECODE   EQU      073       Error code.  Sign = Flasher
FX2     EQU      074       X2=Location in format (via BAL)
FX3     EQU      075       X3=Negative field width remaining to be keyed
VLROT   EQU      076       LCR pointer for verify
VDEST   EQU      077
        PZE      0         0100, lost keystroke routine from IOID+040
        BRM      KEYL      0101
        BRD      $-2       0102
QUE8RA  EQU      0103      Return address from QUE8 call.
```

TABLE SKELETON DEFINITIONS - CONTINUED

```
WPA     EQU     0104     Scratch word pair
WPA1    EQU     0105
WPB     EQU     0106     Scratch word pair
WPB1    EQU     0107
WPC     EQU     0110     Scratch word pair
WPC1    EQU     0111
WPD     EQU     0112     Scratch word pair
WPD1    EQU     0113
DQSLOT  EQU     0114     Current position in disc queue.
OV36BL  EQU     0115     Multi mode block address.
FMTST1  EQU     0116     1st 24 bits of format backspacing stack.
FMTST2  EQU     0117     2nd 24 bits of format backspacing stack.
MAXDOC  EQU     0120     Largest binary document number
MODE3   EQU     0121     24 Flags
LASSEC  EQU     0122     Last sector address of current batch
SSACOL  EQU     0123     Saved column number for return to keying position
RETCOL  EQU     0124     Column number for return to last position
RETNXC  EQU     0125     Saved 'NEXTCR'
RS      EQU     0126     Digits to right of decimal point, LZERO field
SCPBAD  EQU     0127     Used by screenprint.
KV      EQU     0130     Used by key verify.  byte 0 is scrambled char
MODE2   EQU     0131     More single bit flags, like 'MODE'
NEMODE  EQU     0132     Next mode selection, byte 0
STATS3  EQU     0133     4th word of op. stats. table
SNDINQ  EQU     0134     2770 - used for send inquiry
RCVINQ  EQU     0135     2770 - used to receive inquiry response
RCVSPL  EQU     0136     2770 - used for mini block address - spool info
                0137     Reserved

Keystroke queue at 0140 - 0167

WPE     EQU     0170
WPE1    EQU     0171
WPF     EQU     0172     Extra word for S80-5C
WPF1    EQU     0173     Extra word for S80-5C
MAPWRD  EQU     0174     Mapping word for screen addressing
                0175-0177 Reserved
```

USER TABLE FLAG WORDS

MODE

0	1=Normal mode from keyboard, 0=feed thru or V#
1	1=Prog ctrl off, 0=on
2	1=Release key in process, 0=not
3	1=Verify active, 0=inactive
4	1=Ver corr (1 char) in process, 0=not
5	1=Waiting at end of field (must RLS/Left Zero), 0=not
6	1=1st pass (disc to screen), 0=not
7	1=ASD on, 0=not
8	1=Special mode
	Note: Bits 8,9, 10 = 0 is entry mode
9	1=Verify mode
10	1=Find mode
11	1=Skip/Release inhibit (must fill), 0=not
12	1=Character input inhibited, 0=allowed
13	1=New record, 0=existing record
14	1=Aux dup inhibited, 0=not
15	1=Index inhibited, 0=not
16	1=Psuedo new record, 0=not
17	1=No job assigned, 0=assigned
18	1=Must enter inhibit, 0=not must enter
19	1=Perform validation checks, 0=not
20	1=1st column of record, 0=not 1st col of rec
21	1=Waiting for minus key (LZ FLD), 0=not
22	1=Not 1st col of field, 0=1st col of field
23	1=Left zero field, 0=not left zero field

MODE2

0	Cursor state flip-flop
1	Verify reconstruct
2	Search-state on
3	Extended precision accumulators
4	Correction made in K.V. to current field If reverification option specified in CONFIG
5	Mode X,T (Decentralized batch transmission) in process.
6	Reserved
7	Reserved
8	SYSIN is active
9	NEWFMT 1=new style format
10	TRMBIT current logical term value (1=true)
11	EXPBIT current logical expression value (1=true)
12	IFTHEN 1=currently between "IF" & "THEN" in format
13	Right justify is active on key verify
14	Reserved for special user
15	Reserved for special user
16	Reserved for special user
17	Reserved for special user
18	Reserved
19	Reserved
20	No disc record (NDR)
21	INDSET-FIND mode in progress
22	1=In a record, 0=not in a record
23	Waiting for new mode char

MODE3

0-18	Reserved
19	1=DOC KV # 2 (set in DOC key-verify initiation (in overlay 38 entry 6)). This bit remains on during the DOC Key-verifying. It is turned off when new record is hit and the system reverts back to entry mode.
20	1=DOC KV # 1 (set by KVDOC command in format). This bit remembers that a KVDOC was executed. At NXTREC, if this bit is on then the document key-verify operations are initiated by calling overlay 38 entry 6. The overlay clears this bit.
21	Reserved
22	1=Out-of-balance (set at SREC in key ver) This is what the OUTBAL test tests in the format
23	1=Batch has marked DOC (set by mark command in format)

MODEXT

0	Unconditional release in process
1	Mode key
2	Record up
3	Record delete
4	Record insert
5	Home key
6	Entry mode bringup
7	DOC up (always ignore search-active)
8	DOC down (always ignore search-active)
9	Immediate-next-record (ignores any search-arg)
10	This-record (i.e., no alteration of sector, recloc)
11	Record down
12	Prog key
13	Do-not-update (BINREC/BINDOC) (Used in conjuction with bits 6, 10 and maybe others)
14	Restore (return-to-keying-position in entry mode)
15	Save-keying-position (on only during record release)
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved

VRECB

0	Record correction (Prog or rec ins)
1	Field correction (shift field forw)
2	Column correction (shift corr/reset)
3-23	Reserved

DISC SECTOR POOL AND REQUEST QUEUES

There are three disc sector pools pointed to by the following fixed residents words:

BUFLST - Free buffer list (always starts with a dummy entry that has no buffer attached)
DISCQ1 - Pending I/O (all requests except PUTREL)
DISCQ2 - Pending I/O (PUTREL requests, only)

Word -1	Forward Pointer							
Word 0	Q 1	N 1	A 1	R 1	U 1	E 1	RC 3	address of "here" pointer in 256 word disc buffer 15
Word 1	W	Contents depends on type of request (see below)						
Word 2	H	Disc Read/Write Request Caller (see below)						

Word 0:

Q	1=disc I/O request pending, 0=request complete
N	1=non-standard request (see below)
A	1=repeat the write request (a 2nd PUTREL/PUTHLD initiated before first PUTREL complete)
R	1=multiple read requests exist for this sector
U	Unused
E	1=Unrecoverable I/O error. This bit may be set because of a reject (invalid sector address, etc.) or because of 8 unsuccessful retries on a disc I/O error. With this bit set, PRODSC will cause the error return to be taken on the initiating GETSEC, PUTHLD, etc.
RC:	After 8 retries are counted, this 3 bit counter will overflow setting the error bit (bit 5).

Word 1: W 1=write request, 0=read request

The remainder of Word 1 may have 3 different forms, depending on the type of request.

For standard read and write requests:

W	0	20 bit relative sector address

For non-standard requests, there are two formats. The first is for an overlay load:

0	0	7 bits reserved	3 bit length -1	relative starting sector address of overlay in zone 000

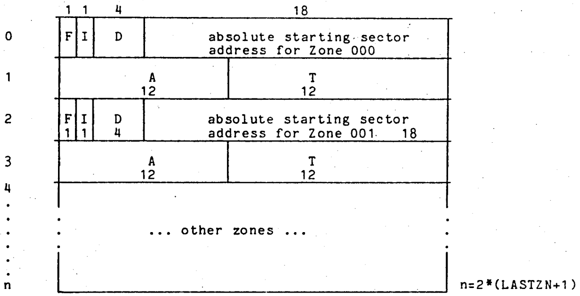
The second is for single sector absolute sector I/O (TRAN50)

W	1	4 bit IDOS logical drive	18 bit absolute sector address

Word 2: H 1=word is User Table address of level 7 caller
0=word is address of Submonitor Control Block of lever 8 caller

ZONE POINTER TABLE

@ZONPT	pointer to first word of table
LASTZN	highest numbered zone defined to the system



F: 0 = sectors available in this zone
1 = this zone is 100% full

I: 0 = allocation permitted
1 = allocation inhibited

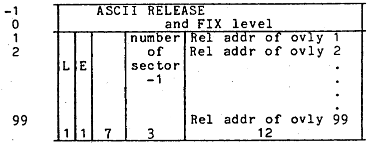
D: 4-bit IDOS logical drive number of zone (0 - 11)

A: Number of sectors available for allocation in this zone (12 bits)

T: Total number of sectors in this zone (12 bits)

OVERLAY DIRECTORY

@OVDIR	Points to the word preceding the overlay directory in RAM (since the RAM overlay directory is preceded by a pair of words containing the release and fix level, @OVDIR points to the second of these words)
M#OVLY	The largest overlay number defined



L = 1, Overlay is loaded (in DAT000)
E = 1, Overlay exists

Submonitor Control Blocks

Level 8 processing is controlled by a multi-tasking monitor that supervises the execution of several tasks. These tasks, called submonitors, are each controlled by an 8 word Submonitor Control Block (SCB):

0		address of disc queue slot if sector buffer attached	
1	Submonitor state		*1
2		restart address	
3	Reserved		*2
4	Use depends on state		
5	X1 restoration value		
6	X3 restoration value		
7	ASCII name of Submonitor		

*1 States: 0 - waiting for CPU
1 - running
2 - not in use
3 - waiting for extenal restart
4 - waiting to call (not currently used)
5 - waiting to start disc IO
6 - disc IO complete
7 - waiting for LPRINT

*2 Displ. 4 word used:
4 - address of submonitor being called
5 - sector address
6 - disc queue entry address of non-standard disc request

DATA IV currently has the following level 8 submonitors:

Submonitor Name	Title
OVL	Overlay Submonitor
TRA	Transfer Program Submonitor
PRT	Print Submonitor
ALO	Allocation/Deallocation Submonitor
DYN	DYNAMO Submonitor
COM	Communications Submonitor
SPL	Spool Submonitor

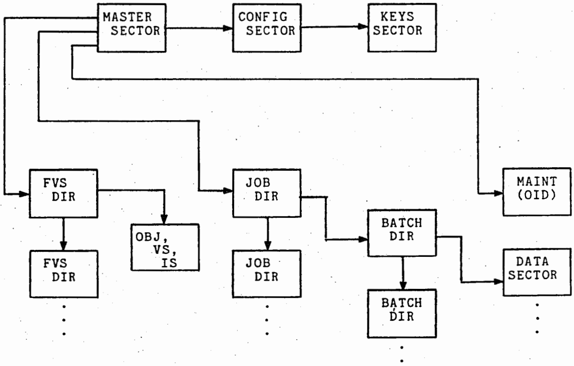
ZONE LAYOUT

	Sector	
Zone 000	0000	Scratch area
	0001	Allocation Sector - Zone 000
	0002	Master Sector
	0003	CONFIG Sector
	0004 - 0015	Reserved
	0016	Reserved for NP/80
	0017	Master Sector (Copy)
	0020 - 7677	Directories, Overlays, Data, etc.
Zone 001-377	0000	Scratch area
	0001	Allocation sector for this zone
	0002 - 0017	Reserved
	0020 - 7677	Directories*, Data, etc.

*Directories, format object code, and value sets may be in zones 0-3, only. Therefore, they require a 14-bit directory address: 2 bits for the zone and 12 bits for the relative sector address.

Data sectors and index sectors require a 20-bit address: 8 bits for the zone and 12 for the relative sector address.

DATA IV DISC LAYOUT



SECTOR STRUCTURE

The following standard 4-word header is at the beginning of all but the allocation and overlay object code sectors:

Word 0: Forward Pointer ....ZZZZZZZZSSSSSSSSSSSS

Forward pointer indicating the relative sector address of the next sector in this chain. Zeros indicate the last sector of the chain.

Word 1: Backward Pointer....ZZZZZZZZSSSSSSSSSSSS

Backward pointer indicating the relative sector address of the previous sector in this chain. Zeros indicate the first sector of the chain.

Word 2: Here Pointer .......ZZZZZZZZSSSSSSSSSSSS

Here pointer indicating the sector's own relative sector address.

Word 3: Code word.

If this sector contains format object code, a value set, or an index set information sector or index key sector, this word contains the ASCII number of the format, value set, or index set. (Bit 8 or 16 is set according to the rules used in the FVS directory; ie, bit 8 is on for index set numbers and bit 16 is on for value set numbers.)

For all other types of sectors, the following format is followed:

CCCCDDDDDDZZSSSSSSSSSSSS

CCCC 4-bit code for sector type:

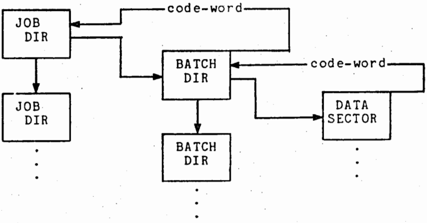
0000 - Master Directory (remainder of code word also zero)
0100 - Job Directory (remainder is zero)
0010 - Batch Directory (remainder is zero)
0101 - Maintenance or OID Sector (remainder is zero)
0011 - Format/Value-Set Directory (remainder is zero)
1000 - data sector which is not full} (remainder of code
0000 - data sector which is full } word non-zero)

SECTOR STRUCTURE - CONTINUED

NOTE: The following applies only to Batch Directory and Data Sectors.

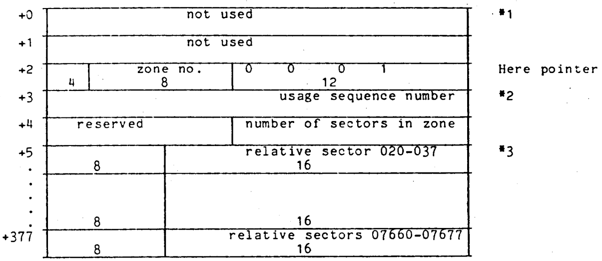
DDDDDD	For Batch Directory Sectors, this contains the leftmost 6-bits of the 8-bit displacement of this entry in the Job Directory Sector. For Data Sectors, this contains the leftmost 6-bits of the 8-bit displacement of the entry in the Batch Directory Sector.
Z....S	The zone and sector address of the Job Directory Sector or Batch Directory Sector for Batch Directory and Data Sectors, respectively.

In other words, the code word acts as a pointer for data sectors that points back to the batch directory entry for it. Similarly, the code word acts as a pointer for batch directory sectors that points back to the job directory entry for it.



ALLOCATION SECTOR

This sector is always relative sector 0001 in every zone. Note that the first 16 sectors (0000-0017) of a zone are not allocated by DATA IV. The relative sector 000 of zone 000 cannot be used, since the here pointer would be zero.



*1: bit 0 = 1, zone has been accessed.

*2: Each time OPENDB is called (eg, when DATA is brought up or MAINT is executed), this sequence number is checked to be identical in all DATzzz files. If they don't match, an incomplete restore may have occurred and MAINT must be run to set them all equal. If they all match, then they are all bumped by one. They will recycle to 0.

*3: 251 word allocation table for this zone. Bits 8 to 23 of each word represent 020 sectors of a zone. Zero bits indicate sectors in use or not assigned to the zone. One bit indicates sectors available for allocation.

MAINTENANCE (OID) SECTOR

	Standard 4 word header	
+0004 to +0077	DAILY LOG INFORMATION	*1
+0100 to +0207	DETAIL DISC ERROR LOG	*2
+0210 to +0377	START AND STOP TIMES FOR DATA	*3

*1 Variable number of entries, 3 to 14 words per entry. A word of all ones, indicates the end of the entries.

Word 1 = Day
2 = # Drives for this Entry
3 = Error Count for Drive #0
.
.
.
12 = Error Count for Drive #11

*2 Four-word entries.

Word 1 = Day
2 = Bits 0-8, reserved; bits 9-23, time (0-86400)
3 = Bits 0-23, status returned from $JDISC
4 = Bits 0, read or write
1, reserved
2-5, logical drive number
6-23, sector address

*3 Three-word entries.

Word 1 = Day
2 = Start Time
3 = Stop Time

NOTE: The "MAINT" flag is in word 0212. To bypass running MAINT after abnormal DATA IV shutdown, set word 0212 to a non-zero time. (0212 corresponds to the shutdown time for the last execution of DATA IV and is set to zero when DATA is initialized.)

MASTER SECTOR

0	0	00003				Pointer to CONF
1	0	00000				Back Pointer
2	0	00002				Here Pointer
3	0	00000				Code Word
4	(t-86400) seconds since midnight = TIME					
5	ASCII day number (001-366)					
6	0	first sector addr of FVS dir 14				
7	0	first sector addr of JOB dir 14				
010	0	first sector addr of MAINT dir 14				OID sector
011	reserved					
012 013	ASCII release level name (eg. AI14-D)					
014 015 thru 0156	L 1	E 1	reserve 7	number secs-1 3	rel sec addr overlay 1 rel sec addr overlay 2 rel sec addr overlay 99	*1
0157	reserved					
0160 0161 thru 0177	0 0 . 0	zones 000 thru 017 zones 020 thru 037 ... zones 0360 thru 0377 16				*2
0200	reserved 8	no. zones - 1 8	largest zone no. 8			
0201	reser. 6	absolute sector address of zone 000 18				set by MAKESY
0202 thru 0377	reserved					

*1 L=1, overlay is loaded
E=1, overlay exists

The rightmost 12 bits of each of these words contains the relative sector address (within zone 0) of the first sector of that overlay.

*2 A 16 word table maintained by ZONEMA. Bits 8 to 23 of each word represent 020 possible zones. One bits indicate defined zones. Zero bits indicate undefined zones.

CONFIG SECTOR

<table>
<tr><td></td><td colspan="5">Standard 4 word header</td><td>*1</td></tr>
<tr><td>+4</td><td colspan="5">Time of last change (t-86400 seconds)</td><td></td></tr>
<tr><td>+5</td><td colspan="5">Day of last change</td><td></td></tr>
<tr><td>+6</td><td>KB
1</td><td>AUD
1</td><td>-Reserved-
(6)</td><td>Size of screen in words /16
(6)</td><td>FWA of screen /16
(10)</td><td>*2</td></tr>
<tr><td>+0046-
+0105</td><td colspan="5">-Reserved-</td><td></td></tr>
<tr><td>+0106</td><td colspan="4"></td><td>Number of disc buffers 5</td><td>*3</td></tr>
<tr><td>+0107</td><td colspan="5">Reconfiguration password (scrambled) 0=no password</td><td>*4</td></tr>
<tr><td>+0110</td><td colspan="5">Reserved</td><td></td></tr>
<tr><td>+0111</td><td colspan="5">Print/log control password (scrambled) 0=no password</td><td>*4</td></tr>
<tr><td>+0112</td><td colspan="5">Supervisor password (scrambled) 0=no password</td><td>*4</td></tr>
<tr><td>+0113</td><td colspan="5">Reserved for future (3 words)</td><td></td></tr>
<tr><td>+0116</td><td colspan="5">SYSFIG (CONFIG)</td><td>*5</td></tr>
<tr><td>+0117</td><td colspan="5">SYSFG1 (CONFG1)</td><td>*5</td></tr>
<tr><td>+0120</td><td colspan="5">HASP Communications Word (SYSFG2/CONFG2)</td><td>*5</td></tr>
<tr><td>+0121</td><td colspan="5">2770 Communications Word (SYSFG3/CONFG3)</td><td></td></tr>
<tr><td>+0122</td><td colspan="5">Performance Parameters</td><td>*6</td></tr>
<tr><td>+0123</td><td colspan="5">reserved</td><td></td></tr>
<tr><td>+0124</td><td colspan="5">reserved</td><td></td></tr>
<tr><td>+0125</td><td colspan="4"></td><td>COMWT8 7</td><td></td></tr>
<tr><td>+0126-
+0137</td><td colspan="5">Reserved</td><td></td></tr>
<tr><td>+0140-
+0142</td><td colspan="5">Jobname of current log file</td><td></td></tr>
<tr><td>+0143-
+0144</td><td colspan="5">Batch # of current log file</td><td></td></tr>
<tr><td>+0145</td><td colspan="5">PRSTAT flags</td><td></td></tr>
<tr><td>+0146-
+0225</td><td colspan="5">Reserved for terminal and 8121 initialization</td><td>*7</td></tr>
<tr><td>+0226-
+0265</td><td colspan="5">SPOOL PARAMETERS</td><td></td></tr>
<tr><td>+0266-
+0306</td><td colspan="5">SPOOL Conversion Control Table (17 words)</td><td></td></tr>
<tr><td>+0307-
+0336</td><td colspan="5">Reserved (24 word area)</td><td></td></tr>
<tr><td>+0337-
+0377</td><td colspan="5">Reserved for user (33 word area)</td><td></td></tr>
</table>

* Footnotes on next page

Footnotes for CONFIG Sector

*1 Forward pointer points to KEYS sector

*2 One word for each terminal
KB:0=Source Keyboard,1=029
AUD:0=no audible alarm

*3 Minimum is 4

Recommended value is number of terminals + 2; additional buffers needed for communications, $EDIT, etc.

*4 Defaults for Passwords:
8 blanks - Reconfiguration or Master Password (020430614)
12345678 - Print/Log (052675507)
PASSWORD - Supervisor (070702334)

*5 System Configuration Words:
Word 1 (SYSFIG)

bit(s)	
0	0=48 char screen, 1=81
1	1=MOD II with extended memory; 0=no extended memory
2	0=no channel adaptor
3	0=no memory log
4	0=no card reader
5	0=no printer, 1=use IDOS printer
6	0=9-track tape, 1=7-track tape
7	0=low density tape, 1=high density tape
8	0=no shift override of field type, 1=allow shift override
9	0=no concurrent bisync, 1=concurrent bisync
10	0=count records, 1=count documents
11	0=one beep/error; 1=one beep/keystroke after error
12	0=EBCDIC, 1=ASCII output to tape
13	0=LAM/8436 1=LAM/8437,
14	0=MLAM/8436 1=MLAM/8437,
15-17	0=production system, non-zero = screen number +1 of DEBUG terminal. (3 bits)
18-19	number of tape decks (2 bits)
20-23	number of disc drives (0-11) (4 bits)

Word 2 (SYSFG1)

0	0=no 12 channel printer for received data (BISYNC)
1	0=extended error recovery (BISYNC), 1=no extended error recovery
2	0=do not count FIND mode statistics, 1=count FIND mode
3	0=DATA IV print spooling, 1=bisync direct print
4	0=field reverification, 1=no reverify option
5	0=do not keep detail operator statistics by batch, 1=do keep by batc
6	0=no decentralized batch transmission, 1=yes
7	0=unconditional decentralized batch transmission, 1=conditional
8-10	Reserved
11-13	3 bit Baud flag: 0 = 600 1 = 1200 2 = 2000 3 = 2400 4 = 4800
14-18	number of 8121 printers
19-23	number of terminals

Word 3	(SYSFG2) HASP Workstation Communications
0	0=no HASP Workstation, 1=yes
1	0=no card reader, 1=yes
2	0=512 byte block message size, 1=400 bytes
3	0=manual answer, 1=auto answer
4-5	Compression: 00 = no compression 01 = trailing blank truncation 10 = blank compression 11 = full compression
6	0=do not send spanned records, 1=send spanned records
7	0=do not send transparent text, 1=send transparent text
8	0=no timeout, 1=20 second idle timeout
9	0=terminal-CPU, 1=terminal-terminal
10	0=primary station, 1=secondary station
11	not used
12-18	Console message size 1-120 bytes
19-23	Console number 0-31

*6

bits	contents
0-2	not used
3-6	BUFBLK
7-9	PAGSEC
10-12	PAGLIF
13-15	DYNBAL
16-17	NKSMIN
18	not used
19-23	LEV78B

*7 060 words of binary zero, to be set by $SETPRINT command words 0146 to 0165 are 8121 page print control: 1 word for each possible terminal printer.

byte 0 = number of printed lines per page
byte 1 = number of increments to advance each line (1 increment = 1/48 inch)
byte 2 = total number of lines per page

Words 0166 to 0225 are 8121 Terminal control: 1 word for each possible screen. This word is a copy of the TERMN word from the User Table for the screen.

KEYS SECTOR

The KEYS sector contains codes for translating hardware keyboard input to the proper software code desired. The following table lists the software codes for the DATA IV Control Functions.

OCTAL CODE	CONTROL FUNCTION	KEY(S)
200	Record Up	REC ¦
201	Backspace Character	FIELD <--
202	Skip Right Character	FIELD -->
203	Record Down	REC ¦
204	Release	REL
205	Mode	MODE
206	Record Insert	SHIFT REC ¦
207	Erase	SHIFT ERASE/HOME
210	Home	ERASE/HOME
211	Reset	CORR RESET
212	Record Delete	SHIFT REC ¦
213	Correct	CORR RESET
214	Skip (same as 243)	SHIFT REL
215	Left Zero	LEFT ZERO
216	Left Zero	LEFT ZERO
217	Skip Right Field	SHIFT FIELD -->
220	Backspace Field	SHIFT FIELD <--
221	Auto Skip/Dup	ASD
222	Program Control	PROG CTRL
223	Illegible	?
224	Valid	VALID
225	Index	INDEX
226	Program 1	PROG 1
227	Program 2	PROG 2
230	Program 3	PROG 3
231	Program 4	PROG 4
232	Program 5	PROG 5
233	Program 6	PROG 6
234	Print Screen	CTRL DUP
235	Display Accumulator	TOTAL
236	Document Down	CTRL REC ¦
237	Restore	CTRL REL
240	Multipunch	SHIFT MULT/SKIP
241	Duplicate	DUP
242	Auxiliary Duplicate	AUX DUP
243	Skip (same as 214)	SKIP
244	Document Up	CTRL REC ¦
245	Search Switch/FIND Mode	CTRL ERASE/HOME
246	Inhibited Key	
247	Reserved for custom systems, normally inhibited	
250	Character Insert*	CTRL FIELD -->
251	Character Delete*	CTRL FIELD <--

*Must be optioned for in OPTION.

NOTE: On keypunch-style keyboards, NUMERIC replaces SHIFT.

JOB DIRECTORY (on Disc)

```
                 |---------------------------------------------------------|
                 |    Standard 4 word header                               |
                 |---------------------------------------------------------|
          +4     |    Time of last change (t-86400 seconds)                |
                 |---------------------------------------------------------|
          +5     |    Day of last update                                   |
-----------------|---------------------------------------------------------|
 0    |   +6     |    Word length of entry (even binary)                   |
      |          |---------------------------------------------------------|
+1    |   +7     |    Job name (1-9 ASCII characters)                      |
      |          |    left adjusted blank fill.                            |
      |          |1 1 1 1 1     5                 14                       |
+4    |   +012   |S|O|B|D|X| reserved| Batch Directory 1st sector address  |  *1
+5    |   +013   |NFOR     |NVAL    |NACCP          |U|   SIZE              |
      |          |# of     |# of    |number of      | | binary record size  |
      |          |formats  |V. Sets |accumulators   | |    in bytes         |  *2
      |          |1-15     |0-15    |0-23           | |                     |
------|----------|---------------------------------------------------------|
+6    |   +014   |  Scrambled password required for access                 |
      |          |     0 = No password required                            |
------|----------|---------------------------------------------------------|
+7    |   +015   |     MAXSIZ          |   starting sector address of      |
      |          |  binary maximum     |      Batch Index chain            |  *3
      |          |record size (bytes)  |      Zero = No Batch Index   14   |
------|----------|---------------------------------------------------------|
+010  |   +016   |  Packed decimal format and value set id numbers.        |
  .   |          |  Left to right with formats first.  Last half           |
 MAX  |          |  word has binary zero fill (if any).  Extra word        |  *4
      |          |  (to make even length) has binary zero fill (if         |
+030  |          |  any).                                                  |
......|..........|---------------------------------------------------------|
```

An entry in the job directory cannot be moved once defined. All entries are even length.

*1
S = Job has been "selected" by $SELECT command
O = Batches have been ordered for transfer by $ORDER command
B = BUSY - Job is temporarily unavailable due to $PURGE or $ORDER processing
D = Double Accumulators are defined for this job
X = Mode X-T enabled for this job ($XTSET)

*2
NFOR = Number of formats defined for this job (4 bits). Range: 1-15. Same value as NFOR in User Table.
NVAL = Number of Value Sets defined for this job (4 bits). Range: 0-15. Same value as NVAL in User Table.
NACCP = Highest accumulator number defined for this job. (5 bits). Range: 0-23.
U = Special user use (1 bit).
SIZE = Record size for fixed length records (10 bits). 0= variable length records.

*3
MAXSIZ = maximum record size permitted under variable length records (10 bits). (0=fixed length records)

Program MAINT will clear the BUSY flag and delete the Batch Index.

*4 Packed decimal example:

```
Formats IDs       |  9  |  4  |  3  ||  3  |  9  |  2  |

Decimal to
    Binary        | 1001  0100  0011 || 0011  1001  0010 |

Regroup        | 100 | 101 | 000 | 011 || 001 | 110 | 010 | 010 |

Binary to
    Octal      |  4  |  5  |  0  |  3  ||  1  |  6  |  2  |  2  |
```

Therefore, the two format id's 943 and 392 would be packed into the octal word 45031622.

BATCH DIRECTORY (on Disc)

		Standard 4 word header					
0	+4	Batch id "number" (1-6 characters) left adjusted ASCII with blank fill					*1
2	+6	4 bits of zero	relative sector address of first data record 20				
3	+7		record count	undefined if bit 0=1			*2
4	+010	- reserved -	document count				*3
5	+011	State and request flags (defined on next page)					
6	+012	Time last used (t-86400) binary seconds					*4
7	+013	Day code (3 ASCII digits)					
010	+014	Transaction code (3 ASCII characters)					
011		4 bits of zero	relative sector address of last data record - LASSEC 20				*5
012	+016	L	Last entry operator id 23				*6
013	+17	L	Last verify operator id 23				*6
014	+020	H 1	C 1	W 1	Offset from sector header 7	sector address of next entry 14	*7
015	+021	- Reserved for special user use -					
016	+022	4 words for each accumulator 12 digit ASCII character right justified Up to 23 accumulators = 92 words					*8

Minimum entry = 14 words if no accumulators
Maximum size = 106 words if 23 accumulators
All entries are fixed size for each job
A maximum of 18 entries per sector (no accumulators)

*1 - 0 = no entry (Purged or never existed)
*2 - 0 = empty batch (no records)
*3 - 0 = no documents (no $DOCHDR records)
*4 - Always negative
*5 - If the whole word is 0, the address is assumed to be unknown.
*6 - L = 1 for last operation on batch, else L = 0
*7 - For ordered job, points to next batch in order chain. H = 1 for head, C = 1, part of order chain, W = used during $ORDER. (Offset is always even, so low order zero is dropped.) Set = 0 when batch created.
*8 - Accumulators are set to ASCII zeros when the batch is created.

"State and Request Flags" word in Batch Directory Entry
(Word 5 of batch directory entry)

1	reserved		
2	1=complete	(COMP)	
3 4	00=Find 10=Verify 01=Entry 11=Verify-Complete		← Last operation performed on this batch
5	1=Has been written to output device	(WWR)	
	1=Has been read from input device	(WRD)	
7	1=Has been reformatted	(WRF)	
8	1=Was created by reformat	(CBRF)	
9	1=Display request	(DIS)	
10	1=Purge request	(RPUR)	*
1	1=Write request	(RWR)	*
12	1=Read request	(RRD)	*
13	1=Reformat request	(RRF)	*
14	1=Locked	(LOCK)	*
15	1=Batch in use	(INPR)	*
16	1=Active Log File	(LOG)	*
17	1=Document Marked	(MARK)	
18	1=Indset Batch	(XSET)	*
19 20 21	- Reserved -		
22 23	Reserved for special user use		

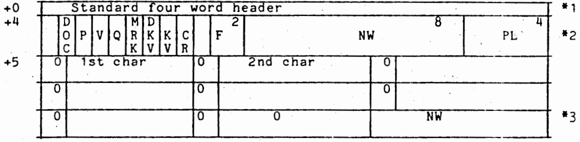
* If any of these flags are on, a "batch active" condition exists and access to the batch is inhibited

BATCH INDEX

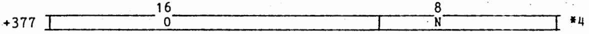
	Standard 4 word header
+4	Batch Name
+5	
+6	Displacement in Batch Directory sector
+7	Batch Directory sector address
	.
	.
	more entries
+0377	

Note: The Batch Index is deleted when MAINT or CLEAN is run and is then rebuilt by DATA at job bringup time.

DATA SECTORS



Sector trailer (1 word) when code word is negative:



*1 Special flag in bit 0 of Code word:
0 = sector is full; N = 0400 is implied
1 = sector is not full; word 0377 has N of 004-0377

*2 Record header word:

DOC:	This record starts a new document
P:	Program level changed in record
V:	Validation override used
Q:	Unintelligible key used
MRK:	This document has been "marked"
DKV:	Has been document-key verified
KV:	Has been key verified
CR:	Has been altered since entry
F:	# of trailer bytes (1-3).
NW:	Number of words in entry. Record size in bytes = (NW-1)*3-F.
PL:	Program level 0 to 14

*3 NW: Number of words in entry

*4 Sector trailer word (only present if bit 0 of code word is 1):

N: Number of words being used in this sector (004-0377).
Also is relative address in sector where unused area begins.

Algorithm for # of records in a sector:

Total sector length = 256 x 3 = 768
Sector Header = -4 x 3 = 12
Available for data = 252 x 3 = 756
(words) (chars)

$$NWR = \left[\frac{RS + 3}{3} \right] + 1$$

$$NRS = \left[\frac{252}{NWR} \right]$$

Where, RS = Record Size in characters
NWR = Number of Words per Record
NRS = Number of Records per Sector

$[\quad]$ = integer divide (ie, truncate to integer)

NOTE: On jobs with variable length records, the record mix will have to be considered. The system will try to fit each record into the current sector.

FORMAT/VALUE SET DIRECTORY (on Disc)

<table>
<tr><td></td><td colspan="6">Standard 4 word header</td><td></td></tr>
<tr><td>+4</td><td>0</td><td>A</td><td>0</td><td>B</td><td>0</td><td>C</td><td rowspan="3">*1</td></tr>
<tr><td>+5</td><td>reserved
6 bits</td><td>S
D</td><td>Screen
Code
2 bits</td><td>G</td><td colspan="2">starting sector address of
format object code
14</td></tr>
<tr><td>+6</td><td colspan="4">record size (bytes)
10</td><td colspan="2">Reserved</td></tr>
<tr><td></td><td>0</td><td>A</td><td>0</td><td>B</td><td>1</td><td>C</td><td rowspan="3">*2</td></tr>
<tr><td></td><td colspan="4">reserved</td><td colspan="2">starting sector address of
Value Set
14</td></tr>
<tr><td></td><td colspan="5">reserved</td><td>number of words
in Value Set item
8 bits</td></tr>
<tr><td></td><td>0</td><td>A</td><td>1</td><td>B</td><td>0</td><td>C</td><td rowspan="3">*3</td></tr>
<tr><td></td><td>reserved
4</td><td colspan="5">starting sector address of
Information sector
20</td></tr>
<tr><td></td><td colspan="6">reserved</td></tr>
<tr><td></td><td colspan="6">binary 0</td><td rowspan="3">*4</td></tr>
<tr><td></td><td colspan="6">not defined</td></tr>
<tr><td></td><td colspan="6">not defined</td></tr>
</table>

ABC is the 3-digit ASCII name of the format, Index Set, or
Value Set: 000 < ABC < 999
bit 16 = 1 (Value Set)
bit 8 = 1 (Index Set)

Formats, Value Sets, Index Sets, and empty entries can be intermixed in any order.

*1 - 3 word entry for format ABC

SD: 0=single accumulator width, 1=double precision accumulators

Screen Code: 0: Quarter screen, 1: Half, 2: Full, 3: Double
Screen code includes the message line. These are the minimum screen sizes for format.

G: 0=old style formats, pre-release B0. 1=new sytle formats, release B0 and later.

*2 - 3 word entry for Value Set ABC
*3 - 3 word entry for Index Set ABC
*4 - 3 word empty entry

84 entries per sector. The first word of each entry must contain a valid value.

FORMAT OBJECT CODE SECTOR

	Standard 4 word header -- fourth word is format name in ASCII
+4	Four 63 word pages of compiled calling sequences. page 0
+0103	page 1
+0202	page 2
+0301	page 3

Notes:

The code word is actually the ASCII format name. By the nature of ASCII code, this yields a code word with bits 0-3 = 0011.

Compiled calling sequences may not be split across pages. Therefore, several NOP instructions (06700000) may appear at the end of a page.

The last sector of the format may not be full, i.e., it may contain less than four pages. The remainder of the sector is garbage.

VALUE SET SECTOR

	Standard 4 word header -- fourth word is Value Set name in ASCII
+4	Time Value Set was complied
+5	Day Value Set was complied
+6 thru +0377	String of Value Set entries

Notes:

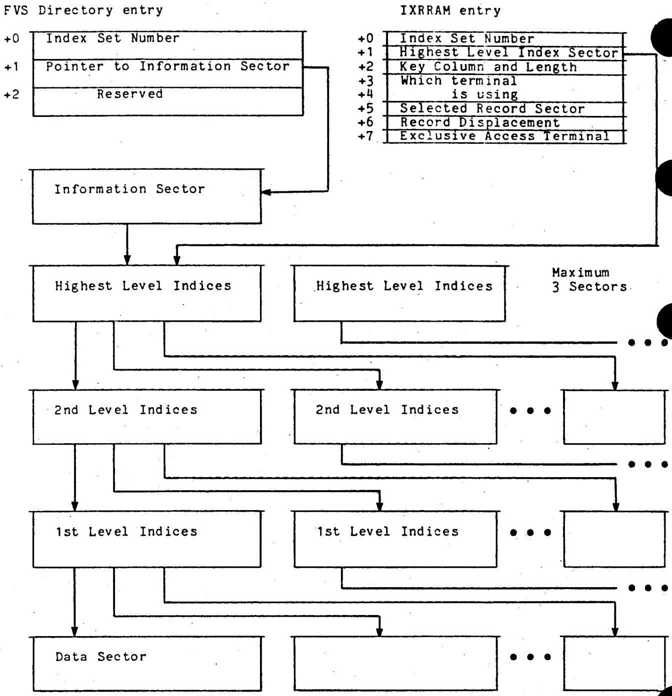
The code word is actually the ASCII Value Set name. By the nature of ASCII code, this yields a code word with bits 0-3 = 0011.

Each Value Set entry is 1 to 250 words long, the length being defined in the Format-Value Set Directory entry.

Each entry is a string of ASCII characters, starting on a word boundary, left justified with garbage fill. Entries may not span across sector boundaries.

Unused words at the end of a sector are filled with binary zeros. Unused words at the end of the last sector of the Value Set are filled with 037777777.

INDEX SET ORGANIZATION



INDEX SETS (on Disc)
Information Sector

	Standard 4 word header	
+4	Reserved	
+5	ASCII INDSET number with bit 8 on	
+6	INDSET number in binary	
+7	Sector address of highest index level	*1
+10	First column number of key	
+11	Number of columns in key	
+12	Job name of the data batch	
+15	Batch ID of the data batch	
+17	Sector address of first data batch sector	
+20	Sector address containing batch directory entry for data batch	
+21	Displacement of batch directory entry for data batch	
+022 . . . +0377	reserved	

*1 Index level sectors are built by having an index entry for each sector in the data batch. The highest index level must be 3 sectors or less. Index levels will continue to be built from the data batch until a level with 3 or fewer sectors is reached. Each index level differs from the first only in that the sector address points to a sector in the preceeding index level.

INDEX SECTOR

		Standard 4 word header	
+4	L 1	Sector address of data batch sector or next lower level index sector	*1
+5 . .		Key value of highest key in data sector or previous level index sector (left justified with blank padding to word boundary)	Index entry
. . . . +0377			

*1 L = 1 if this is the last entry. The index entry length in words is $\frac{(\text{KEYLENGTH} - 1)}{3} + 2$

The key value is that of the last (highest) record in the sector.

Note that records cannot be inserted into an index beyond the highest key.

OVERLAY OBJECT CODE

	Word		
	0	N	where N is binary value of the overlay number eg, overlay 17 would begin with 00000021
Sector 1	1	Object code	
	255		

The 1 to 5 sectors are contiguous

	Word	
	0	
Sectors 2 to 5 (if needed)	1	Object code
	255	

riority Overlay Requests

request for a priority overlay is indicated by setting a single-word switch non-zero. There are three priority overlay request words:

FMOREQ	when non-zero, indicates a request for overlay 21 entry 1 to set up a system flashing message. FMOREQ is set by Level 4 when a system message number is found in ECODE of a terminal's User Table.
PDOREQ	when non-zero, indicates a request for overlay 17 entry 1 to perform PAGDIR reorganization and Master and Maintenance Sector re-writes. PDOREQ is set by Level 4 once every minute.
:REQ8L	when non-zero, it is a pointer to a list of overlay requests which are not associated with any terminal. This is used by the Communications Submonitor to request its overlays and by the Allocation Submonitor to request the overlay to do deallocation.

erminal Overlay Requests

request for a terminal overlay is indicated by storing an overlay request word in REQ8 in the terminal's User Table and incrementing the resident word OVRREQ (count of terminal overlay reqests). The format of an overlay request word is:

Q		X		M	Overlay #	entry #
1	3	1	2	1	8	8

Q: 0 = queue keystrokes during overlay execution
1 = toss keystrokes during overlay execution
X: 1 = special request
M: 0 = intercept "MODE" key
1 = pass "MODE" key as data key

Upon completion, the overlay returns to the Overlay Submonitor by branching to one of two exits:

EXOVL	Exit with no further overlay activity required for this terminal. The submonitor then decrements OVRREQ and zeroes bits 4-23 in REQ8 in the terminal's User Table.
NXOVL	Exit to another overlay. Caller places new overlay request word is REQ8 in the terminal's User Table. OVRREQ is not decremented by the submonitor. The new overlay request will be honored after all other User Tables are scanned.

o WRITING OVERLAYS

```
          DCN  39
          BRA  ENTRY1
          BRA  ENTRY2
           o
          BRA  ENTRYn
ENTRY1    EQU  $
           o
          LD1  USER8C
* THE FOLLOWING CODE WILL SET UP A REQUEST
* FOR THE SECOND ENTRY POINT OF THIS OVERLAY
          LDA  OV39E2
          STA  REQ8,X1
          BRA  NXOVL
ENTRY2    EQU  $
           o
           o
          BRA  EXOVL
OV39E2    DCN  39*256+2
```

LOADOV Procedure

```
// LOADOV
/I=R80-38
/I=R80-0B
/I=relocatable program
/O=name
/T=0100000
//
// MAKESY
/I=name,O=overlay number
//
```

o WRITING TRANSFER PROGRAMS

```
          BSS       02400
START     DCN       0
           .
           .
          BRM       WAIT8
          SKZ       TRSTOP
          BRA       ABORT
           .
           .
NORMAL    INR       START
          BRR       START
           .
           .
ABORT     BSS       0
* DISPLAY ERROR MESSAGE
          BRR       START
BUFFER    BSS       256
          DCN       0
```

Note: To use the IDOS routines SYSIN and SYSOUT from a transfer program, you must include the following equates in your program.

```
SYSIN  EQU   01737
SYSOUT EQU   01740
```

LOADOV Procedure

```
// LOADOV
/I=R80-38
/I=R80-0B
/I=relocatable program
/O=name
/T=0100000
//
```

Debug is a dynamic display of system activity. Normally, the Debug display is kept in a System Block in memory; however, it may be configured to display on a screen. To do so, answer no to the CONFIG question "IS THIS A PRODUCTION SYSTEM?" and then enter the terminal number plus one of the screen to contain the Debug display (i.e., for terminal 0 enter 1, for terminal 3 enter 4, etc.) Only terminals 0 through 6 may be configured for the Debug display. (NOTE: to configure a screen as a DEBUG screen, DEBUG must have been OPTIONed for.)

NOTES: Show Mode G will give a snapshot display of the debug information.

Show Mode Z will give a dynamic display of the DEBUG information.

DEBUG DISPLAY FORMAT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	:SS SECOND COUNT	EOR WHEN EOR ACTIVE *	IOx IOB FAILURE COUNT **	-4- WHEN LEVEL 4 ACTIVE *	-I- WHEN INDSET LOOKUP ACTIVE *	xxx NAME OF CURRENT LEVEL 8 SUBMON. ***	-7- WHEN LEVEL 7 ACTIVE *	OVL WHEN OVERLAY ACTIVE *	xx NAME OF OVERLAY IN RAM (OCTAL)	TRA WHEN TRANSFER PROGRAM ACTIVE *	xxx xxx NAME OF TRANSFER PROGRAM IN RAM		xx NUMBER OF SYSTEM BLOCKS IN PAGDIR	xx NUMBER OF ENTRIES IN FVSRAM	xx NUMBER OF JOBS IN JOBRAM	xx NUMBER OF ENTRIES IN IXRRAM
2	QUE QUEUE LENGTHS →	xx NUMBER OF TERMS IN EOR	xx NUMBER OF SOFT LOST KEYSTROKE	xx NUMBER OF TERMS IN VALUE SET LOOKUP	xx NUMBER OF TERMS IN INDSET LOOKUP	xx NUMBER OF TERMS IN PAGE FETCH	xx NUMBER OF KEYSTROKES QUEUED	xx NUMBER OF OVERLAY REQUESTS QUEUED	xx NUMBER OF DISC REQUESTS QUEUED	LKD WHEN DISC I/O LOCKED	xxx CURRENT OR LAST DISC USER +	Ex COUNT OF DISC ERRORS **	xx DISC OPERATION AT LAST ERROR ++	xx xxx xx DISC STATUS AT LAST I/O ERROR		
3	TOO TERMINAL NUMBERS →	nn T= LAST EOR	nn T= LAST LOST KEYSTROKE	nn T= LAST VALUE SET LOOKUP	nn T= LAST INDEX SET LOOKUP	nn T= LAST PAGE FETCH	nn T= LAST KEYSTROKE PROCESSED	nn T= LAST OVERLAY LOAD	xx CURRENT OR LAST DISC OPERATION ++	D SSS SSS LOGICAL DRIVE AND SECTOR OF LAST DISC I/O			-T- RESERVED FOR TAPE	xxx xxx xxx RESERVED FOR TAPE ERROR STATUS		
4	xx xxx RP AT LAST LEVEL 4 INTERRUPT		xx NUMBER OF HARD LOST KEYSTROKES	xxx NAME OF LAST VALUE SET USED	xxx NAME OF LAST INDEX SET USED	xxx NAME OF LAST FORMAT PAGED	x LAST KEYSTROKE PROCESSED	RESERVED						RESERVED FOR COMMUNICATIONS SUBSYSTEMS		

NOTES

* BLANK WHEN INACTIVE.

** BLANK WHEN ZERO.

*** OVL – OVERLAY SUBMONITOR
TRA – TRANSFER PROGRAM SUBMONITOR
PRT – PRINT SUBMONITOR
ALO – ALLOCATION/DEALLOCATION SUBMONITOR
DYN – DYNAMO SUBMONITOR
COM – COMMUNICATIONS SUBMONITOR

\+ TERMINAL NUMBER IF LEVEL 7
SUBMONITOR NAME IF LEVEL 8

++ RD – STANDARD READ
WR – STANDARD WRITE
NR – NONSTANDARD READ
NW – NONSTANDARD WRITE
OV – OVERLAY LOAD

DYNAMO

DYNAMO is a dynamic memory display, patch, and trace routine that must be OPTIONed for. The DYNAMO display occupies two lines and appears at the DEBU screen in the portion of the screen normally used for the message and status lines.

DYNAMO Operations (Runs @ Level 3)

Bottom two lines of DEBUG screen:

Modes

Display Change Trace Patch ASCII DISP	DIS CHG TRC PAT ASC	XXX XXXXX location	XXXXXXXX contents of location optional 36 ASCII Characters	XXXXXXXX +1	XXXXXXXX +2	XXXXXXXX +3
Go Trace Stop Trace CCE Condition Codes	GO STP Gcc Scc	XXXXXXXX Register RA Before execution of intercepted instruction.	XXXXXXXX RB	XXXXXXXX X1	XXXXXXXX X2	XXXXXXXX X3

Key Functions

Home	Start CHG, cursor appears in location field, key in 5-digit location.
← →	Move cursor left/right one column.
0 7	Octal digits entered on screen, cursor advances.
Index F5	Start DIS, use location as address, refresh display.
← →	Location is decremented/incremented by 1, refresh display.
Valid F4	Display indirect, contents of location becomes new location.
? F3	Display restore, location before last indirect is restored.
ASD F1	ASCII display, 12 words at location moved to screen.
Prog1 F6 ↓ ↓ Prog 4 F9	At location, patch X words (PROGX) using the octal words on first line of screen.
Prog 5 F10	Activate GO trace at location; count set=000.
Prog 6 F11	Activate STOP trace, as above; also single shot execution.
↓	Replant trace at location +1.
↑	Replant trace at address of word at location.
CORR RESET TAB	Terminates trace mode.
Prog F2	Display disc. Key IDOS sector address.
↓	To display lower address in sector.
↑	To display higher address in sector.
Z^s	Rewrite changed sector.
All Other Keys	In CHG mode, ignored; otherwise, refresh display.

To Start DYNAMO

Press the unshifted HOME key to start DYNAMO at the Debug screen. The message line should have a solid block cursor at the left end of the five position location field. The left most 3 places on the line should have "CHG" indicating change mode.

MESSAGE LINE:

```
XXXcccllll1 aaaaaaaa bbbbbbbb cccccccc dddddddd
```

where:

```
XXX is CHG - change mode, enter a location into the lllll field.
       DIS - octal display mode, the contents of lllll are displayed at
             aaaaaaaa, the contents of lllll+1 are displayed at bbbbbbbb,
             etc.
       ASC - ASCII display mode, aaaaaaaa etc is ASCII representation of
             location lllll.  36 bytes are displayed.
       PAT - patch has been made beginning at location lllll.
       TRC - a trace or stop has been planted at location lllll.
```

ccc is count of the number of times the instruction at location lllll has been executed in trace or stop mode.

lllll is the current location in RAM being displayed, patched or traced.

STATUS LINE: contents depends on function being performed.

To Display RAM:

Press the HOME key.
Key the 5 octal digit RAM location.
Press the INDEX (F5) key.

To display in ASCII.
PRESS ASD (F1) key.
36 bytes (12 words) beginning at location lllll are displayed.

To Display MOD II Information

Press the HOME key.
Key the address WWWPP; where WWW = window (8 bits)
                              PP = page (0-31)
Press the MODE key.

Display appears as:

1. Mapper RAM
2. Window Register
3. Memory Parity Register
4. Mapper Parity Register

To STOP Execution At A Specified Location:

The TRC mode is used to stop execution at a specified location.

The stop occurs before the instruction is executed. So you can check the condition codes before a branch, etc.

Press HOME key.
Key the 5 octal digit location.
Press PROG6 (F11) key.

When the stop occurs, the status line will show the contents of registers RA, RB, X1, X2 and X3 in octal from left to right. At the far left end of the status line the 3 byte display Scc will display. cc is the octal condition code at the time of the stop. The count ccc on the message lin should show 001 the first time.

To cancel the stop and allow the system to continue press RESET (TAB).

To release the stop and catch it again the next time press PROG6.

NOTE: This may be used to stop trace at all levels above level 3.

To PATCH RAM (4 or fewer consecutive words)

Press HOME key.
Key the 5 octal digit lowest location to be patched.
Press the INDEX (F5) key.

The four locations beginning at lllll will display.
Verify that you are at the correct spot in RAM.

Press HOME key.

Press the RIGHT arrow to position the solid cursor to the octal digits to be changed and key the correct octal digits. The actual changes are not made to RAM until the next step is performed.

Press PROG1(F6), PROG2(F7), etc. to patch 1, 2, etc. words.

PAT will display at the left most end of the message line.

DUMP PROCEDURE

1. AUTO to MANUAL.
2. RESET then STEP.
3. LOAD into TIR: 71100001
4. MANUAL to AUTO then to MANUAL

   a. Tape Dump - (9 Track only)
      1. Mount a scratch tape
      2. MANUAL to AUTO
         The system will dump records 1024 words long (06000 bytes)

   b. Disc Dump
      1. MANUAL to AUTO

5. When the dump is complete, the system will halt with X3=00000000. To get a formatted dump:

```
// DUMPV2
/INPUT=(T8 or T16 or D)
//
```

DUMPV2 requires a 0200 sector contiguous file called DUMP47. This can be created by running the program MAKD47.

```
// MAKD47
/O=drivetype                (where drivetype = 8230, 8240, 8260 or 8270)
/Banks=n.                   (where n = 1 for 96k systems or smaller
                                      = 2 for 192k Mod II systems   )
//
```

6. DUMPV2 exits to A80-DB to print the dump. If an unformatted dump is wanted, or only part of a dump, or if the directories are clobbered causing A80-DB to loop:

   a. Set switch zero up.
   b. LOAD memory address where dump should start into TIR.
   c. // A80-DB.

7. If low memory is clobbered and the 71100001 will not execute a dump, the address of the dump routine is also in the Format Vector Table. Try: (in step 3 above)

   LOAD into TIR
   71101767

Performance Monitor

DATA IV must be OPTIONed for the performance monitor in order to use it to monitor system activity. In addition, the transfer program A80PMR must be copied onto the customer pack in order print the performance report.

To start the Performance Monitor

```
$SETPP  STARTPM
$SETPP  STARTPM (OVLOG)
$SETPP  STARTPM (KEYSTROKE=05)*
$SETPP  STARTPM (RPLOG, OVLOG, KEYSTROKE=05)*
```

* Note: The keystroke to be monitored must be a control key and is entered modulo 0200. For example, the MODE key is 0205; to monitor it, the code to enter in the $SETPP command is 05.

To stop the Performance Monitor

```
$SETPP  STOPPM
$SETPP  STOPPM (NOREPORT)
$SETPP  STOPPM (REPORT=MASTER133,1)
$SETPP  STOPPM (TITLE="ABC", REPORT=X,1)
```

Performance Parameters

LEV78B -	Number of terminals that PROKS will attempt to process before returning CPU to submonitors. (LEV78B = N, 0<N≤32)
NKSMIN -	Maximum number of keystrokes to process out of a terminal's queue at one time. (NKSMIN = N, 0<N<5)
DYNBAL -	Causes system to perform dynamic adjustment to the system's level78 balancing. (DYNBAL = N, 0<N<8, or DYNBAL = NO)
PAGLIF -	Sets the age at which an unused page will be discarded by the garbage collector. (PAGLIF = N, 0<N<8 where the "life expectency" is 2 to the Nth, or PAGLIF = NO)
PAGSEC -	Causes the system to read into memory more than just the first page in a sector of format object code. (PAGSEC = N, 0≤N≤3 where N is the number of seconds to advance the "time" word in the page directory entry.)
BUFBLK -	Enables the system to break down N sector buffers into system blocks. (BUFBLK = N, where N can't be greater than 1/4 the number of sector buffers.)
COMWT8 -	Gives priority to the Batch Communications. (COMWT8 = N, 1≤N≤99)

Note: Show mode K will display the current parameter settings.

DATA IV SYSGEN UTILITY PROGRAMS

This section contains descriptions of the special utilities used by DATA IV and VISION during the sysgen procedures.

NAME: AAQ

PURPOSE: Insure that desired output file was created. Usually used immediately after an assembly or load step.

JCL:

```
// AAQ
/I=filename@drive #
//
```

STEPS: AAQ searches the IDOS directory on the specified drive and checks to see that the last entry in the directory is the specified file; this indicates that the file was the last one created on that pack.

MESSAGES: If the filename is not the last name in the IDOS directory, AAQ assumes the reason the file was not created was due to lack of contiguous space and therefore displays the following message on screen 0:

```
filename NOT CREATED!  MAX # OF CONTIGUOUS SECTORS=0XXXXX  "AAQ"
```

This message can be misleading since lack of contiguous space is not the only reason why a file wouldn't be created (eg, an input to a load step was misspelled, there are unresolved virtuals in a LOADOV step, etc.)

EXAMPLE: The following JCL will check to see that the file DATA was created in the previous step.

```
// AAQ
/I=DATA
//
```

NAME: CCFGEN

PURPOSE: To create a control file that contains the JCL statements to delete and/or copy files from one drive to another. For example, the AI14 COPYS control file uses CCFGEN to create the JCL to copy all non-optional DATA IV software from the absolute master onto the customer pack.

JCL:

```
// CCFGEN
/S=source           Drive # containing files to copy.
/D=destination      Drive # from which files will be deleted and
                    to which files will be copied.
/U=usefile          Drive # containing files named on parameter
                    cards (format 2).
/O=filename         Name of file to contain the generated JCL
                    statements.
     .
     .
parameter cards
     .
     .
//
```

PARAMETERS: The parameter cards may have one of three forms (all forms may be intermixed in the same jobstream.)

Format 1:	cols 1 - 6	Name of file to copy from Source disc
Format 2:	cols 1 - 11	.USE filename, where "filename" is the name of a file containing Format 1-type statements; the first record in the file must contain .USEFILE in columns 1-8.
Format 3:	cols 1 - 6	Name of file to copy from Source disc
	cols 8 - 13	Name of file on Destination disc if different from name of file on Source disc

NOTE: Since CCFGEN only creates the JCL, you must execute the control file created in order to actually perform the deletes and copies.

EXAMPLE: The following JCL will copy file A from drive zero to drive one; it will also copy all files named in file USEIT from drive 0 to drive 1.

```
// CCFGEN
/S=0,D=1,O=X,U=0
A
.USE USEIT
//
// X
```

where the contents of USEIT is:

```
.USEFILE
B
C
```

MEMORY LAYOUT

72-K		96-K
0	Interrupt Locations	0
060	Screens, User Tables, Disc Request Queues, Sector Buffers, System Blocks, etc.*	060
	Zone Pointer Table	
	Overlay Directory	
$DATA	Optional Resident	$DATA
	Fixed Resident	
@FV	Format Vector Table	@FV
047200	IOID5 (8121 Print Output)**	067200
047300	PUSER (Keyboard Input)	067300
047400	Overlay Area	067400
052000	Transfer Program Area or 12 Sector buffers for level 7	072000
057777		077777

*All memory not allocated to screens, user tables or executable code is organized into the following:

Sector Buffers - 0400 (256) words
number set at CONFIG time

System Blocks - 0100 (64) words

Mini Blocks - 020 (16) words

Micro Blocks - 04 (04) words

**If there are no 8121 printers, this area will be used for system blocks.

USER TABLE SKELETON DEFINITIONS

For non-MFE systems, the first seven words of the user table is a short routin that is executed by level 3. The routine calls IKEY to input a keystroke and then places it in the user table's keystroke queue. For MFE systems, MFE passes the keystroke to the DATA IV level 3 routine which places that keystroke in the keystroke queue for the appropriate terminal.

The keystroke queue consists of single word entries; the first byte of each entry is the (unconverted) keystroke that was entered and the remaining two bytes are the address of the next word in the queue for that user table.

```
USERN  BSS     1        Entry to KEYS routine.
USERN1 BRM     IKEY     Get key input.  (Non-MFE systems, only.)
USERN2 STB*    KQBOT    Add keystroke to
USERN3 STB     KQBOT      bottom of the keystroke queue.
USERN4 LDB     LV3SVB   Restore RB.
USERN5 INR     KQLEN    Increment keystrokes in queue.
USERN6 BRD     USERN    Return and debreak.
*
KQLEN  EQU     7        Number of keystrokes in the queue.
KQTOP  EQU     010      Top of keystroke queue; always contains one dummy entry.
KQBOT  EQU     011      Bottom of keystroke queue.
KQUP   EQU     012      An "UP  KQTOP" instruction to get a keystroke.
*
SCR    EQU     013      Execute to place character on screen
FWA    EQU     014      First word of screen
LWA    EQU     015      Last word address + 1 of data portion of screen
MFWA   EQU     LWA      First word address of message line
SLWA   EQU     LWA      Last word + 1 of screen
PSROT  EQU     016      Character location of cursor (PZE SROT)
DEST   EQU     017      Location on screen of cursor
MODE   EQU     020      24 flags (see below)
OPPTR  EQU     021      Sector address of OPERATORS entry
JOBDES EQU     022      Pointer to JOBRAM entry
FORPTR EQU     023      Pointer to PAGDIR entry
FSROT  EQU     024      Character location of start of current field (PZE SROT)
FDEST  EQU     025      1st word of current field
REQ8   EQU     026      Overlay request word.
RECLOC EQU     027      Relative loc of current record in sector
WIDTH  EQU     030      -(Width of field)
MODPTR EQU     031      Pointer to in-front modifier.
LCR    EQU     032      Execute to get character from existing record
ACCP1  EQU     033      Pointer to first 4 accumulators. 0-3.
ACCP2  EQU     034      4-7
ACCP3  EQU     035      8-11
ACCP4  EQU     036      12-15
ACCP5  EQU     037      16-19
ACCP6  EQU     040      20-23
NFOR   EQU     041      Number of formats
NVAL   EQU     042      Number of value sets
SIZE   EQU     043      Number of words in data area of screen
MSIZE  EQU     044      - Size of screen in words
MSIZEB EQU     045      - Size of screen in bytes
KBTYPE EQU     046      Keyboard descriptor word from config
LASTKS EQU     047      Last three keystrokes processed
TVUSER EQU     050      Trail verify user pointer
TERMN  EQU     051      Terminal index 0-31, terminal printer number
BATCHN EQU     052      ASCII batch ID - 6 characters - even boundary
BACHN1 EQU     053      continued
BINDOC EQU     054      Binary document No.
TERMAS EQU     055      Terminal number in ASCII right adjusted, blank filled
BATCH  EQU     056      Sector address of batch dir entry or batch index entry
LINKBS EQU     057      Return address from LINKVS.
BINREC EQU     060      Binary record number
MAXREC EQU     061      Largest binary record number in the batch
NEXTUR EQU     062      Unconditional keystroke return address
STATS  EQU     063      Operator statistics table
STATS1 EQU     064      2nd word of op. stats. table.
STATS2 EQU     065      3rd word of op. stats. table.
BINCOL EQU     066      Binary column counter.
ALARM  EQU     067      Audible alarm instruction (IO KBTYPE or NOP)
```

USER TABLE SKELETON DEFINITIONS- CONTINUED

```
SAVPTR EQU    070      Save pointer for backspacing.
MODEXT EQU    071      Mode extension flag word.
DISCRA EQU    072      Return address when disc I/O complete (level 7)
VRECB  EQU    073      Non-zero when in verify mode; points to verify rec buffe
RECHDR EQU    074      Current record header word
RECSIZ EQU    075      Binary record size
SECTOR EQU    076      Current relative sector address
NEXTCR EQU    077      Return address from BRM NEXTC
COLUMN EQU    0100     Pointer to column count on screen (FWA of status line)
 CODE  EQU    0101     Error code.  Sign = Flasher
FX2    EQU    0102     X2=Location in format (via BAL)
FX3    EQU    0103     X3=Negative field width remaining to be keyed
VLROT  EQU    0104     LCR pointer for verify
VDEST  EQU    0105
       PZE    0        0106, lost keystroke routine from IOID+040
       BRM    KEYL     0107
       BRD    $-2      0110
QUE8RA EQU    0111     Return address from QUE8 call.
WPA    EQU    0112     Scratch word pair
 PA1   EQU    WPA+1
WPB    EQU    0114     Scratch word pair
WPB1   EQU    WPB+1
WPC    EQU    0116     Scratch word pair
WPC1   EQU    WPC+1
WPD    EQU    0120     Scratch word pair
WPD1   EQU    WPD+1
DQSLOT EQU    0122     Current position in disc queue.
OV36BL EQU    0123     Multi mode block address.
FMTST1 EQU    0124     1st 24 bits of format backspacing stack.
FMTST2 EQU    0125     2nd 24 bits of format backspacing stack.
MAXDOC EQU    0126     Largest binary document number
MODE3  EQU    0127     24 Flags
LASSEC EQU    0130     Last sector address of current batch
SSACOL EQU    0131     Saved column number for return to keying position
RETCOL EQU    0132     Column number for return to last position
RETNXC EQU    0133     Saved 'NEXTCR'
RS     EQU    0134     Digits to right of decimal point, LZERO field
SCPBAD EQU    0135     Used by screenprint.
KV     EQU    0136     Used by key verify.  byte 0 is scrambled char
MODE2  EQU    0137     More single bit flags, like 'MODE'
NEMODE EQU    0140     Next mode selection, byte 0
STATS3 EQU    0141     4th word of op. stats. table
SNDINQ EQU    0142     2770 - used for send inquiry
RCVINQ EQU    0143     2770 - used to receive inquiry response
RCVSPL EQU    0144     2770 - used for mini block address - spool info
REMOTE EQU    0145     Address of Control Block for Remote Terminal
WPE    EQU    0146     Extra word scratch pair
 PE1   EQU    0147           Used by P1CVR for re-entrancy
WPF    EQU    0150     Extra word for S80-5C
WPF1   EQU    0151     Extra word for S80-5C
WINDOW EQU    0152     Window Number (See Diagram Below)
BLOCK7 EQU    0153     Level 7 Block Word (See Description Below)
L7SVX2 EQU    0154     Level 7 Save of X2
L7SVX3 EQU    0155     Level 7 Save of X3
FMADR  EQU    0156     Pointer to Flashing Message Block.
CURC   EQU    0157     Counts Level4 Until Cursor Change Needed.
CUR    EQU    0160     Off Screen Copy of Blinking Word.
```

ords 0161 - 0163 are not used.

USER TABLE NOTES ARE ON THE FOLLOWING PAGE

USER TABLE NOTES

BLOCK7 ≠ 0, PROKS blocked at terminal; = 0, PROKS not blocked

bit	contents
0	1=terminal ready to be restarted
1	used by $DRIVE
9-23	restart address

WINDOW: terminal's window number.

bit	contents
0	not used
1-8	window number
9-13	logical page number
14	odd parity
15	1=read only
16-18	Bank number
19-23	physical page

TERMN: terminal printer information.

bit	contents
0	1=terminal currently printing; 0=not printing
1-2	not used
3-4	reserved
5-11	not used
12	1=screen print not allowed at this terminal
13	1=terminal printer is assigned
14-18	Diablo printer unit # assigned to terminal
19-23	binary terminal number

OVERLAY DIRECTORY

@OVDIR	Points to the word preceding the overlay directory in RAM (since the RAM overlay directory is preceded by a pair of words containing the release and fix level, @OVDIR points to the second of these words)
M#OVLY	The largest overlay number defined

-1	ASCII RELEASE				
0	and FIX level				
1	L	E		number of sector -1	Rel addr of ovly 1
2					Rel addr of ovly 2
					.
					.
					.
					.
99					Rel addr of ovly 99
	1	1	7	3	12

L = 1, Overlay is loaded (in DAT000)
E = 1, Overlay exists

OVERLAY REQUESTS QUEUES

There are two types of overlay requests. The first consists of those that are required by terminals (eg, Overlay 2 is used for Show Mode processing) and the second consists of those not associated with terminals (eg, Overlay 17, Entry Point 1 which is requested by Level 4 to free page directory blocks.) These two general categories of overlay requests are handled in different ways.

Terminal Overlay Requests

OVRREQ	Number of outstanding terminal overlay requests.
REQ8,X1	Terminal overlay request word.

The format of a terminal overlay request word is:

Q		X		M	Overlay #	entry #
1	3	1	2	1	8	8

Q: 0 = queue keystrokes during overlay execution
1 = toss keystrokes during overlay execution
X: 1 = special request
M: 0 = intercept "MODE" key
1 = pass "MODE" key as data key

Special Overlay Request Queue

:GETOV	Address of top of special overlay request queue. (Next special overlay to be processed.)
:PUTOV	Address of bottom of queue.
:FREOV	Address of free stack for special overlay requests.

Pointer to next queue entry
Ovly # * 256 + Entry Point #

Note: The special overlay request queue always has a dummy entry.

SCB's are pointed to by Control Block (CB) pointers which have the following format:

Pointer to next entry
Address of SCB

Only the CB's for those SCB's that are active or frequently active are part of a queue of CB's. As other SCB's become active, an associated CB pointer is added to the queue and as one becomes inactive, its CB is removed. The Submonitor Scan program (S80-8) scans the CB chain when it "looks" for submonitors to be serviced. (Initially the overlay, printer, and spool SCB's are the only ones active. Others, such as the transfer program submonitor, become active as required.)

The following are the resident words associated with the CB queue:

SCQT	CB that points to current SCB or last SCB processed.
SCQTOP	Top of CB queue.
:INCBL	Bottom of CB queue.

MASTER SECTOR

0	0				00003	Pointer to CONFIG
1	0				00000	Back Pointer
2	0				00002	Here Pointer
3	0				00000	Code Word
4	(t-86400) seconds since midnight = TIME					
5	ASCII day number (001-366)					
6	0				first sector addr of FVS dir 14	
7	0				first sector addr of JOB dir 14	
010	0				first sector addr of MAINT dir 14	OID sector
011	reserved					
012 013	ASCII release level name (eg. AI15-A)					
014 015 thru 0156	L 1	E 1	reserve 7	number secs-1 3	rel sec addr overlay 1 rel sec addr overlay 2 rel sec addr overlay 99	*1
0157	reserved					
0160 0161 thru 0177	0 0 . 0				zones 000 thru 017 zones 020 thru 037 ... zones 0360 thru 0377 16	*2
0200	reserved 8		no. zones - 1 8		largest zone no. 8	
0201	reser. 6				absolute sector address of zone 000 18	set by MAKESY
0202	address of format vector table					
0203 thru 0377	reserved					

*1 L=1, overlay is loaded
E=1, overlay exists

The rightmost 12 bits of each of these words contains the relative sector address (within zone 0) of the first sector of that overlay.

*2 A 16 word table maintained by ZONEMA. Bits 8 to 23 of each word represent 020 possible zones. One bits indicate defined zones. Zero bits indicate undefined zones.

CONFIG SECTOR

	Standard 4 word header	*1
+4	Time of last change (t-86400 seconds)	
+5	Day of last change	
+6	KB 1 AUD 1 -Reserved- (4) Size of screen in words /16 (7) FWA of screen /16 (11)	*2
+0046- +0105	-Reserved-	
+0106	MAXMFE 8 bits Number of disc buffers 5	*3
+0107	Reconfiguration password (scrambled) 0=no password	*4
+0110	Reserved	
+0111	Print/log control password (scrambled) 0=no password	*4
+0112	Supervisor password (scrambled) 0=no password	*4
+0113	Reserved for future (3 words)	
+0116	SYSFIG (CONFIG)	*5
+0117	SYSFG1 (CONFG1)	*5
+0120	HASP Communications Word (SYSFG2/CONFG2)	*5
+0121	2770 Communications Word (SYSFG3/CONFG3)	
+0122	Performance Parameters	*6
+0123	reserved	
+0124	reserved	
+0125	COMWT8 7	
+0126	3770 Configuration Word 1 (SYSFG6/CONFG6)	
+0127	3770 Configuration Word 2 (SYSFG7/CONFG7)	
+0130	3770 Configuration Word 3 (SYSFG8/CONFG8)	
+0131	Remote Terminal Configuration Word 1 TLIMIT (5)	
+0132	Remote Terminal Configuration Word 2	
+0133	COMWT8 7	
+0133- +0137	Reserved	
+0140- +0142	Jobname of current log file	
+0143- +0144	Batch # of current log file	
+0145	PRSTAT flags	*7
+0146- +0165	8121 Printer Specification Words (16 words)	*8
+0166- +0225	User Table Word TERMN (32 words)	
+0226- +0265	SPOOL PARAMETERS	
+0266- +0306	SPOOL Conversion Control Table (17 words)	
+0307- +0323	$COMM Controller Assignments Table (13 words)	*9
+0324- +0336	Reserved (11 word area)	
+0337- +0377	Reserved for user (33 word area)	

* Footnotes on next page

Footnotes for CONFIG Sector

*1 Forward pointer points to KEYS sector

*2 One word for each terminal
KB:0=Source Keyboard,1=029
AUD:0=no audible alarm

*3 MAXMFE = 0, for non-MFE systems; MAXMFE = maximum number of DATA terminals signed on under MFE
Minimum is 7

Recommended value is number of terminals + 2; additional buffers needed for communications, $EDIT, etc.

*4 Defaults for Passwords:
8 blanks - Reconfiguration or Master Password (020430614)
12345678 - Print/Log (052675507)
PASSWORD - Supervisor (070702334)

*5 System Configuration Words:
Word 1 (SYSFIG)

bit(s)	
0	0=48 char screen, 1=81
1	1=MOD II with extended memory; 0=no extended memory
2	0=no channel adaptor
3	0=no memory log
4	0=no card reader
5	0=no printer, 1=use IDOS printer
6	0=9-track tape, 1=7-track tape
7	0=low density tape, 1=high density tape
8	0=no shift override of field type, 1=allow shift override
9	0=no concurrent bisync, 1=concurrent bisync
10	0=count records, 1=count documents
11	0=one beep/error; 1=one beep/keystroke after error
12	0=EBCDIC, 1=ASCII output to tape
13	0=LAM/8436 1=LAM/8437,
14	0=MLAM/8436 1=MLAM/8437,
15-17	Type of dump: 0 = no RAM dump. 1 = tape dump. 2 = 8230 disc dump. 3 = 8240 disc dump. 4 = 8260 disc dump. 5 = 8270 disc dump.
18-19	number of tape decks (2 bits)
20-23	number of disc drives (0-11) (4 bits)

Word 2 (SYSFG1)

0	0=no 12 channel printer for received data (BISYNC)
1	0=extended error recovery (BISYNC), 1=no extended error recovery
2	0=do not count FIND mode statistics, 1=count FIND mode
3	0=DATA IV print spooling, 1=bisync direct print
4	0=field reverification, 1=no reverify option
5	0=do not keep detail operator statistics by batch, 1=do keep by batc
6	0=no decentralized batch transmission, 1=yes
7	0=unconditional decentralized batch transmission, 1=conditional
8-10	tape buffer size (number of sectors; maximum is 6)
11-13	3 bit Baud flag: 0 = 600 1 = 1200 2 = 2000 3 = 2400 4 = 4800
14-18	number of 8121 printers
19-23	number of terminals

Footnotes for CONFIG sector - continued

Word 3 (SYSFG2) HASP Workstation Communications

0	0=no HASP Workstation, 1=yes
1	0=no card reader, 1=yes
2	0=512 byte block message size, 1=400 bytes
3	0=manual answer, 1=auto answer
4-5	Compression: 00 = no compression 01 = trailing blank truncation 10 = blank compression 11 = full compression
6	0=do not send spanned records, 1=send spanned records
7	0=do not send transparent text, 1=send transparent text
8	0=no timeout, 1=20 second idle timeout
9	0=terminal-CPU, 1=terminal-terminal
10	0=primary station, 1=secondary station
11	not used
12-18	Console message size 1-120 bytes
19-23	Console number 0-31

*6

bits	contents
0-2	not used
3-6	BUFBLK
7-9	PAGSEC
10-12	PAGLIF
13-15	DYNBAL
16-17	NKSMIN
18	not used
19-23	LEV78B

*7 PRSTAT

bits	contents
0	reserved
1	0=no log file, 1=log file
2	0=no printer, 1=printer
3	1=system printer suspended by MODE LC,STOP
4	reserved
5	clear request
6-18	reserved
19	0=don't log C-type messages, 1=log C-type messages
20	0=don't log B-type messages, 1=log B-type messages
21	0=no screen print, 1=screen print turned on
22	0=don't print C-type messages, 1=print C-type messages
23	0=don't print B-type messages, 1=print B-type messages

*8 8121 Printer Specification Words: 1 word for each possible terminal terminal.

bit 0: 0=Elite, 1=Pica
byte 0: number of printer lines per page -1
byte 1: number of increments to advance each line
(1 increment = 1/48 inch)
byte 2: total number of lines per page

Footnotes for CONFIG sector - continued

*9 Controller Assignment Table (CAT): Each word corresponds to one controller with the first word corresponding to the 8436 controller, the second word to the 8437 #0 controller, the third to the 8437 #1 controller, and so on. Each word has the following format:

bits	contents
0	0=controller doesn't exist, 1=controller exists
1-7	valid application bits; 0=not valid, 1=valid bit 1 = remote terminal 2 = 3270 3 = HASP 4 = Bisync 5 = 3770 6 = not used 7 = not used
8-15	current user 0 = none 1 = remote terminal 2 = 3270 3 = HASP 4 = Bisync 5 = 3770
16-23	reserved

WRITING OVERLAYS

```
                DCN  39
                BRA  ENTRY1
                BRA  ENTRY2
                 o
                BRA  ENTRYn
      ENTRY1    EQU  $
                 o
                LD1  USER8C
      * THE FOLLOWING CODE WILL SET UP A REQUEST
      * FOR THE SECOND ENTRY POINT OF THIS OVERLAY
                LDA  OV39E2
                STA  REQ8,X1
                BRA  NXOVL
      ENTRY2    EQU  $
                 o
                 o
                BRA  EXOVL
      OV39E2    DCN  39*256+2
```

Upon completion, the overlay returns to the Overlay Submonitor by branching to one of two exits:

EXOVL	Exit with no further overlay activity required for this terminal. The submonitor then decrements OVRREQ and zeroes bits 4-23 in REQ8 in the terminal's User Table.
NXOVL	Exit to another overlay. Caller places new overlay request word is REQ8 in the terminal's User Table. OVRREQ is not decremented by the submonitor. The new overlay request will be honored after all other User Tables are scanned.

LOADER Procedure

```
// LOADER
/M NAME,VALUE;
/B U=OVERNT; /LOC B=OVERNB;
/O A80:NN; /C N; /I R80-0B,R80:NN;

/L OVLIB;
/L R80-38;  if optional resident not referenced
/L R80-RS;  if optional resident referenced
/L RDS-SY;  if optioned for MFE

//
```

NOTE: Routines loaded with R80-RS must be reloaded if the system is reoptioned.

o

WRITING TRANSFER PROGRAMS

```
                BSS       02400
        START   DCN       0
                 .
                 .
                BRM       WAIT8
                SKZ       TRSTOP
                BRA       ABORT
                 .
                 .
        NORMAL  EQU       $
                BRA       TREND    Normal Return
                 .
                 .
        ABORT   BSS       0
        * DISPLAY ERROR MESSAGE
                BRA       TRABT    Abnormal Return
        BUFFER  BSS       256
                DCN       0
```

LOADER Procedure

```
// LOADER
/M NAME,VALUE;
/B U=TOPMEM; /LOC B=OVERNB;
/O absolute-file; /C N; /I R80-OB,relocatable-file;

/L TRLIB;
/L R80-38;  if optional resident not referenced
/L R80-RS;  if optional resident referenced
/L RDS-SY;  if optioned for MFE

//
```

NOTE: Routines loaded with R80-RS must be reloaded if the system is reoptioned.

DEBUG

Debug is a dynamic display of system activity. Normally, the Debug display is kept in a System Block in memory; however, it may be displayed on a screen by entering Mode D.

NOTE: Show Mode G will give a snapshot display of the debug information.

DEBUG DISPLAY FORMAT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	:SS SECOND COUNT	EOR WHEN EOR ACTIVE *	IOx IOS FAILURE COUNT **	-4- WHEN LEVEL 4 ACTIVE *	-I- WHEN INDSET LOOKUP ACTIVE *	xxx NAME OF CURRENT LEVEL 8 SUBMON. ***	-7- WHEN LEVEL 7 ACTIVE *	OVL WHEN OVERLAY ACTIVE *	xx NAME OF OVERLAY IN RAM (OCTAL)	TRA WHEN TRANSFER PROGRAM ACTIVE *	xxx xxx NAME OF TRANSFER PROGRAM IN RAM		xx NUMBER OF SYSTEM BLOCKS IN PAGDIR	xx NUMBER OF ENTRIES IN FVSRAM	xx NUMBER OF JOBS IN JOBRAM	xx NUMBER OF ENTRIES IN IXRRAM
2	QUE QUEUE LENGTHS →	xx NUMBER OF TERMS IN EOR	xx NUMBER OF SOFT LOST KEYSTROKE	xx NUMBER OF TERMS IN VALUE SET LOOKUP	xx NUMBER OF TERMS IN INDSET LOOKUP	xx NUMBER OF TERMS IN PAGE FETCH	xx NUMBER OF KEYSTROKES QUEUED	xx NUMBER OF OVERLAY REQUESTS QUEUED	xx NUMBER OF DISC REQUESTS QUEUED	LKD WHEN DISC I/O LOCKED	xxx CURRENT OR LAST DISC USER +	Ex COUNT OF DISC ERRORS **	xx DISC OPERATION AT LAST ERROR ++	xx xxx xx DISC STATUS AT LAST I/O ERROR		
3	TOO TERMINAL NUMBERS →	nn T= LAST EOR	nn T= LAST LOST KEYSTROKE	nn T= LAST VALUE SET LOOKUP	nn T= LAST INDEX SET LOOKUP	nn T= LAST PAGE FETCH	nn T= LAST KEYSTROKE PROCESSED	nn T= LAST OVERLAY LOAD	xx CURRENT OR LAST DISC OPERATION ++	D SSS SSS LOGICAL DRIVE AND SECTOR OF LAST DISC I/O			-T- RESERVED FOR TAPE	xxx xxx xxx RESERVED FOR TAPE ERROR STATUS		
4	xx xxx RP AT LAST LEVEL 4 INTERRUPT		xx NUMBER OF HARD LOST KEYSTROKES	xxx NAME OF LAST VALUE SET USED	xxx NAME OF LAST INDEX SET USED	xxx NAME OF LAST FORMAT PAGED	x LAST KEYSTROKE PROCESSED	RESERVED					RESERVED FOR COMMUNICATIONS SUBSYSTEMS			

NOTES

* BLANK WHEN INACTIVE.
** BLANK WHEN ZERO.

*** OVL – OVERLAY SUBMONITOR
TRA – TRANSFER PROGRAM SUBMONITOR
PRT – PRINT SUBMONITOR
ALO – ALLOCATION/DEALLOCATION SUBMONITOR
DYN – DYNAMO SUBMONITOR
COM – COMMUNICATIONS SUBMONITOR
BLANK – NO SUBMONITOR ACTIVE

\+ TERMINAL NUMBER IF LEVEL 7
SUBMONITOR NAME IF LEVEL 8

++ RD – STANDARD READ
WR – STANDARD WRITE
NR – NONSTANDARD READ
NW – NONSTANDARD WRITE
OV – OVERLAY LOAD

DYNAMO

DYNAMO is a dynamic memory display, patch, and trace program that may be used at the DEBUG screen.

DYNAMO Operations (Runs @ Level 3)

Bottom two lines of DEBUG screen:

Modes

Display Change Trace Patch ASCII DISP	DIS CHG TRC PAT ASC	XXX XXXXX location	XXXXXXXX contents of location optional 36	XXXXXXXX +1 ASCII Characters	XXXXXXXX +2	XXXXXXXX +3
Go Trace Stop Trace CCE Condition Codes	GO STP Gcc Scc	XXXXXXXX Register RA Before execution of intercepted instruction.	XXXXXXXX RB	XXXXXXXX X1	XXXXXXXX X2	XXXXXXXX X3

Key Functions

Home		Start CHG, cursor appears in location field, key in 5-digit location.
←	→	Move cursor left/right one column.
0	7	Octal digits entered on screen, cursor advances.
Index	F5	Start DIS, use location as address, refresh display.
←	→	Location is decremented/incremented by 1, refresh display.
Valid	F4	Display indirect, contents of location becomes new location.
?	F3	Display restore, location before last indirect is restored.
ASD	F1	ASCII display, 12 words at location moved to screen.
Prog1 ↓ Prog 4	F6 ↓ F9	At location, patch X words (PROGX) using the octal words on first line of screen.
Prog 5	F10	Activate GO trace at location; count set=000.
Prog 6	F11	Activate STOP trace, as above; also single shot execution.
↓		Replant trace at location +1.
↑		Replant trace at address of word at location.
CORR RESET	TAB	Terminates trace mode.
Prog	F2	Display disc. Key IDOS sector address.
	↓	To display lower address in sector.
	↑	To display higher address in sector.
	Z^s	Rewrite changed sector.
All Other Keys		In CHG mode, ignored; otherwise, refresh display.

RELEASE AI15

To Start DYNAMO

After entering Mode D, enter the password FOURFAZE. The DYNAMO display will appear on the bottom of the screen above the message line. The remainder of the screen will depend on the screen size:

6 lines:	The DYNAMO display will appear by itself.
12 lines:	An abbreviated form of the instructions for using DYNAMO will appear above the DYNAMO line.
24 lines:	The abbreviated form of the instructions will appear above the DYNAMO line and the DEBUG display will remain on the screen.

The DYNAMO line has the following format:

```
XXXccclllll aaaaaaa bbbbbbb ccccccc ddddddd
```

where:

XXX is CHG - change mode, enter a location into the lllll field.
DIS - octal display mode, the contents of lllll are displayed at aaaaaaa, the contents of lllll+1 are displayed at bbbbbbb, etc.
ASC - ASCII display mode, aaaaaaa etc is ASCII representation of location lllll. 36 bytes are displayed.
PAT - patch has been made beginning at location lllll.
TRC - a trace or stop has been planted at location lllll.

ccc is count of the number of times the instruction at location lllll has been executed in trace or stop mode.

lllll is the current location in RAM being displayed, patched or traced.

The contents of the line immediately below the DYNAMO line depends on function being performed.

To Display RAM:

Press the HOME key.
Key the 5 octal digit RAM location.
Press the INDEX (F5) key.

To display in ASCII.
PRESS ASD (F1) key.
36 bytes (12 words) beginning at location lllll are displayed.

To Display MOD II Information

Press the HOME key.
Key the address WWWPP; where WWW = window (8 bits)
PP = page (0-31)
Press the PROG CTL (F2) key.

Display appears as:

1. Mapper RAM
2. Window Register
3. Memory Parity Register
4. Mapper Parity Register

To STOP Execution At A Specified Location:

The stop occurs before the instruction is executed; therefore, you can check the condition codes before a branch, etc.

Press HOME key.
Key the 5 octal digit location.
Press PROG6 (F11) key.

When the stop occurs, the status line will show the contents of registers RA, RB, X1, X2 and X3 in octal from left to right. At the far left end of the status line the 3 byte display Scc will display. cc is the octal condition code at the time of the stop. The count ccc on the message line should show 001 the first time.

To cancel the stop and allow the system to continue press RESET (TAB).

To release the stop and catch it again the next time press PROG6.

NOTE: This may be used to stop trace at all levels above level 3.

To TRACE Execution At A Specified Location:

The trace is a dynamic one which may be stopped at any time by pressing the PROG6 or HOME key.

Press HOME key.
Key the 5 octal digit location.
Press PROG5 (F10) key.

The display for the trace is the same as that for the stop.
To stop the trace press PROG6 or HOME.

To release the stop and catch it again the next time press PROG5.

NOTE: This may be used to stop trace at all levels above level 3.

To PATCH RAM (4 or fewer consecutive words)

Press HOME key.
Key the 5 octal digit lowest location to be patched.
Press the INDEX (F5) key.

The four locations beginning at lllll will display.
Verify that you are at the correct spot in RAM.

Press HOME key.

Press the RIGHT arrow to position the solid cursor to the octal digits to be changed and key the correct octal digits. The actual changes are not made to RAM until the next step is performed.

Press PROG1(F6), PROG2(F7), etc. to patch 1, 2, etc. words.

PAT will display at the left most end of the message line.

DUMP PROCEDURE

1. AUTO to MANUAL.

2. RESET then STEP.

3. LOAD into TIR: 71100001

4. MANUAL to AUTO then to MANUAL

    a. Tape Dump - (9 Track only)
        1. Mount a scratch tape
        2. MANUAL to AUTO
        The system will dump records 1024 words long (06000 bytes)

    b. Disc Dump
        1. MANUAL to AUTO

5. When the dump is complete, the system will halt with X3=00000000.
   To get a formatted and a RAM dump key the following:

```
// DUMPD
   optional parameters
//
```

Where the default options are:

dump from disc file DUMP47@0
print analysis and RAM (0-077777)
96-K system

DUMPD requires a contiguous file called DUMP47. This can be created by running the program MAKD47.

```
// MAKD47
/O=8230 or 8240 or 8260 or 8270
/B=1 for 96K or smaller systems
   2 for 192K systems
//
```

The size of the DUMP47 file is dependent on the B parameter:

```
B = 1, DUMP47 is 0200 contiguous sectors
    2, DUMP47 is 0440 contiguous sectors
```

SINGLE WORDS IN RAM

These are counters, pointers, addresses, etc., used by DATA to keep track of itself and printed by DUMPD. A description of each word follows:

SYMBOL	DESCRIPTION
ACTO@	Address of the pointer to the Debug Display Area. This is present in a system block on production systems or on a screen on Debug systems.
ASCHR	Current system hour in ASCII, in the form bHH.
ASCMIN	Current system minutes in ASCII, in the form :MM.
ASCSEC	Current system seconds in ASCII, in the form :SS.
ASS16	Address of the last 16-word block assigned.
ASS64	Address of the last 64-word block assigned.
BATACT	Mask to inhibit the use of any active batch (see state request bits in the batch directory disc format).
BUFLST	Pointer to the start of the available sector/buffer list.
BLKCNT	Magnetic tape block count.
CHAR	Last PROKS character.
CHARV	The most recently typed in character in verify mode.
CLOCK	Fires level 4 every 1/5 of a second. Set to -12 at level 4 and incremented by level 0 every 1/60 second.
CLOCKS	One second clock.
CNFIG	Disc address of CONFIG sector.
COMMWD	Pointer to Communications Command word (HASP).
COMWT8	Value of COMWT8 performance parameter.
D	Zero = Debug display in system block; non-zero = screen number +1 of Mode-D screen
DAY	Day number, ASCII
$DBASE	Starting sector address for DATA.
DNTADR	Pointer to device entry table for 8121's.
DISCQ1	Pointer to FWA of primary disc request queue.
DISCQ2	Pointer to FWA of secondary disc request queue.
DSETIM	Time of last disc error (T - 86,400).
ECATB	Counter for all disc errors
EORREQ	Number of end-of-record requests outstanding.
EOVDAT	First data sector of $EOV tape label data.
F90M2?	0 = Not a MOD II, 1 = MOD II.
FETREQ	Number of page fetch requests outstanding.
@FV	New location of Format Vector Table.
FVLEN	Current length of the Format Vector Table.
FVSDIR	Disc address of FVS directory.
FVSRAM	Pointer to the Format/Value set directory in RAM.
HDRDAT	First data sector of $HDR tape label data.
HOUR	Binary system hour.
IO6CNT	IO6 chip failure count.
ISVER3	0 = no index sets, -1 = index sets
IXRRAM	Pointer to the Index Set directory in RAM.
JOBDIR	Disc address of job directory.
JOBRAM	Pointer to the job directory in RAM.
KBCINP	The last processed keystroke, before translation.
KBLOG	The total backlog count of keystrokes.
KFREE	Start of keystroke free cell list.
KFSIZE	Total number of keystoke cells in the system.
KFXTND	-(Number of words to extend KFREE); initially set to -16*NTERMS
LASTZN	Highest zone in the system.
LL	Screen line length in words - 020 or 040.
LL2	LL + LL.
LOCKPR	Printer lockout from magnetic tape.
LOGBAT	The sector address of the log file batch.
LOGJOB	Jobname/batch number of the log file.
....+1	Jobname/batch number of the log file.
....+2	Jobname/batch number of the log file.
....+3	Jobname/batch number of the log file.
....+4	Jobname/batch number of the log file.
LOGSEC	Current log sector.
MAGTPE	Count of magnetic tape errors.
MINUTE	Current time in minutes.
M#OVLY	Highest numbered overlay
N	Count of the number of screens.
NDS	Number of disc drives.
NEX4	Address of the next 4-word block available.
NEX8	Address of the next 8-word block available.
NEX16	Address of the next 16-word block available.

SINGLE WORDS IN RAM - CONTINUED

NEX64	Address of the next 64-word block available.
NLGSEC	Count of the sectors in the log file.
NPRSEC	Count of the sectors in the print queue.
OHTOP	Pointer to BAM OCB header chain.
OIDDIR	Disc address of OID directory.
OSOPBD	Operator statistics batch directory sector.
OSOPB1	Operator statistics first data sector.
OVRREQ	Number of outstanding terminal overlay requests.
PAGDIR	Pointer to the page directory in RAM.
PAGLIF	Page directory life in seconds.
PASSWJ	Job define password.
PASSWP	Print/log password.
PASSWS	Supervisor password.
PRTACT	Print active/idle flag.
PTHING	Pointer to the system constant (entered during system bring-up).
PUSER@	Address of PUSER.
@RTVEC	Pointer to optional remote terminal variables.
SEC95P	95% full point in sectors available.
SECAVL	Total sectors available on all drives.
SECNBR	Number of current PROKS sector.
SECPTR	Pointer to the current PROKS sector buffer.
SECEND	Pointer to the end +1 of the current PROKS sector buffer.
SEMOH	BAM flag.
SYSF84	Pointer to CAT.
SYSFG3	2770 Configuration word.
SYSFG2	HASP Configuration word.
SYSFG1	Additional configuration information (CONFIG sector word 0117).
SYSFIG	Configuration word for the system (CONFIG sector word 0116).
SYSNBR	Current (SCB) system sector in RAM.
SYSPTR	Pointer to the system sector buffer in RAM.
SYSEND	Pointer to the end +1 of the system sector buffer.
TENSEC	Ten second clock.
THING	The system constant area.
....+1	The system constant area.
....+2	The system constant area.
....+3	The system constant area.
TIME	Time = seconds since midnight - 86400.
TIMELW	The master dir. sec. last written to disc.
USER	Last PROKS user table address.
USER8C	Current level 8 user table address
VOLCNT	Tape volume reel count.
WHAT!	Interrupts unexpected.
ZONTOT	Kept in word 0200 of the master directory sector. Byte 0, not used; Byte 1 = Total zones -1; Byte 2 = Largest zone number.
@$MAXL	Pointer to the size of the LDTAB device.
@OVDIR	Address of overlay directory.
@ZONPT	Address of zone pointer table.
$LDINV	Required by IDOS, E3, and above.
:GETOV	Pointer to front of overlay queue.
:PUTOV	Pointer to end of overlay queue.
:FREOV	Pointer to free list of 2-word blocks.
DIABCF	Address of start of 8121 complete-flags.
DIABND	Number of 8121 printers.
DIABP2	Address of basic 8121 driver, PRIN22.
DIABSU	Unit number of the 8121 sysprint. If the system printer is not an 8121, then DIABSU is -1.
LPOUT	Value of IDOS LPOUT.
PRSTAT	Printer status word.
PFCB	Pointer to first SCB.
PLCB	Pointer to END+1 of last SCB
PCB	Address of current SCB.
CHAN2	Zero means channel 2 is idle.
CHAN2R	Non-zero means Level 8 is requesting channel 2.
DISCIO	Non-zero means a disc operation is going.
DISCLK	Non-zero means channel 2 locked-out for disc, in use by tape.
DISCRQ	Count of disc requests pending.
REQTAB	$JDISC request table.
....+1	$JDISC request table.
....+2	$JDISC request table.
....+3	$JDISC request table.
....+4	$JDISC request table.
....+5	$JDISC request table.

CONVENTIONAL DEFINITION AND USE OF INDICATORS

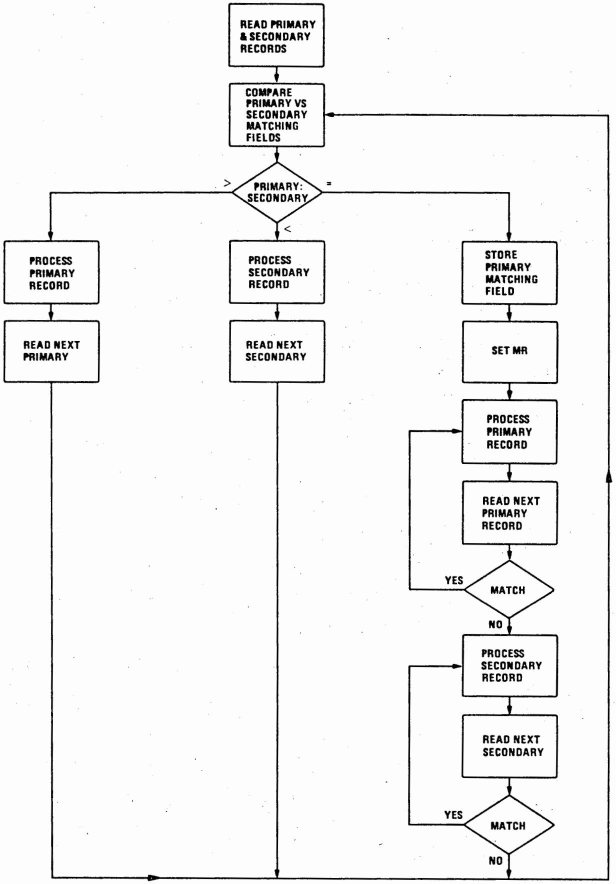
Indicator				Symbol									
Defined/Used	Specification Form	Name	Location	01-99	H0	H1-H9	1P	MR	OA-OG, OV	L0	L1-L9	LR	U1-U8
Where Defined	File	Overflow	33-34						On at 1, 4 Off at 1				
	Input	Record Identifying	19-20	On at 3A Off at 2		On at 3A Off at 4, 7					On at 3A, 3B Off at 2		
		Control Level	59-60								On at 3A, 3B Off at 2		
		Field	65-70	On at 6 Off at 4, 6, 7		On at 6 Off at 4, 7					On at 3B, 6 Off at 2, 4, 6, 7		
	Calculation	Resulting	54-59	On at 4, 7 Off at 4, 7	On at 4, 7 Off at 4, 7	On at 4, 7 Off at 4, 7					On at 4, 7 Off at 2, 4, 7	On at 4, 7 Off at 4	
Defined by RPG Program		External	– –										On at JCL
		Internal	– –		On at 2		On at 1 Off at 2	On at 6 Off at 6		Always On		On at 3B	
Where Used	File	File Condition	71-72										①
	Input	Field-Record Relation (Field Conditioning)	63-64	②		②		①			②	②	①
	Calculation	Control Level (Calculation Conditioning)	7-8							①	①	①	
		Indicators (Calculation Conditioning)	9-17	①	①	①		①	①		①	①	①
	Output	Output Indicators (Output Conditioning)	23-31	①	①	①	①	①	①		①	①	①

① Any of the above defined indicators may be used.

② Only a record identifying indicator is conventionally used.

A768A

MATCHING RECORD ALGORITHM FOR ASCENDING FILES



STATUS BLOCK INFORMATION AND MEANING

```
SAS-20
    STSTBL EQU  S
    LNELAM DCN  O        CODE WORD TO LAM FROM LINE
    %LNDSR DCN  O        DATA SET READY OR NOT READY
    %LNCRR DCN  O        NO ACTION FOR 20 SECS ON LINE
    %LNRVI DCN  O        RVI RECVD
    %LNDIB DCN  O        DATA IN BUFFER
    %LNBID DCN  O        TRYING LINE BID
    %LNRBD DCN  O        BIDS EXHAUSTED RETRY?
    %LNBAK DCN  O        WRONG ACK RECVD, RECORD CHECK
    %LNSYN DCN  O        LINE READING IDLE SYNS FOR AT LEAST 3 SECS
    %LNMT  DCN  O        OK TO SEND BUT NO MSG BLOCK READY
    %LNFUL DCN  O        DATA TO RECVE BUT NO BUFFER TO ACCEPT IT
    %LNCHK DCN  O        BAD BBC RECVD OR SENT
    %LNBEL DCN  O        RECVD BELL SEQUENCE
    %LNOVF DCN  O        OVERFLOW MSG BLOCK RCVD
    %LNADR DCN  O        LINE ADDRESSED
    %LNTSP DCN  O        USER TRYING TO SEND     DATA LINK CHAR, INVALID
    %LNICP DCN  O        INCOMPLETE CONDITION, SEE @LLNKT,@LLCPU,
    *                    @LLABT,@LLNAK
    %LNCLR DCN  O        CANNOT GET CLEAR TO SEND
    %LNHNG DCN  O        DLE W EOT RECVD
    %LNSTP DCN  O        USED TO SEND EOT AS ACK TO TEXT
    @RTRSW BSS  O
    %LNRTY DCN  O        DENOTE GET OR PUT RETRIES NEEDED
    %LNPRT DCN  O        WORD TO INDICATE PRINTER NOT READY
```

LNELAM is set to the following values for the commented conditions:

Value	Condition
1	SENT NAKS TO MSG THEN SENT OR RECEIVED EOT
2	DATA SET IS NOT READY (JUST DROPPED)
3	CPU SENT EOT(DLE-EOT) INSTEAD OF ACK
4	CPU DID NOT ANSWER LINE BID (LINE SENT EOT)
5	CPU DID NOT ACK TEXT SENT (LINE SENT EOT)
6	INCORRECT ACK RECEIVED N TIMES
7	PUTLNE FOUND INVALID TEXT CHAR,FROM USER
8	RCVD EOT INSTEAD OF MORE TEXT (RVI? INQUIRY MOD?)
9	MSG RCVD WAS TO RIG FOR BUFF CHAIN (I SENT EOT)
10	CPU NAKED LINE BID IS TIMES (SENT EOT)
11	CPU SENT EOT AFTER REJECTING TEXT SENT
12	RESERVED
13	CPU NAKED MSG TIL OVERFLOW (LINE SENT EOT)
14	CANNOT GET CLEAR TO SENT ?
15	LINE HAS BEEN IDLE FOR 20 SECS (END RVI SEQ?)
16	SEC TERM, RETREAT AT CONTENTION FOR LINE
17	RESERVED

FORMAT OF LAM COMMON AREA

+ 0	LOGSW 0 = do not log (+) = log continuously (wrap around) (value is the address of the area) (-) = log until log area is full, then set LOGSW = 0 (value is the address of the log area with negative prefix)
+ 1	Log area length (if = 0, LOGSW is set to 0); if bit 0 is on, logger initialized but LOGSW set equal to 0
+ 2	Address of buffer 1
+ 3	Address of buffer 2
+ 4	Address of 16 word custom message
+ 5	Buffer size in bytes
+ 6	Value to use in the IOID for level 1 if low memory address 033 is not available
+ 7	Transparency conversion switch (0 = no, else = yes)
+ 8	Address of TIMER1
+ 9	Address of TIMER2
+10	Address of TIMER3
+11	Address of TIMER4
+12	Address of area containing STATUS information
+13	Address of error statistics table
+14	Address of conversion table if floating (or 0 means was loaded with LAM)
+15	Optional pointer user's real time clock (binary)
+16	Not used
+17	Not Used
+18	Configuration information
+19	Terminal ID received
thru	.
+23	.
+24	Terminal ID to send
thru	.
+28	

ERROR STATUS

@CSQIG	Tally of control sequences recognized but ignored.
@SSS	Count of unrecognized sequences preceded by two sync's. Includes "Wrong ACK" responses.
@ETBI	Number of blocks received.
@ETBO	Number of blocks sent.
@NAKI	Count of NAK's received.
@NAKO	Count of NAK's sent.
@BFSI	Number of IO ins executed.
@BFSO	Number of IO outs executed.

LAM's program letters (ID) are AS.

- Source files are labeled SAS-n, where n (01..51)
- Relocatable are labeled RAS-n, where n (01..51)
- Control files are labeled SAS:n, where n is assorted alpha numerics
- LAM SYSGN source files are labeled SAS+n, when n (40..46)
- LAM SYSGN relocatable files are labeled RAS+n, where n (40..46)

- NOTE: File SAS:CO contains all file documentation.

Control file used to LOAD SYSGN is SAS:CS

SUMMARY OF LAM SOURCE FILES:

SASLM Configuration File - This file is processed by SYSGN and the assembled and loaded with LAM.

SAS-01 Contains: INTLNE
OPNLNE
CLSLNE
ABTLNE
RNGLNE
TRNLNE
HNGLNE

This file can be overlaid and refreshed as needed (as in DATA IV)

This file does not EOP to another file word length:

INTLNE	0174	
OPNLNE	0272	
CLSLNE	0124	0365
ABTLNE	0136	
RNGLNE	0031	
TRMLNE	0017	
HNGLNE	0033	
TOTAL	= 01050	

SAS-30 Point-to-point level 1. Handles line bids and generally those level 1 comm requirements associated with hand shaking (control mode). The auto-answer code resides in this module also.

This file EOP's to RAS-E1
Word length = 0357

SAS-31 Multipoint level 1. Handles polls and selects plus textmode WACKING. This file and SAS-30 are mutually exclusive. (Cannot be loaded together.)

This file EOP's to RAS-E1 as does SAS-30
Word length = 0345

SAS-E1 Contains only an END statement

EOP's to RAS-02
This file allows dechaining modules from the LAM library.

SAS-02 Resident background support for GETLNE and PUTLNE. Contains those portions of GETLNE and PUTLNE that must stay resident while LAM is processing a data stream to or from the user. Both GETLNE and PUTLNE can be overlaid on a file basis.

EOP's to RAS-E2
Word length = 0300

SAS-E2 Contains only an END statement.

EOP's to RAS-50
Function same as SAS-E1

SAS-50 PUTLNE code. This routine accepts logical records from the user and builds transmission ready message blocks and passes them onto LVL1.

EOP's to RAS-E43
Word length = 0341

SAS-E3 Contains only an END statement.

EOP's to RAS-51
Function same as SAS-E1

SAS-51 GETLNE code. This routine deblocks message blocks passed from LVL1 code into LOGICAL records and moves them to the user.

EOP's to RAS-E4
Word length = 0464

SAS-51 Contains only an END statement.

EOP's to RAS-25
Function same as SAS-E1

SAS-25 ASCII/EBCDIC conversion table.

EOP's to RAS-03
Word length = 0400
(DATA IV removes this module from the EOP chain and uses it in overlay 40).

SAS-03 Background save register routine. This module also contains the virtual %LAMVR DCA .048B2M008.
048 day of year into system test.
B2 release level.
M008 apar level.

EOP's to RAS-10
Word length = 022

SAS-10 (Start of level 1 routines)
Level 1 - read and write message block routines.

EOP's to RAS-11
Word length = 0701

SAS-11 Contains routines:
@SNDSQ used to send short control sequences
@WRSEQ used to send buffer of trail pads
@PAD16 used to build 16 word buffer flled with lead or trail pads

EOP's to RAS-12
Word length = 0125

SAS-12 Contains routines:
@CHKSW - deciphers sequence input

EOP's SAS-13
Word length = 0162

SAS-13 Contains routines:
@CHKST - Ensures certain level 1 controller statuses to caller for Send or Receive (Statuses 8436 controller).
@SET1 - Clear to send timeout setter.
@SET2 - Response timeout setter.
@SET3 - Not used.
@SET4 - Read text block.
@SET5 - WACK/TTD timeout.
@SET6 - IDLE timeout.

Generally, these routine are associated with timer setting and timeout routines.Also contains controller I/O routines used to CNTL and I/O the 8436-2 LNE-routines.

EOP's to RAS-15 (there is no SAS-14)
Word length = 0453

SAS-15 Contains routines:
Level 1 communications RAM logger

EOP's to RAS-16
Word length = 0140

SAS-16	Contains routines: @BCCRT - BCC and LRC block check accumulator routines. @ERCTR - Error retry counter routine. Plus, all BSC control characters are defined in this file. EOP's to RAS-18 Word length = 0172 (EBCIDC) 0161 (ASCII)
SAS-18	Optional controller status display routine. Not used by production software users. Used as debug routine. EOP's to RAS-20 (there is no SAS-19) Word length = N/A
SAS-20	Contains no executable code. Contains most constants and variables, switches, flags, etc. for LAM background and level 1, End of EOP chain Word length = 0234
SAS-28	NOP version of SAS-18 EOP's to RAS-20word length = 02 WORD LENGTH = 02 Should be assembled to RAS-18 (used by all Production Users).
SAS-38	Expanded Version of SAS-18. Used for debugging only. Should be assembled to RAS-18. EOP's to RAS-20 Word length = N/AWORD LENGTH = N/A
SAS+40	Main file of SYSGN program.
SAS+41	Terminal ID routine.
SAS+42	Branch table for ? routines.
SAS+43	Contains all ? routines.
SAS+44	Assorted subroutines used by mainline and ? routines.
SAS+45	Decision table and display DCA's.
SAS+46	Replacement records used by ? routines.
SAS+47	Future source for display of LAM configuration.
SAS+C7	Control file modified by SYSGN assemblies and loads RAS-LM to AAS-LM

SYSGN Configuration Word

Bit 23 = 0	if 2780 terminal
= 1	if 3780 terminal
Bit 22 = 0	if compression off
= 1	if compression on
Bit 21 = 0	if auto-EM insertion off
= 1	if auto-EM insertion off
Bit 20 = 0	if point-to-point
= 1	if multipoint
Bit 19 = 0	if inquiry mode off
= 1	if inquiry mode on
Bit 18 = 0	if extended line bid retry on
= 1	if extended line bid retry off
Bit 17 = 0	if terminal-to-CPU
= 1	if terminal-to-terminal
Bit 16 = 0	if primary station
= 1	if secondary station
Bit 15 = 0	if switched line
= 1	if private/leased line
Bit 14 = 0	if terminal ID exists
= 1	if none
Bit 13 = 0	if manual answer Moden
= 1	if auto-answer Modem
Bit 12 = 0	buffer size is 512 bytes
= 1	buffer size is 400 bytes
Bit 11 = 0	if EBCDIC
= 1	if ASCII
Bit 10 = 0	if blank compression
= 1	if full character compression
Bit 9 = 0	if transparent text is to be translated
= 1	if transparent text is not be to be translated
Bits 0 - 8	not used

PRINTER CARRIAGE CONTROLS

Carriage Operation After Printing		3780 ESC SEQS.	2780 ESC SEQS.	Value Passed By GETLNE	ASCII Difference
Single Space		ESC /	ESC /	012	ESC Q
Double Space		S	S	0222	R
Triple Space		T	T	0223	S
Skip to Chan	1	A	A	014	
	2	B	B	0201	
	3	C	C	0202	
	4	D	D	0203	
	5	E	E	0204	
	6	F	F	0205	
	7	G	G	0206	
	8	H	H	0207	
	9	I		0210	
	10	J		0211	
	11	K		0212	
	12	L		0213	
Space Suppress		M		0220	

Restrictions on ASCII Use:

1. Only the buffered sync controller (8436-1) can be used. It must be strapped for the ASCII sync character.
2. No multi-point or transparency
3. Only the Binary Synchronous (BSC) line protocol is supported.

LINE TRACE FORMAT

LAM will optionally log all communications activity in a user designated area of contiguous RAM. Switches to control logging as well as the log area address and length, are located in the Common Area. The first word of the log area always ponts to the next available word of the log. The format of entries is as follows:

I/O INPUT BUFFER ENTRY - 16 words

Each word has a blank in the left byte and data in the right byte..

I/O OUTPUT BUFFER ENTRY - 16 words

Each word has a period in the left byte and data in the right byte.

I/O STATUS ENTRY - 2 words
Word 1 = "STn"   Where n = value of % INDEX when status taken
Word 2 = 0200000 = Ring indicator   Bit Number = 7
0100000 = Data lost = 8
040000 = Output needed* = 9
020000 = Data set ready*# = 10
010000 = Clear to send* = 11
04000 = SYN received# = 12
02000 = Not used = 13
01000 = Receiving carrier# = 14
0400 = Input ready#

I/O CONTROL ENTRY - 2 words

Word 1 = "CTL"

Word 2 = 01 = Reset receiver
02 = Request to send
04 = Reset transmitter
010 = Set data terminal ready
020 = Reset data terminal ready

TIMER ENTRY - 2 words

Word 1 = "TMn" where n = ASCII 0 through 6

Word 2 = Amount of time the clock was set for in tenths of seconds.

When TMn (n=0), second word is indirect contents of common +15 (real time clock).

	0	0	0	0	0	0	0	0
	000	000	000	000	000	000	000	000
Bit	02	35	68	911	1214	1517	1820	2123

* Required to initiate output
\# Required to initiate input

RBS Operator Console Display

The indicators appearing in the display grid are of several general types:

(a) Those that reflect the operating status of the system and flash on and off as the normal course of events proceeds, e.g., "CARRIER OFF".
(b) Those that come on when certain options or operating modes are selected by the operator, e.g., "KYBRD".
(c) Those that come on, accompanied by an audible alarm, when an error in the communication of the data is detected, e.g, "LINE CHECK".
(d) Those that flash when a local device requires attention, e.g., "PRINTER".

All of the indicators are listed below in alphabetical order and an explanation or cross-reference given.

AUTO-EM	Valid only on a system configured as 2780, this comes on when the operator uses the EM command to enable automatic truncation of trailing blanks and insertion of the EM character.
AUTO RSTRT	This comes on to indicate that Automatic Restart mode has been selected with the AUTO RSTRT key. (Restart)
BELL	This comes on when the system receives a signal from the other station indicating that voice communication is desired. It turns off when CHECK RESET is pressed.
BID	This comes on when the system bids for (requests) control of the line, and goes off when it obtains a positive response, i.e., control is granted or denied. Pressing TERM RESET stops the bidding and turns off the indicator.
BID RETRY	This comes on when the system has abandoned an attempt to send after it failed to obtain any response to its request for control of the line (bids). It is turned off (and the bidding restarted) by pressing CHECK RESET.
CARRIER OFF	The carrier signal is the tone that is used to carry the data over the telephone lines. Thus, in order to receive data, the data set must first receive the carrier signal. In a leased-line system which has exlusive use of the telephone line and normally receives a constant carrier signal, the indicator will come on only when a malfunction occurs. In a switched-line system, the periodic loss of carrier is normal, and thus the indicator may flash on and off.
COMPRESS	This is on while the data compression option is selected (when the COMPRESS key was used). Note that this pertains to transmitted data only; decompression of received data is performed as required without operator intervention. Note: Compression type is shown in the SYSGN information.
DATA IN BUFFER	This comes on when there is data in the Four-Phase processing unit awaiting transfer to either the host system or to the printer (or the spool file). It normally goes off when the data is transferrred to its destination, but will also go off when TERM RESET is pressed or the transfer is aborted via keyboard command.
DATA SET READY	This indicator reflects the status of the local data set. On a leased line system, it remains on while the data set has power and is not in test mode. On a switched-line (dial-up) system, its implicaton depends somewhat on the type of data set, but generally, it is on when the data set has power, is not in a "test" mode, and a telephone connection has been established. Note that it does not necessarily mean that the data set is communicating with the other station.
EOF	This comes on when the operator presses the EOF (END of FILE) key to signify that the last card is in the input hopper of the card reader.
HOME	This is on when pressing the LINE/HOME key switches the system into HOME mode. Receiving to disc (spooling version) is allowed.
INCPLT	(Incomplete) This error indicator comes on when the other station aborts the transmission.

INQ MODE	This comes on when the operator selects the "inquiry" mode (via INQ command) on a system in a multi-point 3780 configuration.
KYBRD	This comes on to indicate that the operator has selected keyboard mo (via the KB key), or flashes to indicate that a record has been processed and either another record, EOT or KB must be entered.
LINE	This will be on when pressing the LINE/HOME key switches to LINE mode.
LINE CHECK	If the system is attempting to receive, it means that a redundancy check occurred. The indicator will be turned off after a successful retransmission. If the system is attempting to send, it indicates the lack of response or the receipt of any response other than the specific acknowledgement expected. Compare with RECORD CHECK below.
LOG	This comes on when the operator selects the log mode (via the LOG command), wherein the communications line activity is logged.
MANUAL	Applicable only for systems having auto-answer data sets, this comes on when the operator selects manual operation via the MAN command (this applies to dial-up systems only).
OVERRUN	This comes on if the input buffer overflows, i.e., the incoming message is too long.
PRCSCR IRPT	(Processor Interrupt) This indicates that a reverse interrupt (RVI) has been received from the host system. This prevents completing the sending of data to the host system and may have been accompanied by a printer message from the host system. The indicator will remain on until START or CHECK RESET is pressed, unless AUTO RSTRT is on.
PRINTER	This is on while the printer is available, and flashes if it requires operator intervention.
PUNCH	This comes on when the PUN command is entered to allow data to be transmitted as "punch" records to another Four-Phase site.
READER	This is on while the card reader is operating, and flashes if it requires operator intervention.
RECORD CHECK	At a terminal that is transmitting, this indicates that the receiving station responded to a block of data, but did so with the wrong odd/even acknowledgement. The transmission will be repeated and, if the proper acknowledgement is obtained, the indicator will go off. If the acknowledgement is still unsatisfactory after 15 retries, the indicator will stay on until the operator presses CHECK RESET. Compare with LINE CHECK above.
SEND/ RECV	One or the other of these will be on while the system is sending or receiving data. Both will be off when the line is idle.
SPOOL	Applicable only to Spooling RBS systems (AQ), this will be on when th operator selects the spool mode using the SPOOL key, whereby incoming printer data is stored on the disc.
SYNC TIMEOUT	This comes on when only SYNC (synchronizaton) characters have been received for three seconds. It turns off when a data character is received or when CHECK RESET is pressed or the system accepts a subsequent bid.
TERM ADDR	This comes on to indicate that the terminal system has rejected a line bid because it is not ready to receive. The printer may be unavailabl or the system may be in HOME mode. The indicator will turn off if CHECK RESET is pressed or the system accepts a subsequent bid.
TRANSMIT TIMEOUT	This comes on to indicate a hardware malfunction when the Processing Unit raises "Request To Send" and the data set fails to respond promptly with "Clear To Send". Note, a clear-to-send interrupt is required.
TRNSPCY	This comes on when the operator selects "transparency" mode by pressing the TRNSP key, or if ¢nnnnnn,TS controls a transmission.
TRNSPCY CHECK	This comes on when RBS detects a data-link control character in data that is not being transmitted in transparency mode. It is turned off when either START or TERM RESET is pressed.

RBS KEYBOARD OPERATIONS

The principal mode of the RBS keyboard is that of a system control device where the operator selects and initiates system functions (command mode) by pressing the function keys or by entering command words followed by the EOM key. The function keys and command words for a non-spooling system are discussed below. The command words which apply only to a spooling system are described in Section AQ, RBS Spooling.

The system can be switched to keyboard mode with the KB function key so that the keyboard replaces the card reader as the system input device. This permits the operator to construct card images on the video display unit and "read" them into the system for transmission or printing. These card images must be terminated by the CURSOR RETURN key rather than the EOM key or the keyboard entry is interpreted as a command even though the system is in keyboard mode.

Function Keys

SPOOL (F1)	(Spooling RBS only) This is a "pushbutton" key which enables/disables the spooling of received data to the disc. Spooling may be enabled at any time, but can only be disabled while RBS is not receiving. When spooling is enabled, the SPOOL indicator is on. (If configured for NOPRNT, spooling is always enabled.)
LINE HOME (F2)	This is a "pushbutton" key which switches RBS between LINE mode and HOME mode. ("Cards" may come from either the card reader or keyboard.) Non-spooling RBS (diskette) LINE allows card-to-line and line-to-print HOME allows card-to-print Spooling RBS LINE allows card-to-line and card-to-disc line-to-disc and line-to-print disc-to-line and disc-to-print HOME allows card-to-print and card-to-disc disc-to-print and line-to-disc
AUTO RSTRT	This is a "pushbutton" key that enables/disables the automatic restart of a transmission interrupted by a processor interrupt (RVI) from the host system. The may be enabled/disabled at any time.
COMPRESS (F4)	This is a "pushbutton" key which enables/disables the compression feature in which repetitious data (e.g., multiple blanks in a line) are suppressed during transmission. The degree of compression is determined when the compression feature is enabled. Applies only to the transmission of data.
EOT (F5)	Press this key to signal "end of file" for keyboard input when in "keyboard" mode. Turns off the KYBRD indicator.
KB (F6)	This is a "pushbutton" key that switches RBS between command mode and keyboard mode. Note that switching from keyboard mode with the KB key is "soft" end-of-file and permits concatenation of keyboard and card reader data to the command line. Keyboard mode may also be exited with the EOT key: see EOT above.
EOF (F7)	Press this key when the last stack of cards is in the reader. When the input hopper empties, RBS will terminate the transmission (or close the disc file if cards are being read to disc on RBS spooling). If the hopper empties wihout EOF on, either put more cards into the input hopper or press EOF followed by START to cause end-of-file.
TRNSP (F8)	This is a "pushbutton" key that enables/disables transparency mode. Transmission in transparency mode allows data bytes that would normally be interpreted as Bisync data link characters to be transmitted as raw data.

BELL
(F9)

This is used as a signal that voice communication is desired after two RBS sites have established a dial-up phone connection. It cannot be used between RBS and a host system. Because it requires the transmission of a control code, it cannot be used while data transmission is taking place. When the key is pressed, the audible alarm sounds at the remote system and the BELL indicator appears on its display unit. The operator of the remote system can respond by pressing the BELL key, which will produce the same effect at the originating system. Both operators can then press CHECK RESET to turn the BELL indicators off and switch their telephones from "data" to "talk".

START
(F10)

Pressing this key starts or restarts (after an error condition or processor interrupt) the transmission of data. It also restarts the card reader after DATA CHECK. START resets the indicators of, and provides recovery from, the following conditions:

TRNSPCY CHECK (transparency check)
LINE CHECK
INCPLT (incomplete)
PRCSR IRPT (processor interrupt)

Note: For transparency check, the record must be corrected and reentered. Thus, if the error is encountered while transmitting a disc file, immediate recovery is not feasible as the file must be rewritten. In the case of an incomplete transmission, it is up to the station that transmitted the message to initiate recovery.

CHECK
REST
(F11)

Pressing this key clears the following indicators:

BELL
TERM ADDR
OVERRUN
RECORD CHECK
BID RETRY
SYNC TIMEOUT
PRCSR IRPT (see START above)
PRINTER (stops indicator from flashing)

TERM
RESET
(TOTAL)

Pressing this key initializes the system and makes it ready to receive data. The card reader or printer will be stopped, a transmission or reception in progress or pending will be aborted (on the spooling system the queue is cleared) and the following indicators will be turned off:

TRANSMIT TIMEOUT
DATA IN BUFFER
BID
TRNSPCY CHECK

Note: In Spooling RBS, card-to-disc, key-to-disc, and disc-to-print are not halted with this key, but require the use of the CRR, KBR, or PRR command. See Section AQ.

CURSOR
RETURN

When the keyboard is enabled, press this key to 1) initiate transmission or processing of a record, 2) clear the entry field and return the cursor, and 3) keep the device enabled for more data input.

ERASE
HOME

Press this to restore the cursor to the first byte-postion of the entry field without initiating any other action.

Shifted
ERASE
HOME

This clears message lines 5 and 6 of the operating display. If the communications line is idle, the SELECT OPTIONS message will appear. On a spooling system this also does a DSD: see Section AQ, Operator Commands.

Control
ERASE
HOME

This will erase the line statistics displayed as a result of an STT command. See Operator Commands below.

Non-Spooling RBS Operator Commands

The command words listed below can be entered with the keyboard's typewriter keys while the system is in "command" mode or in "keyboard" mode. The EOM key must be pressed after entering a command. Those commands that are described as "enable/disable" commands operate as on/off "pushbuttons": entering the command enables the function if it is disabled, and vice versa. Note that each keystroke is displayed on line 23 of the operating display and that the cursor control keys described under "Editing Keys" can be used as necessary to correct miskeyings. Additional commands, valid only in the RBS/SPL are described in Section AQ.

ABT Abort the current reception if possible. This should be used to abort when spooling to disc. During non-spooling operations, its use is preferable to pressing TERM RESET.

ALT This command can be used to switch devices (2780<>3780). The buffer size for a 2780 is always 400 bytes; the 3780 uses the size as given in the LAM SYSGN. This command is not permitted on multipoint or onpoint-to-point with Terminal ID.

DIS Enable/disable display of interrupt level 1 activity. As soon as an interrupt occurs on level 1, the right of the top grid line is used for the display

The display consists of nine three-character fields, each corresponding to a bit position in the controller status word. From left to right within each of the fields (except the one corresponding to bit 13, which is not used), an alphabetic character identifies the status bit, the presence or absence of a block cursor indicates the last reported state of the bit (on = true), and a one decimal-digit counter advances every time the bit is reported differently. Note that interrupts are requested by changes in only six of the status bits (Data Set Ready, Receiving Carrier, Clear To Send, Ring,Input easily and Output Request Indicator), and thus, changes in other status bits may not be trapped by the display routine. The status bit fields are as follows(left-to-right):

x = block cursor   z = counter (0-9)

xGz	Ring Indicator
xLz	Transmitted Data Lost
xOz	Output Needed
xDz	Data Set Ready (Modem Ready)
xCz	Clear to Send
xSz	Sync received
	Unassigned (data bit 13 of status word is not used)
xRz	Receiving Carrier
xIz	Input Ready (Data Ready)

EJE Causes the printer to be advanced to top-of-form if the printer is available.

EM Enable/disable AUTO-EM. This command is used only for 2780 configurations. When enabled, AUTO-EM allows the transmission of variable length without requiring EM codes to be punched on short cards (it automatically truncates trailing blanks). It is not permitted with 3780 configurations, in which variable length records are standard.

HNG Enter this on dial-up systems to "hang-up" the telephone. It causes an ending sequence (DLE-EOT) and disconnects an auto-answer modem.

INQ Enable/disable inquiry mode. Inquiry mode on the 3780 permits the interleaving of inputs to the host system from remote sites sharing a dedicated communications line (multi-point). The command is invalid if the system is not SYSGNed as a multipoint 3780. Not valid for 2780.

LOG Enable/disable the logging of communications activity.

LOK To lock the keyboard to prevent an unauthorized use, key in LOCK, press EOM, then enter any three letters and press EOM. The system remains locked until the three letters and EOM are re-entered. Unsuccessful attempts to enter the password cause a message to be displayed on line 24.

MAN Enable/disable manual-answer. This can be used on dial-up systems with auto-answering modems to force the manual-answer mode. This command is invalid if the system is not SYSGNed for auto-answer. Note that the modem must also be switched to manual.

Non-Spooling RBS Operator Commands (CONT).

PUN Transmit to another Four-Phase RBS site as "punch" data.

QUIT Execute a TERM RESET (or TTR on spooling) and exit from RBS. Active communications will be aborted.

STT Display line statistics (effective BAUD and error rate.) The system calculates the statistics based on the data so if the line has been idle, another STT should be done to get accurate values. The BAUD value includes the clear-to-send delay and any delays in transfer due to the CPU or RBS. The error rate is the ratio of bad blocks to total blocks expressed as a percentage. A non-zero value implies hardware or communications line trouble.

To Create an Operational Diskette

1. Assemble and load the application program.

2. Copy the excutable application program to the DKOS Source Disc.

3. Copy FMONTR to the diskette:

```
// FLCOPY
/INPUT=FMONTR.
/OUTPUT=FMONTR.
/MONITR.
/CLEAR
/P.
/SIZE=xx   Where xx is 24, 28, 72, 96 (default 96)
//
```

4. Copy the DKOS processors and routines to the diskette:

```
// DKTGEN
```

5. Copy the application program to the diskette:

```
// FLCOPY
/INPUT=NAME1.
/OUTPUT=NAME2.
/A.                  (Optional-adds checkpoint routine to end of file)

/B=NAMEX.            (Optional-causes NAMEX to load at diskette IPL)
//
```

Bypassing the Auto-Boot Sequence

To bypass the auto-boot sequence for 4100 and 4300 processing units, press the DOS button on the unit while executing the IPL procedure. DKOS processors can then be accessed through FMONTR. The DOS button must be pressed at the end of each processor, otherwise the auto-boot sequence takes over.

If the processing unit has a BOOT switch, the auto-boot sequence is bypassed using the following procedure:

a) Enter the diskette IPL word in the console keys.

b) Set the AUTO/MANUAL switch to MANUAL.

c) Press the SYSTEM RESET switch.

d) Press the BOOT switch.

e) Set console key 0 up.

f) Set the AUTO/MANUAL switch to AUTO.

The auto-boot sequence is bypassed as long as key 0 remains up; to return to auto-boot mode, return key 0 to the down position.

Taking a Checkpoint

In order to take a memory dump, the user must know the memory address of the checkpoint routine. A listing of the diskette directory by DIRDSP will give it. The procedure for a checkpoint is as follows (EXCEPTION: See note for a 3270 application program):

a) Set the AUTO/MANUAL switch to MANUAL.

b) Press SYSTEM RESET and then STEP to clear all I/O activity.

c) Enter 710XXXXX into the console keys, where XXXXX is the checkpoint routine's memory address.

d) Set DISPLAY SELECT switches to TIR (000).

e. Press LOAD and then set the AUTO/MANUAL switch to AUTO.

NOTE: For a 3270 application, in Step C - enter 71100001 into TIR.

The checkpoint routine copies the contents of memory into the diskette file DUMP, then halts. Copy the DUMP file to an IDOS or DOS disc and use FILDMP to print it.

Reloading After a Checkpoint

The 2260 local and remote simulators do not automatically rearm interrupts, so these programs cannot be restarted from the checkpoint. To restart the program from the checkpoint:

a) IPL the system.

b) Enter // DUMP on keyboard 0.

c) After the DUMP file has executed, it will halt. This is a double halt, that is cleared by moving the AUTO/MANUAL switch to MANUAl, then to AUTO, twice.

DKOS System Dump

If an unexplained halt or looping occurs while any of the DKOS procedures are executing, the user can print (need a line printer) a memory dump with the following procedure:

a) Set the AUTO/MANUAL switch to AUTO.

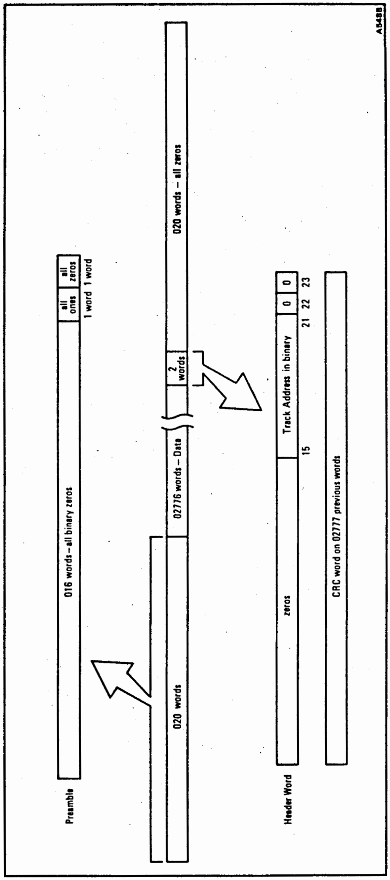
b) Press SYSTEM RESET and then STEP.

c) Enter 71000001 into the console keys.

d) Set the DISPLAY SELECT switches to TIR (000).

e) Return the AUTO/MANUAL switch to the AUTO position.

DISKETTE TRACK FORMAT



$DSKT REQUEST TABLE

Word	Bits	Meaning
REQTAB	23-0	Bit 23 = Head at track 0 Bit 22 = File unsafe Bit 21 = Busy Bit 20 = Illegal request Bit 19 = CRC error Bit 18 = Header error Bit 17 = Out of range Bit 16 = Incomplete length of transmission Bit 15 = Not used
REQTAB+1	23-16	Bit 23 = Restore Bit 22 = Reset file unsafe Bit 21 = } Seek (either bit = 1) Bit 20 = } Bit 19 = Write Bit 18 = Read Bit 17-16 = Not used
	15-9	Track address
	8-0	Not used
REQTAB+2	23-9	Location in memory of data buffer
	8-2	Not used
	1	0 = Save last 2 words in buffer 1 = Don't save words
	0	0 = Check header and CRC 1 = Don't check header and CRC and don't save last 2 words

A546A

RBS SPOOLING OPERATOR CONSOLE DISPLAY

The RBS Spooling Display consists of the Device Status Display (Queue Display) on the upper half of the screen and the condition indicator grid on the lower half. The condition indicators are described in Section AN, Non-Spooling RBS.

DEVICE STATUS DISPLAY

The Device Status Display (DSD) provides three categories of information:

- The state of the current process in terms of the file and the device that has the file engaged.
- The number of available sectors on the disc drives assigned to the system.
- If unattended mode, the actions the system has taken to recover when communications were interrupted due to error detection.

The DSD is automatically updated every five seconds (every 5 minutes in unattended mode), but it can be updated on demand by entering the DSD command. The display occupies lines 2-6 of the operating display, but between updates, lines 5 and 6 may be overlaid by other information, e.g., bad card image.

The format for the display is as follows:

```
    |s|   |nnnnnn@d|#recds| mode/#events     |d/#sectrs
(2)    RDR         |      |UNATTENDED        |0  ssss
(3)    KB          |      |CHCKS             |1  ssss
(4)    CTQ         |      |STRTS             |
(5)    ccc         |      |TERMS
(6)    PRT         |      |ABRTS
```

The left half of the display is concerned with current processes. The devices are listed with appropriate status symbols on the left and file names with record counts appearing on the right. The devices are:

RDR	Card reader
KB	Keyboard
QUE	Transmission queue
ccc	Communications line (SND when transmitting, RCV when receiving data, PUN when receiving punch data)
PRT	Printer

The status symbols for RDR, KB, and QUE are:

A	Active (card-to-line or key-to-line)
S	Sending (disc-to-line)
P	Pending (...the availability of the communications line)
F	Full (applicable to QUE only; the queue is full, other devices may be waiting for space)

In placing a status symbol, a symbol for the device that is the source of the data being sent or received is posted to the left of the communications line entry ("ccc"). The source symbols are:

K	Keyboard
R	Reader
Q	Control queue
$	$NEWFILE

The status symbol W will appear to the left of PTR when the printing of a file has been halted by an STP command.

A file name appearing to the right of RDR or KB means that the device is creating or augmenting a file on the disc, and that is the only time that a file name will be posted with either of those devices. A file named at the right of QUE means that it is a control file and that it has reached the top of the queue and its execution is either in-process or pending. A file name at the right of "ccc" means that it is being sent (ccc = SND) or received (ccc = RCV or PUN). A file named at the right of PTR means that it is being printed.

The record count posted for files being created or augmented by RDR or KB represents the number of records entered since the file was last opened (not necessarily the total number of records in the file). The count posted for a file being executed from the queue (QUE) represents the number of records transferred since the last EOT. No count is posted for files being printed.

The identification number of the active drive, and the octal number of sectors available on that drive will appear at the extreme right side of the display. If the drive is an 8240, it will be asterisked, e.g., "0*05500".

The expression "ATTENDED" will normally appear near the center of the display. If the system is switched to unattended mode by the UNT Command, however, the expression "UNATTENDED" will appear together with the following categories of line restart efforts:

CHCKS	Check resets
STRTS	Starts
TERMS	Terminal resets
ABRTS	Aborts

A six-digit field to the right of each category will register the number of occurrances during receptions, and another six-digit field to the right of the first will register the number during transmissions. The counts will be in decimal and will start from zero when the system is switched to unattended mode.

VCQ DISPLAY

The contents of the transmission queue can be displayed by entering the VCQ command. The name of the control file at the top of the queue appears at the left of display line 2, and the queue extends first to the right, and then down the screen. The format for each entry is nnnnnn@d f where nnnnnn is the name of the control file, d is the drive number, and f is the directory flag. The flag character and their meanings are listed under "Disc Management". If there are no entries in the queue, lines 2-6 will be blank.

RBS SPOOLING OPERATOR COMMANDS

The operation of the keyboard function keys and commands common to spooling and non-spooling RBS systems are described in Section AN. The commands listed are valid only for RBS Spooling (AQ).

Commands may be entered at any time and must be terminated by the EOM key.

BSP	This initiates backing up and reprinting when the contents of a disc file are being printed in response to a PRT command. The STP command can be entered to halt the print operation prior to entering BSP. When the command is entered, a prompt will request the number of sectors to be reprinted. Enter the number of sectors (in decimal) and then press EOM. A sector of disc storage holds about five full (132 column) lines of data; when compression can be used, and/or, when lines are shorter, a greater number will be stored per sector. Pressing EOM without entering a number will cause printing to restart at the beginning of the file. If the STP command was used to stop the printer, the RES command must be entered to restart it, but otherwise, reprinting will begin immdiately.
CHP	This initiates changing the name of the print spool file (from PRINT), thereby making it available for printing or other processing. When the command is entered, a prompt will appear at the bottom of the operating display indicating that the new name can be entered. Up to six characters can be used for the name; the first character cannot be a number (numbers will not be accepted). Pressing the space bar deletes the character under the cursor; this and the INSRT key being disabled prevent the occurrance of blanks in the file name. Press EOM after the name has been entered to execute the change or press TAB to withdraw the request. If input to the file from the communications line is taking place when the command is entered, the change will be effected when EOT is received. Spooling will not be disabled by the use of this command; a new spool file will be created to accommodate input from the line.
CHU	This initiates changing the name of the "punch" spool file (from PUNCH). The rules applicable to the CHP command also apply to CHU.
CRR	This terminates input from the card reader without affecting other devices (as the TTR command would and the TERM RESET key might). Any unprocessed records are lost; if cards are being transmitted, the transmission will be aborted. If a disc file is being created from cards it will be closed by CRR, but no processed records will be lost.
DSD	This forces an update of the disc status display that is included in the operating display. This is explained in detail under "Operating Display" above. Note that the status information is automatically updated by the program at regular intervals.
JOB	This command causes an exit from RBS to the IDOS processor JOB to reclaim recently deallocated disc sectors. Since JOB is an off-line function, it must be executed when the system is idle, and the RBS program reloaded.
KBR	This terminates the keyboard in the same manner as CRR terminates the card reader. Key-to-line (or key-to-disc) is disabled and the keyboard functions solely as a control device. Compare: TTR.
LST	This command can be used in HOME mode to list a deck of cards on the printer. Transfer cards will be printed as normal records and will not be acted upon.
PRR	This terminates the printer (i.e., stops the printing of a disc file) and allows a different file to be specified for printing with a PRT command. It does not affect other devices. Compare: STP, TTR.

PRT	This initiates the printing of a disc file. When the command is entered, a prompt appears at the bottom of the operating display indicating that the file name can be entered. Enter the name and then press EOM to start the printing or press TAB to withdraw the request. If the file is located on a disc drive other than drive 0, the drive number must be specified with the @ symbol, e.g., FILNAM@1. Note that entering the @ symbol automatically places it in the seventh character position, regardless of the number of characters in the file name, that the space bar deletes the character under the cursor, and that the INSRT (right arrow) key is disabled.
RES	This causes the printing of a file started with a PRT command and stopped with an STP command, to be resumed.
STP	This stops the printing of a file started with a PRT command. Compare: PRR. Also see RES.
TTR	This is a total terminal reset equivalent to TERM RESET and the execution of the ABT, CRR, KBR, and PRR commands.
UNT	Enable/Disable unattended mode, wherein certain error recovery procedures will take place automatically. See "Unattended Operation" under "Disc Operations".

RAM DUMPS

1. Taking dumps on the RBS <u>spooling</u> system with a line printer:

    AUTO to MANUAL
    Write down the register contents
    LOAD into TIR 71100001 (71100003 for only the LAM log)
    SYSTEM RESET
    STEP
    LOAD
    MANUAL to AUTO and the dump will begin printing

2. Interpreting the dump:

    If you failed to write down the register contents before printing the dump, RP at the time of the dump may be found indirectly through word 1. The LAM log, if LOG was active, begins at START, near 037500, and is as long as memory permits. Refer to the LAM section in this manual for a description of how to interpret the log.

UNATTENDED MODE

RBS with spooling and the appropriate data linkage permits the host system to place calls to obtain jobs as well as to return the results without an operato in attendance. It can be set to "unattended mode" (with the UNT Command), wherein the recovery from certain error conditions is automatic. If a line-to-printer operation is attempted and the printer is found to be unavailable ("not ready") spooling is enabled. If $NEWFILE transfer statements attempt to load the queue beyond its capacity (20 "Q" statements), RBS will continue to transmit to the initiator of the queued $NEWFILE statements, but will refuse communications from the line until space developes in the queue. The system will attempt to recover from communications errors by simulating CHECK RESET, TERMINAL RESET, or START, as necessary. The transactions that take place while the system is in unattended mode are logged in the Disc Status Display.

TRANSFER STATEMENTS

A transfer statement is a special 80-byte record that is used to create and manipulate disc files other than the PRINT and PUNCH files. The two kinds of transfer statements are: 1) ETX statements, denoted by the ETX character (003) in the first column and 2) $NEWFILE statements, denoted by that expression in the first eight columns.

The $NEWFILE transfer statements, which enter the system through the communications line, can create files with following data, or can augment existing RBS files. $NEWFILE statements can also cause file names to be loaded into the control queue for transmission and/or can delete files. The general characteristics are otherwise the same as those of the ETX statement, except that PRINT and PUNCH are legal names. If a $NEWFILE statement is unrecognizable as such by the RBS, perhaps because of a syntax error, the $NEWFILE is converted to ********* and its following data will be treated as normal print or punch data.

FORMAT

The transfer statement must be in column (byte) 1 of the record, and it may be punctuated by a period. The format is as follows:

¢ nnnnnn@d, ccccccccc   ...or... $NEWFILE nnnnnn@d, ccccccc

where:

¢ is the displayed equivalent of the ETX character. This is produced by a 12-3-9 (&-3-9) multipunch on a card input or a CTRL-C keystroke on the system keyboard. If the statement is printed (in HOME mode), # is substituted because ETX cannot be printed on an RBS line printer.

$NEWFILE is the code word that identifies a record received from the line as a transfer statement. The word must be followed by a space.

nnnnnn, or nnnnnn@d, is the name of the file to be created or operated upon. It may be separated from the ETX character by spaces. Up to six characters, other than spaces or punctuation, can be used for a file name, and a disc drive location can be specified by adding @d (if no drive parameter is entered, drive 0 will be used). The name, including the drive number if applicable, must be terminated by a comma. Any blanks embedded in the file name will be deleted. Thus, NAME A, will be recorded as NAMEA. ccccccccc denotes the parameter field that specifies what is to be done with the named file. Parameters can be entered in any order, and may be separated by spaces. The absence of parameters is considered a syntax error.

PARAMETER DEFINITIONS

D	Create the named file (delete any existing file by that name). The file is closed by a subsequent transfer statement, an EOT if the statement was entered from the keyboard, or by an EOF if it was entered from the card reader.
A	Add the subsequent records to the named file (create the named file if it does not already exist. The file will be closed by any subsequent tranfer statement or, depending on the entry device, by an EOT or EOF. Only files created by RBS can be augmented.
X	Delete the named file. If the named file does not exist, no error indication occurs. If X alone is specified, the named file will be immediately deleted, but if it occurs with other parameters that call for either the creation or the transmission of the file, it is not deleted until the file is sent successfully (the assumption is that all files are created for the purpose of eventual transmission). Note that the deletion of a file does not release the disc space it occupied; the reclamation of disc space requires the exeuction of the IDOS processor JOB. See the JOB command under RBS SPOOLING OPERATOR COMMANDS above.
S	Send the named file. If accompanied by D, the file is created and then sent. If accompanied by A, the file is created or augmented and then sent. The transmission is aborted and an error message posted if the named file does not exist and D or A is not included, or if the named file is not an IDOS chained file. The record length should be 80 bytes; records of other lengths are blank-filled or truncated as necessary. This parameter causes the named file to be merged with data being transmitted or initiates a transmission for the purpose of sending the named file, but it does not terminate a transmission. The E parameter (perferable), the entry of a ¢,E transfer statement, or the pressing of the EOT (for keyboard) or the EOF (for card reader) function key will terminate the transmission after the file is sent.
E	Send and EOT automatically at the end of this file." E accomplishes the same effect as pressing the EOT or EOF function key. It can be used with S in transfer statements within a control file to cause the files invoked by the control file to be sent in a series of separate transmissions. This use allows intermediate receptions and makes it easiler to determine the recovery point if a transmission error occurs during the execution of the control file. It is not necessary to include E in a transfer statement that queues a control file because EOT will always be generated after a control file is executed from the queue. Note: ¢,E can be used (without the file name) to send EOT if the E parameter was left off the statement that initiated the transmission.
F	Freeze the processing of transfer statements until an EOT or EOF is encountered. This permits files containing transfer statements to be created when used with D or A. It also permits control files to be transferred per se (rather than executed) when used with the combination of S and T. F is ignored if used in combination with Q.
Q	Queue the named file for transmission. This causes the specified file name to be place in the queue and releases the device from which the statement originated. When a queued file reaches the top of the queue, it is scanned for internal transfer statements, and they are processed in sequence. EOT is automatically generated after the execution of each control file, but not between files merged by a control file unless the internal transfer statements include E. A queued file does not have to contain transfer statement, thus ¢ FILE,Q where "FILE" contains only data records, is legal.
T	Treat the named file, or all files subject to the named control file, as transparent data. This does not affect the scanning for transfer statements. To transmit a file as transparent data thru the action of transfer statements, both the creating and the sending statements must have the T parameter. An alternative is to use the TRSNSP function key.
N	Do not perform the usual translation of transparent text from ASCII to EBCDIC for transmission. N is ignored if it does not occur with T.

SYNTAX ERRORS

If ETX or $NEWFILE is missing, the statement will not be recognized as a transfer statement--it will be treated as data. If the comma that separates the name field from the parameter field is missing or some other punctuation is substituted, the effect will be that the file name will be too long and there will be no parameter field; an error will be indicated. If the comma is misplaced, the delineation between the file name and the parameter field will be confused; the results will vary. Finally, if the parameter field is blank or holds an illegal combination of parameters, an error will be indicated. The parameters that are legal for various kinds of operations are given below. As a general rule, parameter or a combination of parameters that does not result in an action, e.g., TN without S or D is illegal.

Statement Origin	Immediate Destination of Named File	Legal Parameters
Card/Key	Line	X S E T N
Card/Key	Disc	X Q E A D T N S*
Card/Key	Queue	X Q T N E
Control File	Line	X S E T N
Line ($NEWFILE)	Disc	F E A D T N X**Q**
Line ($NEWFILE)	Queue	T N E X**Q**

* S changes the destination of the file from the disc to the line
** X and Q must be okayed when RBS is configured

If an error is detected in a transfer statement entered from the keyboard or card reader, the incorrect statement is displayed and the system pauses for re-entry. If a transfer statement entered from a control file or from the host system using $NEWFILE is not recognizable or has no parameter field, it is ignored (incoming $NEWFILE data is then treated as normal data). But if it is recognizable, but has an illegal combination of parameters, the system tries to correct it as follows: If A and D occur together, D is deleted. If Q and S occur together, Q is deleted. If the combination is still illegal, parameters are deleted until a legal combination is found. The deletion order is X Q A D N. If no legal combination can be found, the statement is ignored. Generally, control file syntax errors are detected when the files are created, provided that the files are created under RBS and that the T parameter is not used with F in the statements that create the files.

RBS DISC FILES - FORMATS

RBS processes standard IDOS SINDSK-type chained files. Files to be transmitted should have record lengths of 80 bytes. Otherwise, the records are truncated or padded to 80 bytes for punch data or to 133 bytes for print data. The last byte of each print record contains the carriage control character. If an EM (end-of-media) character is received on 2780 punch data, it is made part of the data record.

Files created by RBS have a reserved word at the end of each sector which contains a sequence number used by the reconstruct routines of RBS to verify the file contents when RBS is brought up. The starting sequence number for each file is pseudo-random and is stored in a user word of the directory. Each sector in the file is assigned a number one greater than that of the previous sector. The last pseudo-random starting number is kept in the communications region so that it can be used to generate the next one as needed by new files.

Because of the reserved word, only files with RBS format can be added to with the 'A' option of transfer cards; however, any SINDSK-type file can be sent.

RBS DISC FILES - FLAG BYTE

RBS uses the flag byte in the IDOS directory to indicate the various transfer card options which were used when the file was created.

A The file was opened for augmentation (by a tranfer statement with an A parameter} and RBS went down before closing it. Note: RBS must be reinitiated in order to reconstruct the file before JOB, BOJ, COPY, etc. are run, or the file will be lost. See "nnnnnn@d RECONSTRUCT?" message under "Operating Display".

P The file was created by a transfer statement with either a D or an A parameter.

Q The file holds transparent data.

R The file was created by a transfer statement with either DX or AX, i.e., the file is to be deleted after it is sent.

S The combination of Q and R flags.

T The file name has been loaded into the queue (because of a transfer statement with a Q parameter).

U The combination of flags Q and T (queued file of transparent data).

V The combination of flags T and R (queued file; delete when processed).

W The combination of flags U and V.

Remote Software Update

RSU is a communications program which allows the sending of all types of disc files from one Four-Phase site to another. RSU can be a valuable tool for the Systems Engineer by providing quick access to the latest product updates and fixes.

RSU must be executing at both the local and remote sites, before the dial-up line is established. RSU can be executed thru the keyboard or thru a control file.

```
// RSU
/CONFIG=name   (required-configuration filename)
/PRESCAN       (optional-checks the validity of the CONFIG
.               options and the jobstream which follows)
/option
.
/option     JOBSTREAM
.
.
.
//
```

Commands to RSU (options) can be made from either the local or remote site, but only one side can have control of the file flow at any one time. RSU commands will allow both the local and remote site to send files or sectors ranges from it's site, retrieve files or sector ranges from the opposite site, or send commands to cause execution of control files at the opposite site.

Examples of option commands are:

```
1.  /INPUT=name1@1,TARGET=  @0.
2.  /I=sector address@0,NUMBER=count,T=sector address@1.
3.  /I=address,N=count,T=name@0.
4.  /RETRIEVE=name2@1,OUTPUT=name1@0.
5.  /R=sector address,N=count,O=name@1.
6.  /XEC=NAMEF,D.
```

The D parameter signals the remote site not to disconnect the line upon execution of NAMEF.

```
7.  /WAIT
```

The WAIT command causes the local site to sit idle and respond only to commands from the remote site. One side should always be in a wait, and both sides may be in a WAIT for sending console messages.

The F11 key (on the left of ATTN/MODE) cancels the wait at the local site and causes RSU to continue processing of the jobstream. An XEC command may be sent to cancel a wait at the remote site.

SCREEN DISPLAY OF COMMUNICATIONS ACTIVITY

Line	Information Displayed
1	xx: RC = receiving SD = sending RQ = request
	"filename TO filename" if transmitting data, or "TO CPU" if transmitting JCL.
	"filename FM filename" if receiving data.
	"00000 OF 00000" = sector number of sector numbers, e.g. 00001 of 00010 (first of ten sectors).
	yyy: CHN chained file follows CTG contiguous file follows RQT request for file to be transmitted DNY file requested not at remote site JCL job control language file XEC execute control file after exit SEC individual sector(s) sent or read RSC retrieve sector
	Line one can also display "WAIT" if that is the current status of RSU.
2	The value displayed following "%INDEX" corresponds to values given subroutines that are a part of LAM or MLAM.
	RCVBID: if a cursor is displayed following "RCVBID", a line bid has been received by LAM or MLAM has data in its input buffer. If the character position following "RCVBID" is blank, no line bid has been received by LAM or no data is contained in MLAM's input buffer.
	LNECHK and DSR indicate Line Check and Data Set Ready status.
3	Console message area. "ENTER" is displayed at the beginning of the line if the F1 key is pressed.
4-6	System message(s) area. This area can contain either error messages or console messages.
7-8	Contains the output record (80 characters).

SCREEN DISPLAY OF COMMUNICATIONS ACTIVITY

<table>
<tr><th>Line</th><th>Information Displayed</th></tr>
<tr><td>9</td><td>
<p>Nine 3-character fields that indicate the communications controller card status. The status is displayed only if the $STATUS command is included within the configuration parameters.</p>
<p>The first character of each field identifies the status indicator. The status indicators are:</p>
<table>
<tr><th>FIELD</th><th>CHARACTER</th><th>MEANING</th></tr>
<tr><td>1</td><td>G</td><td>Ring Indicator</td></tr>
<tr><td>2</td><td>L</td><td>Data Lost</td></tr>
<tr><td>3</td><td>O</td><td>Output Needed</td></tr>
<tr><td>4</td><td>D</td><td>Data Set Ready</td></tr>
<tr><td>5</td><td>C</td><td>Clear-to-Send</td></tr>
<tr><td>6</td><td>S</td><td>Sync Received</td></tr>
<tr><td>7</td><td></td><td>Not Used</td></tr>
<tr><td>8</td><td>R</td><td>Receiving Carrier</td></tr>
<tr><td>9</td><td>I</td><td>Input Ready</td></tr>
</table>
<p>The second character is either a cursor or a blank. A cursor indicates that the condition is true. A blank indicates that the bit is off.</p>
<p>The third character indicates the how many times the bit was turned on. The counter increments from zero through nine. Its use is primarily as an indication that the various bits are changing state, rather than as a counter of the number of times they change state.</p>
</td></tr>
<tr><td>10-12</td><td>Contains the input record (80/132 characters).</td></tr>
</table>

KEYBOARD MODE

The F1 key (top row to the right of CTRL) allows the operator to enter a message, CURSOR RETURN will send the message. If CURSOR RETURN is depressed when there is no message on the screen, keyboard mode will be exited, and RSU will continue to process the jobstream. If a file is in the process of transmission, the message will be queued until the current file is fully transmitted. Any of the above commands may be entered thru the keyboard, but only if RSU is executed thru the keyboard. Blank lines must be entered between commands, and only one command may be entered at a time.

RSU Keyboard Functions

Key	Purpose
F1	Requests console keyboard mode. When "ENTER" is displayed, a 75-character message can be entered. If "QUED" is displayed, RSU waits until it finds a convenient point to transmit the queued message. Once in keyboard mode, the job stream is interrupted as soon as the file currently being transferred is complete. Keyboard mode is active until it is exited. This permits the operator to send multiple messages.
CURSOR RETURN or EOM	If the cursor is in the position immediately following "ENTER", the keyboard mode is exited. If the cursor is anywhere else, the data up to the cursor is transmitted. "QUED" is displayed until the message is transmitted.
ERASE HOME	Clears the console message area. The cursor is returned to the first position of the message area.
DEL ←	Moves the cursor backward one position and erases the character in the current cursor position.
F6	Used to enter the "*" character on keyboards that have other keytops in the "*" position.
ATTN (MODE)	Used to abort RSU and exit to DOS, IDOS, or the line trace printer program.
F11	Used to cancel a /WAIT condition at the local site. F11 can also be used to cancel waiting for a response to a request for a file.

CONSOLE MESSAGES

RSU can receive console messages of up to 120 characters in terminal-to-CPU configurations. Messages up to a maximum of 75-byte messages can be transmitted.

CONTROL FILES

There are several control files available which will aid the Systems Engineer in transmitting needed files between the branch office and Software Distribution.

CF240M -	CONFIG file that is compatible with that used by Software Distribution (MOD20).
CUPRSU -	Control file to bring RSU into a WAIT state to transmit files under control of the Cupertino site.
XECRSU -	This control file allows the branch to start execution of any RSU control file in Cupertino to perform updates and transmit files when there is no one attending the Cupertino site (before 9:00 A.M.).
CUPDOC -	A list of all control files which can be executed by XECRSU.
BRID -	SIMED file identifying the branch and the S.E. using RSU at the branch site when using XECRSU.

RSU can simulate the 2780, 3780, and MOD20 line disciplines.

FILE: CF240M

BE AWARE!!!

THE CONFIG FILE CF240M AS RELEASED ON THE CPL AZ01 SPECIFIES $PRIMARY AND MUST BE CHANGED TO $SECONDARY.

```
$MOD20
$SPEED=2400
$MANUAL
$MXBSZ=512
$TR2TR
$SECONDARY
$EBCDIC
$COMPRESSION
$CONSOLE=120
$SCREEN=81
```

FILE: CUPRSU

THIS FILE (CUPRSU) CAN BE USED TO RECEIVE ANY UPDATES WHICH ARE AVAILABLE THROUGH SOFTWARE DISTRIBUTION. THE RESTRICTION TO ITS USE IS THAT SOFTWARE DISTRIBUTION MUST BE INFORMED AHEAD OF TIME WHAT UPDATES ARE NEEDED, AND ARRANGEMENTS MUST BE MADE FOR A CONVENIENT TIME.

CONFIGURATION FILE CF200M EXISTS FOR BRANCHES WHICH RUN AT 2000 BAUD ONLY

```
                    // RSU
                    C=CF240M.
/WAIT.
//
```

CONTROL FILES (CONT).

FILE: XECRSU

```
.
.     THIS FILE (XECRSU) SHOULD BE USED BY THE BRANCH SE FOR ALL RSU
.     COMMUNICATIONS WHEN CUPERTINO IS RUNNING IN UNATTENDED AUTOANSWER
.     STATE.  BRID IS A SIGNON FILE WHICH IS SENT TO CUPERTINO AND LISTED
.     INTO A SPOOL FILE WHICH IS A RECORD OF COMMUNICATIONS ACTIVITY.
.     CUPDOC IS A SOURCE FILE WITH A LIST OF ALL EXECUTABLE CONTROL FILES
.     CURRENTLY AVAILABLE.
.
.            TO USE XECRSU, USE SIMED TO CHANGE THE STATEMENT /XEC=BZMSTR
.            WHICH IS CURRENTLY THERE TO THE NAME OF THE FILE WHICH
.            BEGINS THE UPDATE YOU WANT.  A LIST OF SUCH FILES MAY BE
.            FOUND IN THE FILE CUPDOC.
.
.
.                     BEFORE EXECUTING XECRSU, CREATE A SIMED FILE
.                     (NAMED BRID) WITH THE BRANCH NAME -- YOUR NAME
.
.            WHEN THE CUPERTINO SITE IS IN AN AUTOANSWER WAIT STATE, YOU
.            WILL NEED TO EXECUTE:
.
.                           // XECRSU
.
// RSU
/C=CF240M.

/I=BRID@0,T=@0.

/R=CUPDOC,O=CUPDOC.

/XEC=BZMSTR,D.          (BZMSTR DOES A DEMONSTRATION)

/WAIT.

//
```

FILE: BRID

```
**********************************************************************
**********************************************************************

       BRANCH NAME:
       SE NAME:

**********************************************************************
**********************************************************************
```

THIS FILE IS USED TO IDENTIFY TO SOFTWARE DISTRIBUTION WHO IS OPERATING RSU AT THE BRANCH SITE. THIS FILE WILL MAKE IT POSSIBLE FOR US TO KEEP OUR ACTIVITY LOG CURRENT. IT IS FROM THIS LOG THAT WE KNOW WHO TO INFORM IN CASE OF CHANGES WHICH MUST BE MADE TO FIXES AND UPDATES WHICH HAVE BEEN RELEASED. IT ALSO MAKES IT POSSIBLE FOR US TO CONTACT AN SE WHO IS HAVING TROUBLE USING RSU AND TRY TO CLEAR UP ANY MISUNDERSTANDINGS.

IF DUMPS OR LINE TRACES ARE TO BE SENT TO CUPERTINO FOR ANALYSIS BY FIELD SUPPORT STAFF, THIS FILE (BRID) SHOULD BE USED TO INFORM US OF THE FILENAME, WHAT TYPE OF FILE IT IS (I.E. DATA IV/70 DUMP, 3270 LINE TRACE), WHERE IT IS FROM, TO WHOM IT IS GOING. TRY TO USE FILENAMES THAT ARE UNIQUE TO AVOID HAVING YOUR DUMP OVERWRITTEN BY ANOTHER BRANCH.

TO SEND A DUMP OR TRACE, USE SIMED WITH XECRSU AS INPUT AND DMPRSU AS OUTPUT, REMOVE THE XEC COMMAND AND TYPE OVER THE WAIT COMMAND WITH /I=DUMP FILENAME,T=@0. EXECUTE // DMPRSU AND ESTABLISH THE PHONE LINE BY CALLING THE CUPERTINO DATAPHONE.

FILE: CUPDOC

```
************************************************************************
************************************************************************

          THIS DOCUMENTATION IS A LIST OF CONTROL FILES WHICH CAN
          BE STARTED EXECUTING IN CUPERTINO TO SEND UPDATES TO
          PRODUCTS IN THE FORM OF RELOCATABLE AND ABSOLUTE FILES
          (THE FIRST LIST), OR SNEDIT FILES TO BE APPLIED AGAINST
          THE PRODUCTS SOURCE (THE SECOND LIST).  TO RECEIVE
          THESE FILES READ AND USE THE CONTROL FILE XECRSU WHICH
          MAY BE FOUND ON THE CPL AZ01.  BE SURE AND HAVE THE
          BRANCH-SE IDENTIFICATION FILE BRID ON YOUR RSU PACK.

     FOLLOWING IS A LIST OF ALL CONTROL FILES WHICH WILL BEGIN A PRODUCT
     UPDATE.

     NAME      PRODUCT                              APPROX. TRANS. TIME.

     BZMSTR    -DEMO UPDATE CONTROL FILES           -2 MINUTES
 +   BID1AA    -DATA IV/70 V2/V3 REL D1A            -20 MINUTES
     BI12-C    -DATA IV/70 V2/V3 RELEASE D2         -34 MIN  29 SEC

 +   INDICATES THE UPDATE IS MORE THAN SIX WEEKS OLD AND HAS BEEN WRITTEN
     TO TAPE.  ARRANGEMENTS MUST BE MADE TO RECEIVE THIS UPDATE.

*************************************************************************

     THIS LIST IS OF CONTROL FILES WHICH IF EXECUTED AT CUPERTINO THROUGH
     THE CONTROL FILE XECRSU WILL CAUSE THE CURRENT FIX FILE FOR THE
     PRODUCT WHOSE IDENTIFIER APPEARS IN THE FILENAME TO BE SENT TO DRIVE 0
     AT THE REMOTE SITE.

     BAD30     -IDOS E3 AND E4                      -4 MINUTES
     BAB11     -COBOL E0 AND E1                     -2 MINUTES
     BAB12     -COBOL F0                            -2 MINUTES
     BAB13     -COBOL AB13                          -2 MINUTES
     BAG14     -3270 G2                             -4 MINUTES
     BAS06     -NPSP A2.1 THRU A4                   -2 MINUTES
     BHOA      -VERSION 1 HOA                       -20 SECONDS

*************************************************************************
*************************************************************************
*************************************************************************
```

The current CUPDOC may be obtained by using XECRSU or contacting Software Distribution.

HASP MESSAGE BLOCK FORMAT

SYN	SYN	SYN	DLE/SOH	STX	BCB	FCS	FCS	RCB	SRCB	SCB	data	SCB	data	...

SCB=0	RCB	SRCB	SCB	data	...	SCB=0	RCB	SRCB	SCB	data	...	SCB=0	...

RCB=0	DLE/SYN	ETB	bcc	bcc	PADS

HASP PROTOCOL NOTES:

1.) If the DLE precedes the STX (instead of the SOH) then the block contains transparent data and the ETB plus any extra SYN characters will be preceeded by a DLE. Any DLE characters in the data stream will be preceeded by another DLE. This extra DLE is NOT included in the SCB count. For example, the sequence <u>X</u> <u>Y</u> <u>Z</u> <u>DLE</u> would appear as follows:

...	SCB=C4	E7	E8	E9	10	10	...

2.) BCB (Block Control Byte) can be:

8n	Normal transmission, where n is the sequence count.
90	Bypass sequence count validation for this block.
9n	In LOST TEXT message, n is the sequence count the receiver saw.
An	Resets the sequence count to n.

3.) FCS (Function Control Sequence) is two bytes long. Sample interpretations:

CF CF	System Wait-a-Bit (WABT)
80 C0	Printer/Punch WABT
8x Cy	FCS with data stream control. 1 = allow, 0 = suspend x = print stream, assigned left to right y = punch stream, assigned right to left

Example:	8F CF	Allow all data streams
	87 CF	Suspend printer 1, allow all other streams

4.) RCB (Record Control Byte). The RCB defines the type of record being sen

RCB	Meaning of RCB	Meaning of SRCB
00	End of transmission block	None present
90	Request to initiate transmission	Prototype RCB
91	Console message (receive)	SRCB is ignored
92	Console message (send)	SRCB is ignored
93	Card reader	Always hex 80
94	Printer 1	Carriage Control
95	Punch 1	Always hex 80
A0	Perimission to initiate transmission	Prototype RCB
A4	Printer 2	Carriage Control
A5	Punch 2	Always hex 80
E0	Lost text message	Expected BCB *
F0	Signon/Signoff message **	Signon has hex C1 Signoff has hex C2

* In a lost text message, the receiver returns the sequence count actually received in the BCB.

** Signon and signoff messages are transmitted as 80 byte card images with compression and transparency off. There is no SCB!

5.) SRCB (Sub Record Control Byte) as used for printer carriage controls.

Carriage Control Operation	Hex SRCB for Print BEFORE Advancing CC	Hex SRCB for Print AFTER** Advancing CC	Octal carriage Control Character Returned by RCVLNE
Space Suppress	80	A0	0220
Single Space	81	A1	012
Double Space	82	A2	0222
Triple Space	83	A3	0223
Skip to Channel			
1	91	B1	014
2	92	B2	0201
3	93	B3	0202
4	94	B4	0203
5	95	B5	0204
6	96	B6	0205
7	97	B7	0206
8	98	B8	0207
9	99	B9	0210
10	9A	BA	0211
11	9B	BB	0212
12	9C -> 9F	BC -> BF	0213

** For Print AFTER Advancing carrriage controls, two records are returned to the calling application. The first record contains the "real" carriage control and a blank line, and the second record contains a space suppress carriage control and the "real" data lin

6.) SCB (String Control Byte). The SCB "defines" a record by specifying how many bytes of what kind of data follow in a record. There may be many SCBs in a record as records normally contain a varied mix of duplicate blanks, duplicate characters, and non-duplicate characters.

DUPLICATE BLANKS		DUPLICATE CHARS		NON-DUPLICATE CHARS			
Hex SCB	COUNT	Hex SCB	COUNT	Hex SCB	COUNT	Hex SCB	COUNT
00	*						
80	**					E0	32
81	1	A1	1	C1	1	E1	33
82	2	A2	2	C2	2	E2	34
83	3	A3	3	C3	3	E3	35
84	4	A4	4	C4	4	E4	36
85	5	A5	5	C5	5	E5	37
86	6	A6	6	C6	6	E6	38
87	7	A7	7	C7	7	E7	39
88	8	A8	8	C8	8	E8	40
89	9	A9	9	C9	9	E9	41
8A	10	AA	10	CA	10	EA	42
8B	11	AB	11	CB	11	EB	43
8C	12	AC	12	CC	12	EC	44
8D	13	AD	13	CD	13	ED	45
8E	14	AE	14	CE	14	EE	46
8F	15	AF	15	CF	15	EF	47
90	16	B0	16	D0	16	F0	48
91	17	B1	17	D1	17	F1	49
92	18	B2	18	D2	18	F2	50
93	19	B3	19	D3	19	F3	51
94	20	B4	20	D4	20	F4	52
95	21	B5	21	D5	21	F5	53
96	22	B6	22	D6	22	F6	54
97	23	B7	23	D7	23	F7	55
98	24	B8	24	D8	24	F8	56
99	25	B9	25	D9	25	F9	57
9A	26	BA	26	DA	26	FA	58
9B	27	BB	27	DB	27	FB	59
9C	28	BC	28	DC	28	FC	60
9D	29	BD	29	DD	29	FD	61
9E	30	BE	30	DE	30	FE	62
9F	31	BF	31	DF	31	FF	63

* An SCB of 00 means End of Record. However, if the first and only SCB in a record is 00 then the record is an End-Of-Job (EOJ) sequence (See Note 6).

** An SCB of 80 means Spanned Record to next transmission block.

6.) End-Of-Job (EOJ) Sequence.
This sequence is a special record used to terminate the data stream that was begun with a Request to Initiate Transmission message. The RCB of the record indicates which stream is being ended. For example, the last record of a file going to Printer 1 would be followed by the EOJ sequence:

...	RCB=94	SRCB=80	SCB=00	Next RCB	...

MLAM Line Control Block (LCB)

+0 log switch
  0 = do not log
  non-0 = log in memory area starting at address specified as value o log switch. Once initialized with address of log area, switch can be dynamically set on and off. If address initialized with negative prefix (MZE). logging is initialized but log switch is then set to zero.

+1 log area length (in words)
+2 address of user logic error exit routine
+3 reserved
+4 address of LVL4 TIMER routine
+5 transparency conversion switch for sending
  (0= convert to EBCDIC; non-zero = do not convert to EBCDIC)
+6 transparency switch
  0= transmit records in transparent mode
  non-zero= transmit records in non-transparent mode
+7 address of master IOID table if low memory address 033 is not available
+8 address of conversion table if floating
+9 address of MLAM status table (line indicators)
+10 max character length of transmitted console messages
+11 number of transmit line buffers
+12 number of receive line buffers
+13 transparency conversion switch for receiving
  (0 = convert to ASCII; non-zero = do not convert to ASCII)
+14->20 reserved
+21 word length of line buffers (not including 4 word CCB)
+22 address of CCB preceding buffer 1
+23 address of CCB preceding buffer 2
.
.
+n address of CCB preceding buffer n

MLAM STATUS WORD

%STATS DCN 0 = No abnormal condition has occurred.
non 0= Something weird has happened.

non 0 values of %STATS

1 = Found DLE.SYN.ENQ.ITB.ETX.ETB character while sending non-transparen text. Transparency check.
2 = Data Set dropped ready while sending or receiving text. Radial may be required.
3 = LOST DATA on Send (restart job) (MLAM received lost data msg).
4 = LOST DATA ON RECEIVE (restart job) (MLAM sent lost data msg).
5 = Overflow message block received (probably have wrong buffer size).
6 = RCVLNE processing record that overflows user's area.
7 = Reserved
8 = Cannot get clear to send (hardware problem).
9 = Unknown SCB encountered in RCVLNE buffer.
10= RCVLNE encountered an unknown RCB (Record Control Byte) in the message block it is presently deblocking.
11= MLAM has received a startup ENQ/EOT from the remote station, indicating that it is restarting.

STATUS TABLE - Line Incicators

```
%LNTAB EQU    $
%LNDSR DCN    0       0 = DATA SET NOT READY ELSE READY
%LNCRR DCN    0       NON 0 MEANS RECEIVING CARRIER
%LNRSQ BSS    1       EXPECTED BCB CURRENT RITE ADJ; Q TO 017
%LNRT  BSS    1       LAST RECEIVED BCB RITE ADJ. 0200 to 0217
%LNERS BSS    1       EXPECTED BCB ON LAST DATA LOST SENT
%LNERR BSS    1       EXPECTED BCB IN LAST DATA LOST RECVD
%LNER1 BSS    1       RECVD BCB IN LAST DATA LOST RECVD
%LNDRC DCN    0       NON 0 MEANS DUPE DATA RECVD
%LNNBC DCN    0240    NEXT BCB TO SEND IN 02YY FORM
.                     INCREMENTED WHEN ACK IS SENT
.                     OR RESET WHEN RESET BCB RECVD
%LNSON DCN    0       NON 0 WHEN SIGNON RECVD SET BY RCVLNE
                      CLEARED BY USER
%LNSOF DCN    0       SET NON 0 WHEN SIGN OFF RECVD
                      SET BY RCVLNE AND CLEARED BY USER
%LNRFL DCN    0       SET NON 0 WHEN RECVE BUFFERS ARE FULL
                      CLEARED WHEN RECVE BUFFER IS AVAILABLE
%LNT20 DCN    0       SET NON 0 WHEN 20 SEC TIMEOUT OCCURS WHEN
                      USING AUTO ANSWER MODEMS
                      MUST BE CLEARED BY USER
%LNDOB DCN    0       SET NON 0 WHEN MSG BLOCK IS READY TO SEND
                      CLEARED WHEN NO BUFFERS TO SEND
%LNDIB DCN    0       SET NON 0 WHEN MSG BLOCK IS READY TO DEBLOCK
                      SET TO 0 WHEN NO MSG BLOCKS ARE AVAILABLE
%LNNKO DCN    0       SET NON 0 WHEN NAK BEING SENT DUE TO BAD
                      TEXT RECVD
                      CLEARED WHEN GOOD TEXT RECVD
%LNNKI DCN    0       SET NON 0 WHEN NAK RECVD
                      CLEARED WHEN GOOD TEXT OR ACK RECVD
%LNLSR DCN    0       SET NON 0 WHEN DATA LOST MSG SENT
                      CLEARED WHEN GOOD BCB RECVD
%LNLSS DCN    0       SET NON 0 WHEN DAT LOST MSG RECVD
                      CLEARED WWEN ACK OR GOOD TEXT RECVD
%LNTHN DCN    0       NOT USED
%LNRXT DCN    0       SET NON 0 WHEN RETRANSMITTING TEXT
                      CLEARED WHEN GOOD TEXT OR ACK RECVD
%LNIDL DCN    0       SET NON 0 WHEN IDLE ACKING IS OCCURING
                      CLEARED WHEN PERMISSION EXISTS FOR ANYTHING
%LNCLR DCN    0       SET NON 0 WHEN LVL1 CANNOT GET CLEAR TO SEND
%LNSRT DCN    0       RETRY SWITCH FOR SNDLNE/SNDEOJ
                      0= NO RETRY NEEDED RECORD ACCEPTED
                      = NON 0 DO RETRY CALL TTO SSNDLNE OR SNDEOJ
                      = 1 MLAM WAITING FOR PERMISSION TO SEND
                      = 2 NEED LINE BUFFER
                      = 3 NEED BUFFER FOOR SPANNED RECORD
                      = 4 WAITING FOR LVL 1 TO SEND
                      ALL QUEUED BUFFERS
%LNRRT DCN    0       RCLNE RETRY SWITCH
                      NON 0 DO RETRY CALL TO RCVLNE
                      WHEN 0 MEANS RECORD WAS PASSED TO USER
%LNTSP DCN    0       SET NON 0 WHEN SNDLNE INTERCEPTS TRANSP
                      TEXT CHAR. IN NON TRANSP RECORD FROM USER
%LNDTR DCN    0       SET NON 0 WHEN DATA TERM READY
                      CLEARED WHEN DATA TERM NOT READY
%LNTSR DCN    0       SET NON ) WHEN RECVING TRANSPARENT TEXT
                      CLEARED WHEN NON TRANSP TXT RECVD
%LNDMY DCN    0       SET NON 0 WHEN TEXT RECVD AND LOST
                      WHEN NOT EXPECTED
```

STRSAM (Sample NTP/150 Program) - Rel CO

The initial format displayed by STRSAM is named "Z1". The prompt displayed by this format is "PLEASE ENTER FORMAT NAME".

The following keys are STRSAM function keys:

Key	Function
CTRL A	Full screen edit of all fields
CTRL B	Return to initial format
CTRL C	Return to IDOS
ENTER	Full screen edit of all fields
PF1	Chain forward to next format
PF2	Chain backward to previous format

To build a custom STARTER (NTP/150) program based on STRSAM:

1. Make keyboard modifications for program functions keys:

PE7226	Control file to perform table edits
CE7226	STRSAM keyboard changes

2. Run GNSTM to configure the NTP/150 system: It will halt so you can check keyboard changes before performing table edits. The generation questions are identical to those in A76. See the NTP/100/150 System Generation and Debugging Guide.

3. Modify the SCIAUB copy file OCCURS clauses to reflect the number of videos configured in Step 2 in

    EDIT-WORKING-AREAS
    DIRECTORY-AREAS
    START CONSTANTS

```
01  EDIT-RECORD-AREA COPY SCIACB (SCIACB).
    02  EDIT-WORKING-AREAS OCCURS 4 TIMES
            INDEXED BY X-EDIT-HDR.
        03  EDIT-RECORD-HEADER.
            04  FORMAT-NAME, PIC X(6).
            04  FILLER, PIC X(6).
            04  NEXT-FORMAT-NAME, PIC X(6).
            04  PREVIOUS-FORMAT-NAME, PIC X(6).
            04  ERROR-LINE-NR, PIC S999 COMP.
            04  FILLER, PIC S999 COMP.
            04  FILLER, PIC S999 COMP.
            04  REVISION-LEVEL, PIC S999 COMP.
            04  FORMAT-NUMBER, PIC S999 COMP.
            04  ERR-MSG-LTH, PIC S999 COMP.
            04  FILLER, PIC X(6).
        03  EDIT-RECORD-DETAIL OCCURS 12 TIMES
                INDEXED BY X-EDIT-DET.
            04  CURRENT-ROW-COLUMN.
                05  CURRENT-ROW, PIC S999 COMP.
                05  CURRENT-COLUMN, PIC S999 COMP.
            04  FORWARD-TAB, PIC S999 COMP.
            04  FIELD-MAXIMUM-LTH, PIC S999 COMP.
            04  FIELD-MINIMUM-LTH, PIC S999 COMP.
            04  NUMERIC-LOCK-FLAG, PIC S999 COMP.
            04  FIELD-EDIT-RULE-NR, PIC S999 COMP.
            04  FILLER, PIC S999 COMP.
            04  DISPLACEMENT, PIC S999 COMP.
            04  OPTIONAL-REQUIRED-FLAG, PIC S999 COMP.
            04  DATA-ELEMENT-NR, PIC S999 COMP.
            04  FILLER, PIC X(18).
            04  TASK-NUMBER, PIC S999 COMP.
            04  FILLER, PIC X(6).
```

Modify this OCCURS clause to correspond to the number of terminals on your system.

```
02  DIRECTORY-AREA REDEFINES EDIT-WORKING-AREAS
    OCCURS 4 TIMES.
    03   DIRECTORY-HEADER.
         04   SCREEN-FORMAT, PIC S999 COMP.
         04   NUMBER-OF-SECTORS, PIC S999 COMP.
         04   FREE-POINTER, PIC S999 COMP.
         04   HIGH-FORMAT-NAME, PIC X(6).
         04   D-FREE-POINTER, PIC S999 COMP.
         04   FILLER, PIC X(3).
    03   DIRECTORY-DETAIL OCCURS 83 TIMES.
         04   ENTRY-NAME, PIC X(6).
         04   ADDRESS-POINTER, PIC S999 COMP.
```

Modify this OCCURS clause to correspond to the number of terminals on your system.

```
02  START-CONSTANTS OCCURS 4 TIMES
    INDEXED BY X-EDIT-CONS.
    03   HOLD-EDIT-SUB, PIC S999 COMP.
    03   DIR-POINTER, PIC S999 COMP.
    03   TABLE-ADDRESS, PIC S999 COMP.
```

Modify this OCCURS clause to correspond to the number of terminals on your system.

```
02  RETURN-CODE, PIC S999 COMP.
02  LAST-FIELD-FLAG, PIC S999 COMP.
02  SIG-LENGTH, PIC S999 COMP.
02  B0, PIC S999 COMP, VALUE 0.
02  B1, PIC S999 COMP, VALUE 1.
02  B2, PIC S999 COMP, VALUE 2.
02  B3, PIC S999 COMP, VALUE 3.
02  B4, PIC S999 COMP, VALUE 4.
02  B24, PIC S999 COMP, VALUE 24.
02  B80, PIC S999 COMP, VALUE 80.
02  SAVE-FORMAT-NAME, PIC X(6), VALUE SPACES.
02  AT-FIELD-PROCESS-FLAG, PIC S999 COMP, VALUE 0.
02  ERROR-BYTE-POSITION, PIC S999 COMP, VALUE 1.
02  WS-FORMAT-NAME, PIC X(6).
02  FAILED-ERROR, PIC X(80), VALUE "FAILED EDIT".
02  DATA-REQ, PIC X(80), VALUE "FIELD REQUIRES DATA".
02  MIN-ERROR, PIC X(80), VALUE "MINIMUM LENGTH ERROR".
02  NO-FORMAT-MSG.
    03   FILLER, PIC X(7), VALUE "FORMAT".
    03   ERROR-FORMAT-NAME, PIC X(6).
    03   FILLER, PIC X(15), VALUE "DOES NOT EXIST".
    03   FILLER, PIC X(52), VALUE SPACES.
02  PASSED-EDIT-MSG, PIC X(80),
         VALUE "ALL FIELDS HAVE PASSED THE EDITS."
02  INVALID-SUPER-TAB-MSG, PIC X(80),
         VALUE "INVALID SUPER-TAB POINTER ENCOUNTERED".
02  ERROR-MSG, PIC X(80), VALUE SPACES.
```

INDEXES

The three indexes defined in the edit file which must be set prior to accessing the edit information are as follows:

X-EDIT-HDR   Indexes the header information for each format. Before using it, the program must set this index to the logical terminal number.

X-EDIT-DET   Indexes the detail information for each field. Before using it, the program must set this index to the proper field number.

X-EDIT-CONS   Indexes the items in START-CONSTANTS. Before using it, the program must set this index to the logical terminal number.

Also modify:

```
01  WS-CONSTANT
    02   B-NAME PIC X(6) VALUE SPACES.
    02   INITIAL-FORMAT-NAME PIC X(6) VALUE "Z1  ".
    02   NUMBER-OF-SCREENS PIC S999 COMP VALUE 4.
```

Modify this to correspond to the number of terminals on your system.

4. Compile STRSAM (COBOL Source Program).

5. Modify the appropriate control file for the load step:
   STMDBG   Load step for debug system
   STMNON   Load step for non-debug system
   - (a) Communications: Modify all /I. = parameter to /I=parameter to inclclude the relocatables for communications.
   - (b) Modify the bottom parameter in the user file:
     - (1) Communications: Bottom parameter equal to bottom in File C77G-E (see NTP/100/150 System Generation Guide-Section 4).
     - (2) No communications: Bottom parameter equal to octal address at top of last screen (see NTP/200 Programmer's Guide, Table 4-1).
   - (c) Insert relocatable of user code following the comment card in LOADOV.
   - (d) /I = STRLIB must be a user input.

6. Run STMNON or STMDBG.

7. Allocate Format and Edit files - ECIAUA

```
// ACIAU
/  FORMATS=NUMBER OF FORMATS TO BE STORED
/  VIDEO SIZE=24 X 80
/  OUTPUT=NAME OF FORMAT FILE (DEFAULT:FMTFIL)
//
// ACIAUA
/  FORMATS=SAME NUMBER AS ABOVE
/  VIDEO SIZE=24 x 80
/  OUTPUT=NAME OF EDIT FILE (DEFAULT:EDTFIL)
/  NUMBER OF ENTRY FIELDS PER FORMAT=AVERAGE NUMBER
//
```

STARTER - NTP/200

Under NTP/200 COBOL without STM, terminal control is handled entirely by the application program. Unlike NTP/150 the KEY-IN statement will be used. Refer to the NTP/200 COBOL Language Definition Manual and NTP/200 COBOL Programmer's Guide.

To build a CUSTOM NTP/200 STARTER PROGRAM:

1. Allocate Format and Edit File - ECIAUA

2. Normal COBOL development: It must include the format and edit parameters in the format and edit files (working storage-copy member SCIACB). All STARTER subroutines are usable except the STARTER "MOVE" subroutine. "I=STRLIB" must be included in LOADOV JCL.

ADDING EDIT RULE TABLES

The source program to change when adding additional rule tables is SCIALZ which is assembled using the control file CCIALZ. Each rule table is sixteen words long. The right-most sixteen bits of each word are used as the validity mask. For example, if one character field is to be only the characters Y or N, the additions to SCIALZ are

```
EDIT05    BSS  0
          DCN  0          WORD 1
          DCN  0          WORD 2
          DCN  0          WORD 3
          DCN  0          WORD 4
          DCN  040000     WORD 5 - "N" Bit on
          DCN  01000      WORD 6 - "Y" Bit on
          DCN  0          WORD 7
          DCN  0          WORD 8
          DCN  0          WORD 9
          DCN  0          WORD 10
          DCN  0          WORD 11
          DCN  0          WORD 12
          DCN  0          WORD 13
          DCN  0          WORD 14
          DCN  0          WORD 15
          DCN  0          WORD 16
```

EDIT RULE BIT DESCRIPTOR CHART

BITS WORDS	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1																
2																
3	/	.	-	,	+	*	(	)	'	&	%	$	#	"	!	
4	?	>	=	<	;	:	9	8	7	6	5	4	3	210		
5	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	@
6	_				[	Z	Y	X	W	V	U	T	S	R	Q	P
7	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a	'
8						z	y	x	w	v	u	t	s	r	q	p
9																
10																
11																
12																
13																
14																
15																
16																

STARTER UTILITY PROGRAM - STRUTL

STRUTL is used to list the FMTGEN format directory, list all (or a selection) of formats from the format file, modify the format chain information, or display file statistics.

```
                          STARTER UTILITY PROGRAM OPTIONS

                   PF1 = VIDEO DISPLAY DIRECTORY

                   PF2 = LIST FORMAT DIRECTORY

                   PF3 = LIST FORMATS

                   PF4 = MODIFY FORMAT CHAIN DATA

                   PF5 = FILE STATISTICS

                   PF6 = RETURN TO IDOS
***************************************************************************

         FORMAT NAME:

     PREVIOUS FORMAT:

         NEXT FORMAT:
```

STARTER - NTP/200

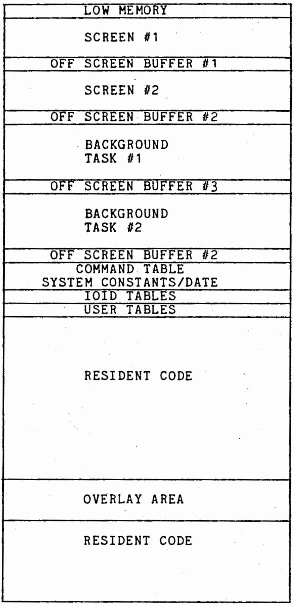
Under NTP/200 COBOL without STM, terminal control is handled entirely by the application program. Unlike NTP/150 the KEY-IN statement will be used. Refer to the NTP/200 COBOL Language Definition Manual and NTP/200 COBOL Programmers Guide.

To build a CUSTOM NTP/200 STARTER PROGRAM:

1. Allocate Format and Edit File - ECIAUA
2. Normal COBOL developement. It must include the format and edit parameters in the format and edit files (working-storage COPY member SCIACB). All STARTER subroutines are usable except the STARTER "MOVE" subroutine. "I=STRLIB" must be included in the LOADOV JCL.

SYSTEM MEMORY ORGANIZATION

SAMPLE MEMORY LAYOUT



USER TABLE DEFINITIONS

To locate the user tables in a dump or in RAM examine the IOID instruction in location 6 to find the keyboard IOID table. Each entry in the keyboard IOID table points the the first word in each user table.

WORD	LABEL (REDEFINED)	DESCRIPTION
000		Word 0 used by IOID BRM for keyboard
001	BKEY	BRM to accept key stroke
002	BBRD	Exit interrupt
003	ECHAR	Key board pointer - queue top
	(KPTOP)	
	(BCKACT)	Background task is active
004	KEYPNT	Key board pointer - queue bottom
	(QUE)	Address of queue address
005	FWA	First word address of screen
006	KTYPE	Keyboard type (RCPY Instruction)
007	CASE	Upper case flag
010	STX2	Save index register 2 for status test
011	IOX2	Save index register 2 for disc routines
012	LOCK	Lock keyboard flag
013	Unused	
014	LKEYS	Legal keys bit flag (1 of 2)
015	LKEYS1	Legal keys bit flag (2 of 2)
016	NAME0	First three characters of document name
017	NAME1	Last three characters of document name
020	LSECTR	Ending sector of document
021	DIRFLG	Directory flags
022	NSECTR	Number of sectors in document
023	SSECTR	Starting sector of document
024	PSSWRD	Password (hashed)
025	HSECTR	Header sector address
026	CURS	Character under cursor
027	CURCHR	Cursor character
030	CURC	Cursor blink rate
031	CLSW	Character/cursor switch
032	CHAIN1	Address of requested operation
033	TASK1	TASK1
034	CHAIN2	Address of next requested operation
035	TASK2	TASK2
036	WAIT	Wait for routine instruction
037	ABRTA	Abort address
040	MD2WN	IV90-2 window register
041	WRTFLG	Write screen flag
042	URA	Save RA
043	UURB	Save RB
044	U23	Save X2,X3
045	TSKADR	Task address
046	DSTAT	Request table disc
047	DEVS	Request table disc
050	RAM	Request table disc
051	SECT	Request table disc
052		Request table disc
053	IMODE	System mode switch
054	CRSCNT	Cursor column Number
055	Unused	
056	CURSCH	Cursor position
057	CURSWD	Character in cursor position
060	LMARG	Left margin column counter
061	RMARG	Right margin column counter
062	TXNAME	Name of the text area
063	TXNAM1	Name of the text area
064	TXFWA	FWA of the text area
065	TXLWA	LWA of the text area
066	NAMEA	Second document name
	(SVDNA)	Temporary storage for document name
067	NAMEA1	Second document name
	(SVDNAM)	

USER TABLE DEFINITIONS - CONTINUED

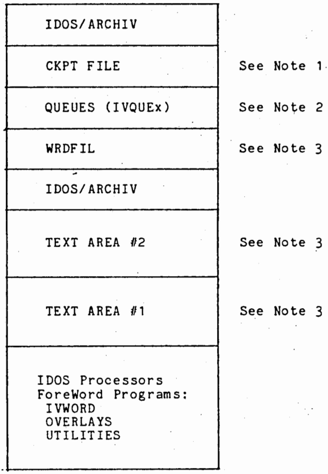
070	TXNMA	Second text area name
	(SVTNAM)	Temporary storage for text area name
071	TXNMA1	Second text area name
	(SVTNM1)	
072	DRVA	Second drive number
073	GLSFWA	First sector address of glossary
074	GLRDSC	Glossary disc number & glossary active flag (1B0)
075	GLTXFW	FWA of glossary text area
076	TABMK	Tab marker
077	AUTAB	Auto tab to column number
100	DECTAB	Decimal tab active
101	CENTR	Center line
102	FLG	Cursor return flag
103	Unused	
104	STRTCH	STRTCH
105	STRTWD	
106	OLDSCH	OLDSCH
107	OLDWRD	
110	NEWSCH	NEWSCH
111	NEWRD1	
112	RHTLWC	RHTLWC
113	RHTLWD	
114	LFTLWC	LFTLWC
115	LFTLWD	
116	UPSCH	UPSCH
117	UPWRD1	
120	PLCH	PLCH
121	PLCH2	
122	BBPO	BPO
123	PPN	PPN
124	FPO	
125	LFTSEC	LFTSEC
126	NEWCNT	NEWCNT
127	THISEC	THISEC
130	RHTSEC	RHTSEC
131	STRSEC	STRSEC
132	OLDCNT	OLDCNT
133	FLX2	FLX2
134	FLCH	FLCH
135	CHPRG	CHPRG
136	RMRGN	Right margin new
137	OLDDSC	Old disk number
140	NEWDSC	New disk number (INSERT)
141	NEWFWA	NEW TXAREA ADDRESS (INSERT)
142	CMODE	Command mode
143	CMDCD	Command code
144	CMDSCH	Command cursor save
145	CMDSWD	
146	PPCMD	Current sector save
147	CDSRET	Return address
150	PRBRA	Address of print routine
151	TOPSKP	Number of lines in the top margin
152	BODY	Number of lines in the body of text
153	LNSPG	Number of lines per page
154	PRSW	PRSW
155	PRCNT	PRCNT
156	PRFLG	PRFLG
157	PRSTP	PRSTP
160	PRSPC	PRSPC
161		
162	PREQT	Printer request table
163	PRTYPE	
164	PRSTAT	
165	PRFWA	
166	PRLMRG	
167	PRUNDR	

170	PRVERT	
171	LFLCH	LFLCH
172	NPLSW	NPLSW
173	WPLSW	WPLSW
174	TBUF1	Wrap around buffer
175	TBUF2	
176	TBUF3	
177	TBUF4	
200	TBUF5	
201	TBUF6	
202	TBUF7	
203	TBUF8	
204	TBUF9	
205	TBUF10	
206	PLCNT	PLCNT
207	RPCH	
210	RPCNTR	RPCNTR
211	RPCNT	RPCNT
212	LBUF1	Address of 1st phrase
213	LBUF2	Address of 2nd phrase
214	RPLSCH	RPLSCH
215	RPLCNT	RPLCNT
216	DOCTIM	Start time document was opened
217	DOCLIN	Number of lines printed from the document
220	KCOUNT	Key stroke counter

The length of the user table is defined in the module STRTUB

SYSTEM DISC ORGANIZATION

The general organization of the disc is as follows:



Note 1: The CKPT file is a 200 sector contiguous file which is automatically created when ForeWord is brought up if there is sufficient disc space on drive zero. This file is used when a automatic or manual dump of the system is taken.

Note 2: The queues are normally 3 sector contiguous files in which are stored the commands to be executed. The queue size can be changed during WRDGEN if the destination system requires expanded queues. Each sector in the queue contains eight queued tasks. ForeWord will automatically create these queues on the system if they do not exist when IVWORD is first brought up. The IDOS file IVQUE0 will contain all tasks which have been put into queue one, etc.

Note 3: Text areas (including the system display file - WRDFIL) must have a "t" as the flag byte. Care must be taken to insure there are no more than 32 entries in the system display file (WRDFIL). Any of the displays which fall in the greater than 32 category will not be located by system initialization.

The organization of the text areas are as follows:

Relative Sector 0	ALLOCATION SECTOR
Relative Sectors 1-7	TEXT AREA DIRECTORY SECTORS
Relative Sectors 8 - end	TEXT AREA DATA SECTORS

ALLOCATION SECTOR

A single sector which is the first sector in the text area. The first sixteen bits of each word are set to zero if the associated sector is in use and set to one if the sector is available for allocation. The allocation table is build by PRETXT up to the size of the text area specified.

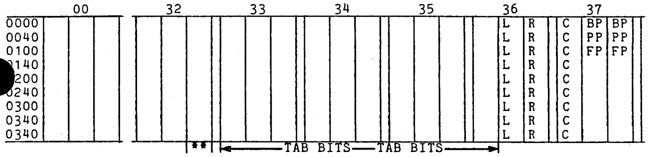
Words	0000 - 0371	Allocation Bit Map.
Word	0372	Total number of sectors in text area
Word	0373	Number of unused sectors
Word	0374	Number of unused directory entries
Words	0375 - 0377	Reserved

TEXT AREA DIRECTORIES

Six sectors (relative 1 thru 7). Each entry is eight words long. Maximum number of entries is 192. The eight word entries appear as follows:

WORD	CONTENTS
0	First three ASCII characters of the document name.
1	Last three characters of the document name.
2	Last sector address relative to the start of the text area.
3	Reserved
4	Left byte is the flag byte for the document. Right two bytes contain the sector count of the document.
5	Staritng sector address relative to the start of the text area.
6	Hash coded passowrd. Zero if no password.
7	Starting sector of the header. Zero if no header.

DATA SECTORS



** - Column 81 indicator column

Words 0 - 32 are the data positions which are visable on the screen.
Words 33 - 35 and part of 36 are tab bits where the bit is set to one if the tab has been set in the corresponding position and set to zero if no tab bit has been set.
Word 36 center and right byte are the left and right margin column numbers for the paragraph of text.
Word 37 left byte is the control flag byte where:

Bit 0 = line in use meaning something was keyed in the line
Bit 1 = start of chapter key was pressed
Bit 2 = start of paragraph or beginning of paragraph
Bit 3 = tabs were used in the line to position the cursor.
Bit 4 = top of form or new page flag
Bit 5 = auto tabs were used in the line to position the cursor.
Bit 6 = line is to be centered
Bit 7 = change made in the line since the last time the change flags were cleared by the LINES command.

Word 37 center and right byte in the first three lines of the sector represent the foreword, present and back pointers for the sector.

CREATING PASSWORD TEXT AREAS

The 'P' parameter in PRETXT will set the password on the text area. See the section on JCL for all the options on running PRETXT. If no parameter is given zeros will be put in the +6 word in the IDOS directory. If a parameter is given the following algorithum is used to 'hash' the two words (representing six ASCII characters) into a single word in the directory:

A word of all blanks (010020040) is subtracted from the first word (i.e. the first three characters) of the desired password. The result of the subtraction is exclusive or'ed with the second word of the password. The result is then stored in the +6 word of the IDOS directory for that text area.
Example:

/P=4PHASE

4PH = 015050110   ASE = 020251505

015050110	First three ASCII characters
010020040 -	Subtract all blanks
005030050	
020251505 XOR	Exclusive Or
025261555	Result - hashed password for directory

System Text Area - WRDFIL

(1) Modifying system display

The system display (ForeWord and the release/date) can be modified by accessing the system text area and rekeying the display. To access WRDFIL first press PASSW and key in '4PHASE'. Now type in 'WRDFIL' and press INDEX, all of the system display formats are now available. Make a copy of the IMODE document in case you screw up.

(2) Modifying system defaults

Archive Area
Queues

FLAG BYTE DESCRIPTION

T	-	A capital 'T' in the IDOS directory designates that the file is a ForeWord text file. It may or may not contain a header in the IDOS area depending on whether or not it had a header in the text area. The records are compressed and are 96 characters long (in order to contain the tabs, flags, and pointers for each line). Because the information is bit oriented in a ForeWord text file it must be considered as binary information in all use of the file while in the IDOS area, i.e. it is not ASCII data.
t	-	A small unshifted 't' in the IDOS directory designates that the file is a ForeWord Text Area. The flag byte is checked whenever a text area is opened from a terminal to allow only contigueous text area files to be opened. Several of the statistics programs use this flag byte also to locate all the text areas.
R	-	A capital 'R' in the IDOS directory designates that the file is a read only ForeWord text file. It must have a password entered in the password field and therefore cannot be altered or deleted without knowing the password. The document can be opened or copied by any terminal but cannot be altered. This allows for 'standard' paragraphs, glossaries, or fill in formats to be created for all to use but only the originator can change the document.

The flag byte in the header of a document is used to control the type of transfer which will take place when the document is being archived. A capital 'T' or 'R' indicates it is to be transfered as a text document and to retain all information on each line to insure proper editing when brought back into a text area. Any other character will cause the document to be transfered to archive as 80 character records and all nulls in the line are converted to spaces. Lower to upper case conversion does not take place. Lower to upper case conversion can be accomplished by using the offline IDOS processor XTXFIL.

The following file nameing conventions are generally observed in the ForeWord System:

1. Source Files

```
ST L xxx     where, L = library module
   R               R = resident module
   O               O = overlay module
   U               U = utility module
   X               X = transfer program module
```

2. Relocatable Files

```
RT L xxx     where, L = library module
   R               R = resident module
   O               O = overlay module
   U               U = utility module
   X               X = transfer program module
```

3. Absolute Files

IVWORD - ForeWord System
IVOxxx - ForeWord Overlays

PRETXT - Builds Text Areas
CHKTXT - System Maintenance on Text Areas
XTXFIL - Offline utility to transfer to/from ARCHIVE
TBLEDT - Builds individual keyboard tables during WRDGEN and the level 7 branch table
TBLIDC - Builds CTLBS (keyboard and 8121 conversion table source)

4. Other files and their descriptions

IVBASE - Output of virtual table from LOADER
IVQUEx - Queues for use in background processing
STATS - Control file used to execute statistics programs
IVSTAT - File used to record statistics
SVSTAT - Renamed IVSTAT to turn off logging of statistics
Txxxx - Standard keyboard table where xxxx is the keyboard number
Cxxxx - Change file for keyboard table used in TBLEDT
WRDGEN - Control file to execute ForeWord Generation Program (ATUGEN)
TXGDF - GENCF file used during WRDGEN
TXTLIB - ForeWord Library
WRDFIL - System Display File
CKPT - Dump file

CONFIGURATION FILE

SECTOR 0 - SYSTEM PARAMETERS.

WORD	CONTENTS
0-1	System name
2-13	System location
14	Number of background terminals
15	Number of additional queues
16	System configured to create random document name
17-20	Cursor position in command menu
21	System will display header when document is opened
22	Number of line printers
23	Number of VDU's (screens)
24	System memory size
25	Time for automatic backup
26	System optioned for communications
27	System type: (A=IV70/90, B=IV90/MOD2, C=MFE)
30	System optioned for 8230 disc type
31	System optioned for 8240 disc type
32	System optioned for 8260 disc type
33	System optioned for 8270 disc type
34	System optioned for DEBUG
35	System optioned for line printer dump
36-37	Supervisor password
40-41	Keyboard-type 1
42-43	Keyboard-type 2
44	Line printer type
45-46	Communications log document name
47-50	Text area name containing communications log document
51	Disc drive number for text area containing comm log document
52	Comunications controller number
53	Queue length in sectors

SECTOR 1 - TERMINAL PARAMETERS.

WORD	CONTENTS
00-06	Terminal parameters for terminal 0 where: 0-1 Default text area name for terminal 0 2 Default drive number for text area 3 Logical printer number 4 Printer type number 5 8121 physical printer number 6 Terminal characteristics Byte 0 = Read only terminal Byte 1 = Keyboard number Byte 2 = Keyboard type
07-15	Terminal parameters for terminal 1
16-24	Terminal parameters for terminal 2
25-32	Terminal parameters for terminal 3
34-61	Terminal parameters for terminal 4
43-73	Terminal parameters for terminal 5
52-60	Terminal parameters for terminal 6
61-67	Terminal parameters for terminal 7
70-76	Terminal parameters for terminal 8
77-105	Terminal parameters for terminal 9
106-114	Terminal parameters for terminal 10
115-123	Terminal parameters for terminal 11
124-132	Terminal parameters for terminal 12
133-141	Terminal parameters for terminal 13
142-150	Terminal parameters for terminal 14
151-157	Terminal parameters for terminal 15
160-166	Terminal parameters for terminal 16
167-175	Terminal parameters for terminal 17
176-204	Terminal parameters for terminal 18
205-213	Terminal parameters for terminal 19
214-222	Terminal parameters for terminal 20
223-231	Terminal parameters for terminal 21
232-240	Terminal parameters for terminal 22
241-247	Terminal parameters for terminal 23

SECTOR 3 - COMMUNICATION PARAMETERS.

WORD	CONTENTS
0-13	Communication parameters for location 0 where: 0-1 System location name 0 2-3 SYSGN file name for location 0 4-11 Location 0 phone number 12 Location 0 is a remote location 13 Data type 14-17 Unused
20-37	Communication parameters for location 1
40-57	Communication parameters for location 2
60-77	Communication parameters for location 3
100-117	Communication parameters for location 4
120-137	Communication parameters for location 5
140-157	Communication parameters for location 6
160-177	Communication parameters for location 7
200-217	Communication parameters for location 8
220-237	Communication parameters for location 9
240-257	Communication parameters for location 10
260-277	Communication parameters for location 11
300-317	Communication parameters for location 12
320-337	Communication parameters for location 13
340-357	Communication parameters for location 14
360-377	Communication parameters for location 15

TBLIDC

Overview:

The TBLIDC program uses the following JCL:

```
// TBLIDC
/0=CTBLS            OUTPUT DCN KEYBOARD/8121 TABLE
/3=TEMP1            INPUT KEYBOARD TABLE FOR LEFT BYTE FROM TBLEDT
/4=T8121            INPUT STANDARD 8121 CONVERSION TABLE
/5=TEMP2            INPUT KEYBOARD TABLE FOR RIGHT BYTE FROM TBLEDT
//
```

This program takes the two keyboard tables produced from TBLEDT during WRDGEN and the 8121 printer table and builds a DCN table used within ForeWord for conversion of input keystrokes and output bytes to the 8121 printer. The left byte is considered primary keyboard, the center byte is the 8121 printer and the right byte is the secondary keyboard. The output from the program is assembly code and is SNEDITed and assembled into the relocatable CTBLR for inclusion in the load step.

```
*******************************************************************************
Alphanumeric Style

000001002003004005006007010011012013014015016017  7200 KEYBOARD
020021022023024025026027030031032033074035076037      020
040041042043044045046047050051052053054055056057      040
060061062063064065066067070071072073054075056077      060
100101102103104105106107110111112113114115116117      100
120121122123124125126127130131132133134137136137      120
140141142143144145146147150151152153154155156157      140
160161162163164165166167170171172173174175300301      160
302303304305134307310311312313314315137317317304      200
303323324325326327330331332333334335336337340174      220
040241                                                240
260135262263264265266267173175                        260
      040                                             300
                                                      320
040                              353   355356         340
060061062063064065066067070 342343344 360 000016      360


*******************************************************************************
DATA IV Keypunch Style

230101002103104105106107110111012013114115016117     029 STYLE KYBRD
120121122123124135126023130131132210074226076056 020     UPPER CASE
040224043054044056221220055060231210054226056056 040     ONLY (7202)
225224100045052074221220055057231210070230071056 060
230   041042072051073016047062064065066067050063 100
046053003076174061075137077135   211227226211227 120
225101102103104105106107110111112113114115116117 140
120121122123124125126127130131132211227231201202 160
201202203204205206204242242210201210205212212203 200
202207215216217221220222223224213214203241204205 220
040200                                           240
225224100045052074221220055037                   260
      203                                        300
                                                 320
040                              200   240237    340
236233234235230231232225226227      242212210 360
```

```
*******************************************************************************
DATA IV Typewriter Style

000101002003104105106107110111012013114115016117     7203 KEYBOARD
120121122123124125126127130131132053074237076037     020  DATA IV
040041042043044045046047050051052053054055056057     040  TYPEWRITER
060061062063064065066067070071072073054075056077     060
100101102103104105106107110111112113114115116117     100
120121122123124125126127130131132211053137211137     120
140141142143144145146147150151152153154155156157     140
160161162163164165166167170171172211240052201202     160
201202203204205206204242242210201210205212212203     200
202207215216217221220222223224213214203241204205     220
040200                                               240
236135226227230231232233173175                       260
      040                                            300
                                                     320
040                              053   055056        340 NOT PF X
060061062063064065066067070071         242212210     360 ISLAND
```

```
*******************************************************************************
2260 Data Entry Style

230101002103104105106107110111012013114115016117     2260 DATA ENTRY
120121122123124125126127130131132210074226076056  020     UPPER CASE
040224043054044056053220137060231210054226056072  040     ONLY. (7204)
225224100045052075053220055057231210070230071073  060
230050102103077135106107110062064065066115116063  100
046074122051124061126076130131132211227226211227  120
225101102103104105106107110111112113114115116117  140
120121122123124125126127130131132211227231201202  160
201202203204205206204242242210201210205212212203  200
202207215216217221220222223224213214203241204205  220
040200                                            240
225224100045052000221220023037         000        260
      203                                         300
                                                  320
040                              200   240237     340
236233234235230231232225226227         242212210  360
```

```
*******************************************************************************
NTP/100, 150 200 (3270) Keypunch Style

230101002103104105106107110111012013114115016117     029 STYLE KYBRD
120121122123124135126023130131132210074226076056 020     UPPER CASE
040224043054044056221220055060231210054226056056 040     ONLY (7225)
225224100045052074221220055057231210070230071056 060
230   041042072051073016047062064065066067050063 100
046053003076174061075137077135   211227226211227 120
225101102103104105106107110111112113114115116117 140
120121122123124125126127130131132211227231201202 160
201202203204205206204242242210201210205212212203 200
202207215216217221220222223224213214203241204205 220
040200                                           240
225224100045052074221220055037                   260
      203                                        300
                                                 320
040                              200   240237    340
236233234235230231232225226227         242212210 360
```

```
*******************************************************************************
NTP/100, 150, 200 (3270) EBCDIC Typewriter Style

174101002103104105106107110111012013114115016117    EBCDIC KB TO INT
120121122123124125126127130131132072074023076037       020
040135100043044045003046052050047073054055056057       040
060061062063064065066067070071042072054016056077       060
041101102103104105106107110111112113114115116117       100
120121122123124125126127130131132211075137211053       120
051141142143144145146147150151152153154155156157       140
160161162163164165166167170171172211053042201202       160
201202203204205206204242242210201210205212212203       200
202207215216217221220222223224213214203241204205       220
040200                                                 240
175135100043044045003046052173                         260
      040                                              300
                                                       320
040                              200   240237          340
236233234235230231232225226227         242212210       360


*******************************************************************************
NTP/100, 150, 250 (3270) Data Entry Style

230101002103104105106107110111012013114115016117  000 DATA ENTRY KYBRD
120121122123124135126023130131132210074226076056  020     UPPER CASE
040224100043054056221220055060231210054226056056  040     ONLY, PF I.
225224100043052044221220055057231210070230071056  060     (7228)
230074050047104041135016003062064065066067051063  100
046053072076073061042137077075045211227226211227  120
225101102103104105106107110111112113114115116117  140
120121122123124125126127130131132211227231201202  160
201202203204205206204242242210201210205212212203  200
202207215216217221220222223224213214203241204205  220
040200                                            240
225224100043052044221220055037                    260
      203                                         300
                                                  320
040                              200   240237     340
236233234235230231232225226227         242212210  360

NTP/100, 150, 250 (3270) ASCII Typewriter Style


*******************************************************************************
174001002003004005006007010011012013014015016017  7227 KEYBOARD
020021022203502402502602703003103207207402307603       020
040041100043044045026046052050047073054055056057       040
060061062063064065066067070071042072054021056077       060
027101102103104105106107110111112113114115116117       100
120121122123124125126127130131132211040137211053       120
051141142143144145146147150151152153154155156157       140
160161162163164165166167170171172211053042             160
            075                     053                200
                                                       220
240                                                    240
175135100043003045016046052173                         260
      040                                              300
                                                       320
040                                                    340
                                                       360
```

```
*******************************************************************************
ForeWord Standard Keyboard

174001002003004005006007020011012013011015016017  7249 KEYBOARD
020022022035024025026027030017032134074023076037       020
040041100043044045003046052050047073054055056057       040
060061062063064065066067070071042072054021056077       060
027101102103104105106107110111112113114115116117       100
120121122123124125126127130131132211040137211053       120
051141142143144145146147150151152153154155156157       140
160161162163164165166167170171172211053133             160
            075                     053                200
                                                       220
240                                                    240
175135136033014045016046176173                         260
      040                                              300
                                                       320
040                                                    340
                                                       360


*******************************************************************************
ForeWord DEVORAK Keyboard

033001030011010076025011004003010024016015002022  DEVORAK KEYBOARD
014037020020031007013074021006042023027174026032       020
040041100043044045003046052050055163167027166172       040
060061062063064065066067070071137123127073126132       060
072101130112105056125111104103110124116115102122       100
114077120117131107113054121106042211075021111053       120
051141170152145056165151144143150164156155142162       140
154057160157171147153054161146047211053042             160
            075                     053                200
                                                       220
240                                                    240
175135174043003045016046176173                         260
      040                                              300
                                                       320
040                                                    340
                                                       360


*******************************************************************************
8121 Character Printer Table

157      040            010   012013014010177       8121 PRINTER
   133   023         135               035   134      020
   041042043044045046047050051052053054055056057      040
060061062063064065066067070071072073074075076077      060
100101102103104105106107110111112113114115116117      100
120121122123124125126127130131132133134174136137      120
140141142143144145146147150151152153154155156157      140
160161162163164165166167170171172173174175176177      160
157      040                  012013      177         200
   133   023         135               035   134      220
   041042043044045046047050051052053054055056057      240
060061062063064065066067070071072073074075076077      260
100101102103104105106107110111112113114115116117      300
120121122123124125126127130131132133134174136137      320
 40141142143144145146147150151152153154155156157      340
160161162163164165166167170171172173174175176177      360
*******************************************************************************
```

PROCESSOR JCL FOR FOREWORD

PASSW

The following is the JCL used to change the password on the customers pack. The JCL is contained in the control file PASSW:

```
// DIRMOD
/ I=WRDCFG,Q.     Unprotect the configuration file for update

/ I=TEMP01,O,Q. Delete the file called TEMP01 just in case it is there.
// ATUGEN         Run the absolute program
/ A=WRDCFG        The name of the configuration file to be updated
/ B=STUGN1        Input cheater file. Not used but required for progr
/ C=TEMP01        Output file name. Is not generated in a password ru
/ Q=STUGNQ        Input question file.
// CBLERR
P                 Password run only. Ask no configuration questions.
//
```

STATS (STATISTICS)

The following is a copy of the STATS control file and an explainati of the JCL.

```
// GENCTR                 Use the processor GENCTR to look for all IDOS
/O=TEMP,E=/*,H=*,G=t. directory entries with a 't' as the flag byte
// ATUST1                  identifing the file as a text area. Create the
/ A=\.                     JCL to run COBOL program against each text area
//                         and print out the headers.
/*
// TEMP                   Execute the control file created above.
// ATUST2                 This program will list and optionally update the
/ A=IVSTAT.                terminal and printer stats which are accumulated
//                         in the file IVSTAT.
// GENCTR                 Use the processor GENCTR to look for all IDOS
/O=TEMP,E=/*,H=*,G=T. entries with a 'T' as the flag byte identifing
\.                         the file as an archived text file.  Build a temp
/*                         file to be read by the next COBOL program.
// ATUST3                 This program will read in the list generated
/ A=NAME, B=TEMP.          in the above step and open each one to access
//                         the header and print the report.
/*
// ATUST5                 Print ASSEMBLY statistics
/ A=PASTAT
//
/*
// ATUST6                 Generates files for the next COBOL program for
/ A=PASTAT, B=TEMP.        special ASSEMBLY statistics.
//
// SORT                   SORT the temporary file
/ I=TEMP, O=TEMP01, M=2000, F=29, L=45
//
// ATUST7                 Print summary of the ASSEMBLE statistics
/ A=TEMP01
/*
// ATUST8                 Deletes documents based on user criteria
/ A=TXAREA
/*
//
```

PRETXT (CREATING TEXT AREAS)

The JCL for running PRETXT is as follows:

```
// PRETXT
/T=NAME OF THE TEXT AREA TO BE CREATED
/P=NAME OF THE PASSWORD TO BE GIVEN TO THE TEXT AREA
/S=NUMBER OF SECTORS TO BE CREATED @ DRIVE NUMBER FOR TEXT AREA.
//
```

The default text area name is TXAREA and the default number of sectors is 03000(1536).

CHKTXT (TEXT AREA MAINTENANCE PROGRAM)

The JCL for running CHKTXT and the options allowed when an error is detected are as follows:

```
// CHKTXT
/T=NAME OF THE TEXT AREA TO BE CHECKED.
/A TO CAUSE ALL SECTORS TO BE SCANNED IN THE EVENT OF AN ERROR INSTEAD
   OF JUST THE ALLOCATED SECTORS.
/S TO CAUSE CHKTXT TO SUPPRESS SCAN IN THE EVENT OF AN ERROR. NO
   ATTEMPT WILL BE MADE TO CORRECT THE ERROR.
/L TO CAUSE CHKTXT TO PRINT A LISTING OF ALL RECLAIMED SECTOR
   ADDRESSES ON THE PRINTER.
//
```

When an error is detected the following options are allowed:

L or Y	Link. The sector shown on the bottom third of the screen is to be linked with the sector show in the center of the screen.
N	Do not link the sectors as shown on the screen. Continue to scan the text area for other possible matches if the option 'S' was not given above.
B	Follow the back pointer in the sector on the bottom third of the screen. Used in an attempt to identify and relink the document.
F	Follow the foreword pointer in the sector on the bottom third of the screen.
C	Continue with the next document in the text area. No changes will be made in the current document and chain linkage errors will still be in the text area. CHKTXT must be rerun to insure all errors are corrected.
R	Reset the screen to the original sector in error.
E	Follow the pointer in the directory for the document to the ending sector and display it on the bottom third of the screen.
T	Indicates that the chain is to be trunctated. The sector in the middle of the screen will be the last sector of the document.

XTXFIL (TEXT AREA TRANSFER UTILITY)

The following JCL is used with XTXFIL to transfer documents to/from text areas:

```
// XTXFIL
/I=NAME OF FILE IN IDOS TO BE TRANSFERED INTO THE TEXT AREA @ DRIVE #.
/N=NAME OF THE TEXT AREA TO BE TRANSFERED INTO @ DRIVE #.
/P=SET A PARAGRAPH FLAG ON EVERY LINE (USED ON NON-TEXT DOCUMENTS)
/T=COLUMN NUMBER IN WHICH TO SET A TAB (MULTIPLE 'T' INPUTS OK)

/O=NAME OF THE FILE IN THE TEXT AREA TO BE TRANSFERED TO IDOS @ DRIVE
   IF THE NAME IS LEFT BLANK (I.E. O=@1.) THEN THE ENTIRE TEXT AREA
   WILL BE TRANSFERRED.
/N=NAME OF THE TEXT AREA TO BE TRANSFERED FROM @ DRIVE #.
//
```

Be aware the I paramater and the O parameter are mutually exclusive in each option entry.

TAKING A DUMP

A. Insure the words 'CKPT TAKEN' are not already on screen 0 of the system. This indicates that ForeWord detected an error and automatically took a CKPT. In this case all that is required is to print or copy the CKPT file.

B. If the words do not appear proceed with the dump of memory.

C. Ensure the disc pack in drive zero is mounted and ready.

D. Set the AUTO/MANUAL switch to NAMUAL.

E. Set the console keys to 70000003.

F. Set all the DISPLAY SELECT switches down (select TIR).

G. Press the SYSTEM RESET switch.

H. Press the STEP switch.

I. Press the LOAD switch.

J. Set the AUTO/MANUAL switch to AUTO.

PRINTING A DUMP

A. Type // CKPT to load the image of the checkpoint into RAM for printing or if the file has been renamed use the new name.

B. Set AUTO/MANUAL switch to MANUAL.

C. Set the CONSOLE KEYS to 71100001.

D. Set all the DISPLAY SELECT switches down (select TIR)

E. Press RESET.

F. Press STEP.

G. Press LOAD.

H. Set the AUTO/MANUAL switch to AUTO.

I. When the processor halts, set AUTO/MANUAL switch to MANUAL.

J. If you have an 8135 printer set all console keys down, for all other printer set console keys 21 and 23 up and all others down.

K. Set the DISPLAY SELECT switches to 001 (select RA).

L. Press LOAD.

M. Set all the DISPLAY SELECT switches down (select TIR).

N. Set AUTO/MANUAL switch to AUTO.

SYSTEM MEMORY ORGANIZATION

FOREWORD RELEASE AY06

SAMPLE MEMORY LAYOUT

MFE/IV ROUTINES AND TABLES	2 PAGES
LOGICAL SCREEN AREA	2 PAGES
LOGICAL USER TABLE PAGE	1 PAGE
LOGICAL OVERLAY AREA	1 PAGE
LOGICAL TRANSFER PROGRAM AREA (PAGE 1 OF 3) ALSO MATH PACK LOGICAL ACCUM PAGE	1 PAGE
LOGICAL TRANSFER PROGRAM AREA (PAGE 2 OF 3) ALSO LOGICAL DOCUMENT BUFFER PAGE	1 PAGE
LOGICAL TRANSFER PROGRAM AREA (PAGE 3 OF 3)	1 PAGE
FOREWORD SYSTEM TABLES	
FOREWORD RESIDENT ROUTINES	
FOREWORD LIBRARY ROUTINES	
MFE/IV LIBRARY ROUTINES	

PHYSICAL MEMORY REQUIREMENTS

FOREWORD RELEASE AY06

The minimum amount of memory required is 192K. To estimate the physical requirements of an AY06 ForeWord System proceed as follows:

```
        No.of physical pages = 28 pages for resident code
                             + no. of tubes and background tasks x 1/4 page
                               for user tables
                             + 1 page for each background terminal
                             + 1 page for each wide document buffer requested
                             + 1 page for Math package
```

The above equation will determine the number of physical pages Foreword will require. After that total is obained 8 pages must be added for MFE and 1 page for each terminal hooked up the system must be added.

USER TABLE DEFINITIONS

FOREWORD RELEASE AY06

WORD	LABEL (REDEFINED)	DESCRIPTION
000	KEYRTN	Addr of non-standard keystroke routine
001	ECHAR	Pointer to top of keystroke queue
	(KPTOP)	
	(BCKACT)	Background task is active
002	KEYPNT	Pointer to bottom of keystroke queue
	(QUE)	Address of address of queue
003	FWA	First word address of screen
004	KTYPE	Keyboard type (RCPY intruction)
005	CASE	Upper case flag
006	STX2	Save index register 2 for status test
007	IOX2	Save index register 2 for disc routines
010	LOCK	Lock keyboard
011	Unused	
012	LKEYS	Legal key bit flags
013	LKEYS1	
014	NAME0	First three characters of document name
015	NAME1	Last three characters of document name
016	LSECTR	Ending sector of document
017	DIRFLG	Not used (Reserved)
020	NSECTR	Number of sectors in document
021	SSECTR	Starting sectory of document
022	PSSWRD	Password
023	HSECTR	Header sector address
024	CURS	Character under cursor
025	CURCHR	Cursor character
026	CURC	Cursor blink rate
027	CLSW	Character/cursor switch
030	CHAIN1	Holds address of requested task
031	TASK1	Contains parameter passed via task or task8
032	CHAIN2	Address of next requested operation
033	TASK2	TASK2
034	CHAIN#	Chain "address" of func. that called "TASK9"
035	TASK#	Return address for chained functions
036	WAIT	Wait for routine instruction
037	ABRTA	Abort address
040	MD2WN	IV90-2 window register
041	WRTFLG	Write screen flag
042	URA	Save RA
043	URB	Save RB
044	U23	Save X2,X3
046	TSKADR	Task address to branch to after time sharing
047	DRQPRI	Disc request queue priority
050	DSTAT	Request table disc
051	DEVS	Request table disc
052	RAM	Request table disc
053	SECT	Request table disc
054		Request table disc
055	IMODE	IMODE mode switch
056	CRSCNT	Cursor column number
057	Unused	

USER TABLE DEFINITIONS - CONTINUED

060	CURSCH	Cursor position
061	CURSWD	CURSCH+1
062	LMARG	Left margin col. counter
063	RMARG	Right margin col. counter
064	TXNAME	Name of text area
065	TXNAM1	TXNAME+1
066	TXFWA	FWA of text area
067	TXLWA	LWA of text area
070	NAMEA	Second document name
	(SVDNAM)	Temp storage for document name
071	NAMEA1	
	(SVDNM1)	
072	TXNMA	Second text area name
	(SVTNAM)	Temp storage for text area name
073	TXNMA1	First sector address of glossary
	(SVTNM1)	
074	DRVA	Second drive
075	GLSFWA	First sector addr of glossary
076	GLRDSC	Glossary disc # & glossary active flag (1B0)
077	GLDXFW	FWA of glossary text area
100	GLSDOC	Glossary document name (First three letters)
101		Glossary document name (Second three letters)
102	GLSTXT	Glossary text area name (First three letters)
103		Glossary text area name (Second three letters)
104	TABMK	Tab marker
105	AUTAB	Auto tab to column store in here
106	DECTAB	Decimal tab active
107	CENTR	Center line
110	FLG	Cursor return flag
111	PTRFLG	
112	STRTCH	STRTCH
113	STRTWD	STRTCH+1
114	OLDSCH	OLDSCH points to char in the bottom (old)
	(TABTYP)	Type of tab (Column manipulation)
115	OLDWRD	Sectr in edit code used by GETOLD to get char
	(TAB)	Used for TABTYP (Column manipulation)
116	NEWSCH	Points to char in the top (new)
	(LCOL)	Beg ptr col # (Column manipulation)
117	NEWWRD	Sectr in edit code used by GETNEW to get char
	CURSEC	Sec # of cursor at function start
120	RHTLWC	Contains LCR WRD pair for the right bracket
121	RHTLWD	RHTLWC+1
122	LFTLWC	Contains LCR WRD pair for left pointer
	(FNWSEC)	Saves footnote start sector on wrap (Format)
123	LFTLWD	LFTLWC+1
	(FNFLAG)	Footnote flags (Format)
124	UPSCH	UPSCH
	(RCOL)	End ptr Col # (Column Manipulation)
125	UPWRD	UPSCH+1
	(WCOL)	Column width (Column Manipulation)
126	PLCH	PLCH
127	PLCH2	PLCH+1

USER TABLE DEFINITIONS - CONTINUED

130	BPO	Back sector pointer
131	PPN	Present sector pointer
132	FPO	Front sector pointer
133	LFTSEC	Address of sector containing left pointer
	(FNSSEC)	Starting sector of the footnote file (Format)
134	NEWCNT	NEWCNT
	(ERRCOD)	Error message code number (Column Manipulation)
135	THISEC	THISEC
136	RHTSEC	Address of sector containing right pointer
137	STRSEC	STRSEC
	(TPELBA)	Str ba of line with beg ptr in top
140	OLDCNT	Col count of line in botm sectr in edit code
	(DEC)	Used for TABTYP (Column manipulation)
141	FLX2	FLX2
142	FLCH	Contains char returned by GETOLD or GETNEW
143	CHPRG	CHPRG
144	RMRGN	Column count of last char in line
145	OLDDSC	Old disc
146	NEWDSC	New disc (insert)
147	NEWFWA	New txarea address (insert)
150	CMODE	Command mode
151	CMDCD	Points to the code to be executed in command mode
152	CMDSCH	Used during a command to save the cursor loc
	(PAGNUM)	Page number (Format)
153	CMDSWD	When used by Format code 2nd wrd of CMDSCH
	(FNNUM1)	First word of footnote number (Format)
154	CMDSW1	Double word footnote # (must follow CMDSWD)
	(FNNUM2)	Second of footnote number (Format)
155	PPCMD	Current sector save
156	CDSRET	Return address
157	DIRFWA	FWA save cell for TDRFND
160	PRBRA	Address of print routine
161	TOPSKP	TOPSKP
162	BODY	BODY
163	LNSPG	Lines/page
164	PRSW	Printer switch-status for print overlay
165	PRCNT	Printer line count-lines per page
166	PRFLG	Line printed flag
167	PRSTP	If NZ,then stop printing after a page
170	PRSPC	Printer line spacing count
171	LNSPAC	Default line spacing
173	PREQ	Printer request table
	(PRTYPE)	
174	PRSTAT	
175	PRFWA	
176	PRLMRG	
177	PRUNDR	
200	PRVERT	
201	PRCU	Print continuous underscore flag
	(MDELBA)	Str ba of line with end ptr in middle (Column manip)
	(HDOCNT)	# of lines for odd header (Format)

SER TABLE DEFINITIONS - CONTINUED

202	BCU	Begin continuous underscore
	(BTELBA)	Str ba of line with end ptr in bottom (Column manip)
	(HDECNT)	# of liines for even header (Format)
203	ECU	End continuous underscore
	(TEDPBA)	Top sec ba of end ptr (Column manipulation)
	(FTOCNT)	Starting sector address of odd footer (Format)
204	SPRUND	Underscore override cont. underscore
	(MEDPBA)	Mid sec ba of end ptr (Column manipulation)
	(FTECNT)	# of lines for even footer (Format)
205	SAVBCU	Save area for BCU
	(BEDPBA)	Bot sec ba of end ptr (Column manipulation)
	(LINCNT)	# of lines of text for mult page footnote (Format)
206	SAVECU	Save area for ECU
	(TTBBTA)	Top sec tab bit addr beg ptr line (Column manip)
	(FNP1ST)	First footnote of page flag (Format)
207	SSPRND	Save area for SPRUND
	(PRSCNT)	Processing new page flag (Format)
210	LFLCH	LFLCH
211	NPLSW	NPLSW
212	WPLSW	WPLSW
213	FPRUDR	Save area for PRUNDR (Form letter merge)
	(BTBBTA)	Bot sec tab bit addr beg ptr line (Column manip)
	(FNMULT)	Multiple page footnote flag (Format)
214	FBCU	Save area for BCU (Form letter merge)
	(TCNBTA)	Top sec control flag bit addr beg line (Column manip)
	(HDOSEC)	Starting sector address of the odd header (Format)
215	FECU	Save area for ECU (Form letter merge)
	(MCNBTA)	Mid sec control flag bit addr beg line (Column manip)
	(HDESEC)	Starting sector address of even header (Format)
216	FPRCU	Save area for PRCU (Form letter merge)
	(FTOSEC)	Starting sector address of odd footer (Format)
217	CXFLAG	Set separator option flag (Form Letter)
	(TLMBA)	Top sec lm ba on beg ptr line (Column manipulation)
	(FTESEC)	Starting sector address of even footer (Format)
220	TBUF1	Wrap around buffer
	(TOPSBA)	Start top sec byte addr (Column manipulation)
	(REQWID)	Number of requested wide pages (Allocate Command)
221	TBUF2	TBUF+1
	(MIDSBA)	Start mid sec ba (Column manipulation)
	(REQOVL)	Number of requested overlay pages (Allocate Command)
222	TBUF3	TBUF+2
	(BOTSBA)	Start bottom sec ba (Column manipulation)
	(REQTOT)	Number of requested total pages (Allocate Command)
223	TBUF4	TBUF1+3
	(OFFSBA)	Start offscreen sec ba (Column manipulation)
	(UPDCFG)	Response to update the WRDCFG file (Allocate Command)
224	TBUF5	TBUF+4
	(TOPLBA)	Str ba of line with beg ptr in top (Column manip)
	(USEXFR)	Response to use the XFER area pages (Allocate command)
225	TBUF6	TBUF+5
	(MIDLBA)	Str ba of line with beg ptr in middle (Column manip)
226	TBUF7	TBUF+6
	(BOTLBA)	Str ba of line with beg ptr in bottom (Column manip)

USER TABLE DEFINITIONS - CONTINUED

227	TBUF8	TBUF+7
	(TBGPBA)	Top sec ba of beg ptr (Column manipulation)
230	TBUF9	TBUF+8
	(MBGPBA)	Mid sec ba of bg ptr (Column manipulation)
231	TBUF10	TBUF+9
	(BBGPBA)	Bot sec ba of beg ptr (Column manipulation)
232	PLCNT	PLCNT
233	RPCH	Contains first char of search string
234	RPCNTR	Used as loop counter in Search & Replace
	(MLMBA)	Mid sec lm ba on beg ptr line (Column manipulation)
235	RPCNT	Contains # of chars in search string
	(TRMBA)	Top sec lm ba on beg ptr line (Column manipulation)
236	LBUF1	Address of first phrase
237	FNORPH	Footnote counter to stop single line fns (Format)
240	RPLSCH	Contains wrd pair pointing to char of search string
	(MRMBA)	Mid sec rm ba on beg ptr line (Column manipulation
241	(BCNBTA)	Bottom sec control flag on beg ptr line (Column manip)
242	RPLCNT	Contains # of characters in replace string
	(SAVCUR)	Save cursor sec for screen restore (Column manip)
243	RPLSAV	Used to save RMRGN during Search & Replace
	(ALTFLG)	Alternating heading/footing page flag (Format)
244	RPLSPL	Special character treatment
	(ALTODD)	Reset to odd header/footer page flag (Format)
245	DOCTIM	Start time document was opened
246	DOCLIN	Lines printed in the document
247	KCOUNT	Keystroke counter
250	KMEM	Keystroke memory flag and sector address
251	KMEM2	
252	KEYLOG	Keystroke log - last 3 keys hit
254	KMEMCH	Keystroke memory LCR pointer to key list
255	KMEMWD	KMEMCH+1
256	SLNSPG	Lines per page for single sheet feeder
257	SSPLMR	Left margin offset for single sheet feeder
260	GRPTOT	Total # of groups in group document
261	NXTGRP	1st grp to display on next page in showall groups mod
	(FNCNTR)	# of footnotes (Format)
262	NONDOC	Contains name of non-immediate document being used currently by message feature
	(FNBFSS)	Starting sector of /BF footnote file (Format)
263	OVPAGE	Pointer to ????? indicating overlay is running in a paged overlay area
264	DOCNM1	Document name
266	DTFWA1	Document's text area first sector address
267	DTDRV1	Document's text area drive
270	DOCNM2	Document name
272	DTFWA2	Document's text area first sector address
273	DTDRV2	Document's text area drive
274	TXFWA1	Text area's first sector address
275	TXLWA1	Text area's last sector address
276	HDFWA1	Doc header's area first sector address
277	TXFWA2	Text area's first sector address

USER TABLE DEFINITIONS - CONTINUED

300	TXLWA2	Text area's last sector address
301	BDSTAT	Broadcast request table disc
302	BRDEVS	Broadcast request table disc
303	BRRAM	Broadcast request table disc
304	BRSECT	Broadcast request table disc
306	WFWA	Wide 1st wrd address
307	DOCTYP	Indicates doc type=FWA when executing wide doc code
310	DTYPE	Also indicates document type
311	SEGMNT	Segment # for wide documents
312	MAXSEG	Greatest segment allowed in current document
314	KURSCH	Video cursor position for wide docs
315	KURSWD	KURSCH+1
316	KRSCNT	Video column counter for wide docs
317	SVFWA	
320	SVFW1	Saves FWA when processing des using offscreen
	(FNUNLN)	Continued footnote underline flag (Format)
322	UDUMY1	
323	MINLIM	
	(FNQUEU)	Number of lines in footnote backlog (Format)
324	MAXLIM	Format expects even word boundary
	(FNBFCH)	LCR pair for pointers into /BF footnotes (Format)
325	PREKEY	
	(FNBFWD)	Starting sector of /BF footnote file (Format)
326	SECPOS	
	(FNBFPS)	Current sector of pointers into /BF footnotes (Format)
327	UDUMY2	
330	DTEMP	Format expects even word boundary
	(LBSIZB)	Line byte size (Column manipulation)
331	DTEMP1	DTEMP1+1
	(LBTSZB)	Bot sec line bit size (Column manipulation)
332	TEMP	
	(CDPTYP)	Convert PTRFLG (Column manipulation)
333	WALLS1	Adrs of first sector - narrow or wide
	(WALLS)	First address of sector allocation table
	(CFTYPE)	(Column manipulation)
	(FNC1)	Current output footnote number (Format)
334	WALLS2	Adrs of second sector - narrow or wide
	(COFFBT)	(Column manipulation)
	(FNC2)	...for continued footnote (Format)
335	WALLS3	Adrs of third sector - narrow or wide
	(LBSIZE)	(Column manipulation)
	(FTFLSE)	Temporary address of even footer file (Format)
336	WALLS4	Adrs of fourth sector - wide
	(LBTSIZ)	(Column manipulation)
	(FTFLSO)	Temporary address of odd footer file (Format)
337	WALLS5	Adrs of first header sector
	(LMOFFB)	(Column manipulation)
	(FNUNDR)	Broken footnote while underlining (Format)
340	WALLS6	Adrs of second header sector
	(RMOFFB)	(Column manipulation)
	(FNCSCH)	Character pointers into continued fn (Format)
341	WALLS7	Adrs of third header sector
	(TOFFBT)	(Column manipulation)
	(FNCWRD)	Word address of continued fn

USER TABLE DEFINITIONS - CONTINUED

342	FSPRNT	
343	WDADRS	Adrs of memory document buffer (Format)
345	FTIME	
	(FNCSEC)	Sector address of continued fn (Format)
346	SCRFLG	
350	LOGID	
352	EDTSEC	
353	EDITX2	
354	HOMSEC	Updated on every C/HOME or S/HOME,cleared when a new document is opened
355	XECLV4	
356	MTIMER	Terminal idle time timer
357	FMTFLG	Format flags
360	FNBFCC	Column count of footnote (list)
361	FNCONT	Footnote continued on next page flag
362	FNPSEC	Next available (present) sector address fn
363	RESCNT	Restores a saved column number
364	RESSCH	Restores a saved cursor location
365	RESWRD	RESSCH+1
366	FNPSCH	Next available (present) char address fn
367	FNPWRD	Next available (present) word address fn
370	FTCRET	Saves return address for BAL's
371	ODDEVN	Odd/even header/footer flag
	(SVPWRD)	Saves password of input document (FM1 only)
372	RSM?	Does middle sector need to be restored?
373	MTHFLG	Math flag and options in effect

The length of the user table is defined in the module STRTUB

ALLOCATION SECTOR -- FOREWORD RELEASE AY06

The first sector (sector 0) of the text area is an allocation table that indicates the sectors in use. Each bit in the table corresponds to one sector in the text area. If a bit is set to one, the sector is available. If a sector is in use, the bit is set to zero. With eight words on a line, each line represents 192 sectors (0300 sectors octal). The entire text area is thus represented by an allocation table one sector long.

Words	0000 - 0371	Allocation Bit Map.
Word	0372	Total number of sectors in text area
Word	0373	Number of unused sectors
Word	0374	Number of unused directory entries
Word	0375	Maximum number of directory entries
Word	0376	Text area type

TEXT AREA DIRECTORIES - FORWORD RELEASE AY06

Six sectors (relative 1 thru 7). Each entry is eight words long. Maximum number of entries is 192. The eight word entries appear as follows:

WORD	CONTENTS
0	First three ASCII characters of the document name.
1	Last three characters of the document name.
2	Last sector address relative to the start of the text area.
3	Line position of cursor.
4	Bit 0 indicates a read-only document. Bits 1-7 are the flag byte. Right two bytes contain the sector count of the document.
5	Startng sector address relative to the start of the text area.
6	Hash coded password. Zero if no password.
7	Starting sector of the DESCRIPTION. Zero if no DESCRIPTION.

DATA SECTORS
FOREWORD RELEASE AY06

Sector structure for ForeWord type T documents

	00			32			33			34			35			36			37		
0000						D	T	T	T	T	T	T	T	T	T	T	L	R	C		BP
0040						D	T	T	T	T	T	T	T	T	T	T	L	R	C		PP
0100						D	T	T	T	T	T	T	T	T	T	T	L	R	C		FP
0140						D	T	T	T	T	T	T	T	T	T	T	L	R	C	X	X
0200						D	T	T	T	T	T	T	T	T	T	T	L	R	C	X	X
0240						D	T	T	T	T	T	T	T	T	T	T	L	R	C	X	X
0300						D	T	T	T	T	T	T	T	T	T	T	L	R	C	X	X
0340						D	T	T	T	T	T	T	T	T	T	T	L	R	C	X	X
0340						D	T	T	T	T	T	T	T	T	T	T	L	R	C	X	S
						**															

** - Column 81 indicator column

LEGEND:

T = one byte of tab indicator bits
C = one byte of control bits for the logical line
  Bit 0 = line in use
  Bit 1 = start of chapter key was pressed
  Bit 2 = start of paragraph or beginning of paragraph
  Bit 3 = tabs were used in the line to position the cursor.
  Bit 4 = line is start of new page
  Bit 5 = auto tabs were used in the line to position the cursor.
  Bit 6 = line is centered
  Bit 7 = line has been changed
D = one byte of display character (the character for col 81 display)
S = one byte of "sector type indicator" (S=binary 0 for type T docs)
X = one byte of "don't care"
L = left margin value (one byte)
R = right margin value (one byte)
BP = back pointer of sector (two bytes)
PP = present pointer of sector (two bytes)
FP = front pointer of sector (two bytes)

DATA STRUCTURES - CONTINUED

Sector structure for ForeWord type W documents

	00		/	/ 065			066		//		074		075	076	077		
0000	1	2	3	1 6 0	X	D	T	T	T	T	T	X	LEFT MARGIN	RIGHT MARGIN	C		BP
0100	1	2	3	1 6 0	X	D	T	T	T	T	T	X	LEFT MARGIN	RIGHT MARGIN	C		PP
0200	1	2	3	1 6 0	X	D	T	T	T	T	T	X	LEFT MARGIN	RIGHT MARGIN	C		FP
0300	1	2	3	1 6 0	X	D	T T T	T T T	T T T	T T T	T T T	X	LEFT MARGIN	RIGHT MARGIN	C	X	S

LEGEND:

T = one byte of tab indicator bits
C = one byte of control bits for the logical line
  Bit 0 = line in use
  Bit 1 = start of chapter key was pressed
  Bit 2 = start of paragraph or beginning of paragraph
  Bit 3 = tabs were used in the line to position the cursor.
  Bit 4 = line is start of new page
  Bit 5 = auto tabs were used in the line to position the cursor.
  Bit 6 = line is centered
  Bit 7 = line has been changed
D = one byte of display character (the character for col 81 display)
S = one byte of "sector type indicator" (S=binary 1 for type W docs)
X = one byte of "don't care"
L = left margin value (one byte)
R = right margin value (one byte)
BP = back pointer of sector (two bytes)
PP = present pointer of sector (two bytes)
FP = front pointer of sector (two bytes)

DATA STRUCTURES - CONTINUED

Sector structure for ForeWord type X documents

```
     ___00_//___0152__0153//0170__0171_0172__0173__0174_____0175_____0176_____0177__
    |  |  |   ||3|3|  ||  |  |  ||  |  |  ||LFT |RHT |  |  |  |  |  |    ||  |  |    ||  |  |    ||  |  |  |
0000| 1| 2| 3 ||1|2| D||T |T |T ||T |X |X ||MRGN|MRGN| C| X| X| X| X|   X||X |X |   X||X |X |   X||X |X |X |
    |__|__|__//9|0|   ||__|__|__//__|__|__||____|____|__|__|__|__|__|____||__|__|____||__|__|____||__|__|__|
    |  |  |   ||3|3|  ||  |  |  ||  |  |  ||LFT |RHT |  |  |  |  |  |    ||  |  |    ||  |  |    ||  |  |  |
0200| 1| 2| 3 ||1|2| D||T |T |T ||T |X |X ||MRGN|MRGN| C| X| X| X|  |  BP||X |  |  PP||X |  |  FP||X |X |S |
    |__|__|__//9|0|__||__|__|__//__|__|__||____|____|__|__|__|__|__|____||__|__|____||__|__|____||__|__|__|
```

LEGEND:

T = one byte of tab indicator bits
C = one byte of control bits for the logical line
    Bit 0 = line in use
    Bit 1 = start of chapter key was pressed
    Bit 2 = start of paragraph or beginning of paragraph
    Bit 3 = tabs were used in the line to position the cursor.
    Bit 4 = line is start of new page
    Bit 5 = auto tabs were used in the line to position the cursor.
    Bit 6 = line is centered
    Bit 7 = line has been changed
D = one byte of display character (the character for col 81 display)
S = one byte of "sector type indicator" (S=binary 2 for type X docs)
X = one byte of "don't care"
L = left margin value (one byte)
R = right margin value (one byte)
BP = back pointer of sector (two bytes)
PP = present pointer of sector (two bytes)
FP = front pointer of sector (two bytes)

FLAG BYTE DESCRIPTION - FOREWORD RELEASE AY06

T -	A capital 'T' in the IDOS directory designates that the file is a standard ForeWord text file (narrow). It may or may not contain a document DESCRIPTION in the IDOS area depending on whether or not it had a document DESCRIPTION in the text area. The records are compressed and are 96 characters long (in order to contain the tabs, flags, and pointers for each line). Because the information is bit oriented in a ForeWord text file it must be considered as binary information in all use of the file while in the IDOS area, i.e. it is not ASCII data.
W -	A capital 'W' in the IDOS directory designates that the file is a wide Foreword text file. It contains a document DESCRIPTION in the IDOS area which specifies a W in the TYPE field. The records are 192 characters long (the last 32 characters contain the tabs, flags, and pointers for each line).
X -	A capital 'X' in the IDOS directory designates that the file is an extra-wide Foreword text file. It contains a document DESCRIPTION in the IDOS area which specifies a X in the TYPE field. The records are 320 characters in length.
t -	A small unshifted 't' in the IDOS directory designates that the file is a ForeWord Text Area. The flag byte is checked whenever a text area is opened from a terminal to allow only contiguous text area files to be opened. Several of the statistics programs use this flag byte also to locate all the text areas.
R -	A capital 'R' in the IDOS directory designates that the file is a read only ForeWord text file. It must have a password entered in the password field and therefore cannot be altered or deleted without knowing the password. The document can be opened or copied by any terminal but cannot be altered. This allows for 'standard' paragraphs, glossaries, or fill in formats to be created for all to use but only the originator can change the document.
C -	The capital 'C' in the IDOS directory specifies a control file.
S -	The capital 'S' in the IDOS directory specifies a source file.

The flag byte in the DESCRIPTION of a document is used to control the way transfers to the archive area take place. A capital 'T', 'W', 'X' or 'R' indicates it is to be transfered as a text document and to retain all information on each line to insure proper editing when brought back into a text area. Any other character will cause the document to be transfered to archive as 80 character records and all nulls in the line are converted to spaces. Lower to upper case conversion does not take place. Lower to upper case conversion can be accomplished by using the offline IDOS processor XTXFIL.

If the type is S the document is transferred in 80-character records with no conversion of special characters to spaces.

The following file naming conventions are generally observed in the ForeWord System - Foreword Release AY06:

1. Source Files

ST L xxx where, L = library module
R R = resident module
O O = overlay module
U U = utility module

2. Relocatable Files

RT L xxx where, L = library module
R R = resident module
O O = overlay module
U U = utility module

where:
x indicates variable information.

3. Absolute Files

IVWORD - ForeWord System
IVOxxx - ForeWord Overlays
PRETXT - Builds Text Areas
CHKTXT - System Maintenance on Text Areas
XTXFIL - Offline utility to transfer to/from ARCHIVE
CNVRTK - Wide document CTRL/K conversion utility
ATUGEN - The ForeWord system generation program
ATUCON - Program used to configure a ForeWord system
ATUSTx - Offline statistics program

4. Other files and their descriptions

IVBASE - Output of virtual table from LOADER
IVQUEx - Queues for use in background processing
STATS - Control file used to execute statistics programs
IVSTAT - File used to record statistics
SVSTAT - Renamed IVSTAT to turn off logging of statistics
PASTAT - File used to record document assembly statistics
WRDCON - Control file used to configure a ForeWord system
WRDGEN - The ForeWord system generation control file
TXGDF - Utility control files
TXTLIB - ForeWord Library
WRDFIL - System Display File
DUMP47 - Dump file
$$IVBR - Used by BROADCAST command, created during initialization
$$IVMW - Used by BROADCAST

CHKTXT (TEXT AREA MAINTENANCE PROGRAM)

CHKTXT must be run on any pre-AY06 text area to convert it to AY06 format. CHKTXT also includes an option (X) to convert documents to an MFE/IV environment where another application is using dual intensity attributes. This option removes all discretionary hyphens from all documents in the text area. This function can be performed by running CHKTXT from a control file of the form:

```
     // CHKTXT
     /T=text area name,X.
     //
```

Do not specify other CHKTXT options when using this conversion feature.

The L option of CHKTXT provides a printed listing of all sectors reclaimed. This information may be useful in recovering deleted documents. CHKTXT can correct the following two types of chain-linkage errors when the L option is specified:

A nonzero back pointer in the first text sector or DESCRIPTION sector of a document.

Zero values in the first-sector and last-sector pointers in the directory entry for a document.

The MFE/IV control file required to correct these types of errors is:

```
     // CHKTXT
     /T=(name),L.
     //
```

The SE mode of CHKTXT is active whenever any options other than the /T (text area) options are specified. The SE mode allows additional commands that may be useful in recovering damaged documents. CHKTXT must be run from a control file. To run CHKTXT while MFE/IV is running, proceed as follows:

a. Exit from Foreword to MFE/IV (press CTRL CURSOR RETURN).

b. Key in // (filename) and press CURSOR RETURN. (filename) must be the name of a control file containing the parameters desired. The options for CHKTXT are as follows:

```
// CHKTXT
/T=NAME OF THE TEXT AREA TO BE CHECKED.
/X DELETE DISCRETIONARY HYPHENS FROM ALL DOCUMENTS CHECKED TO MAKE
   THEM COMPATIBLE WITH DUAL INTENSITY MFE/IV SYSTEMS.
/A TO CAUSE ALL SECTORS TO BE SCANNED IN THE EVENT OF AN ERROR INSTEAI
   OF JUST THE ALLOCATED SECTORS.
/S TO CAUSE CHKTXT TO SUPPRESS SCAN IN THE EVENT OF AN ERROR. NO
   ATTEMPT WILL BE MADE TO CORRECT THE ERROR.
/L TO CAUSE CHKTXT TO PRINT A LISTING OF ALL RECLAIMED SECTOR
   ADDRESSES ON THE PRINTER.
//
```

CHKTXT - CONTINUED

When an error is detected the following options are allowed:

L or Y	Link. The sector shown on the bottom third of the screen is to be linked with the sector show in the center of the screen.
N	Do not link the sectors as shown on the screen. Continue to scan the text area for other possible matches if the option 'S' was not given above.
B	Follow the back pointer in the sector on the bottom third of the screen. Used in an attempt to identify and relink the document.
F	Follow the foreword pointer in the sector on the bottom third of the screen.
C	Continue with the next document in the text area. No changes will be made in the current document and chain linkage errors will still be in the text area. CHKTXT must be rerun to insure all errors are corrected.
R	Reset the screen to the original sector in error.
E	Follow the pointer in the directory for the document to the ending sector and display it on the bottom third of the screen.
T	Indicates that the chain is to be trunctated. The sector in the middle of the screen will be the last sector of the document.

TAKING A DUMP - FOREWORD RELEASE AY06

A. Insure the words 'CKPT TAKEN' are not already on screen 0 of the system. This indicates that ForeWord detected an error and automatically took a CKPT. In this case all that is required is to print or copy the CKPT file.

B. If the words do not appear proceed with the dump of memory.

C. Ensure the disc pack in drive zero is mounted and ready.

D. Set the AUTO/MANUAL switch to MANUAL.

E. Press SYSTEM RESET switch.

F. Press STEP switch.

G. Set console keys to 71100001.

H. Press LOAD switch.

I. Set AUTO/MANUAL switch to AUTO (this will cause a HALT).

J. Move AUTO/MANUAL switch to MANUAL, then back to AUTO (this will clear the HALT).

K. Register X3 will be zero after successful dump.

FORWORD FORMATTED DUMP - FOREWORD RELEASE AY06

This program prints the IVWORD checkpoint file in a format convenient for debugging. You can choose to dump the load map only, the user tables only, memory only, or any combination of these. Use the following JCL to:

Dump the load map only:

```
// WDMP
/P
//
```

Dump the user table only:

```
// WDMP
/U
//
```

Dump memory only:

```
// WDMP
/M
//
```

Dump all:

```
// WDMP
//
```

The above JCL uses file names CKPT, IVBASE, and RTRUSR for the checkpoint, loader save, and user symbol table files. To use some other file names, include the following in the JCL:

```
/I=(checkpoint file name)@(drive)
/B=(loader file name)@(drive)
/R=(user table symbols file name)@(drive)
/J=(jobname)
```

The /J option is used when ForeWord is started with the ASSIGN command.

MEMORY LAYOUT

(Logical layout for extended memory systems)
(Physical layout for non-extended memory systems)

72 K		96 K
0	Interrupt Locations	0
060	Screens, User Tables, Sector Buffers, System Blocks, etc. *	060
	Disc Request Queue	
	Zone Pointer Table	
	Overlay Directory	
$DATA	Optional Resident	$DATA
	Fixed Resident	
@FV 047200	Format Vector Table	@FV 067200
047300	IOID5 (8121 Print Output)**	067300
047400	PUSER (Keyboard Input)	067400
	Overlay Area	
052000	Transfer Program Area or 12 Sector buffers for PROKS	072000
057777		077777

*All memory not allocated to screens, user tables or executable code is organized into the following:

Sector Buffers - 0400 (256) words
number set at CONFIG time

System Blocks - 0100 (64) words

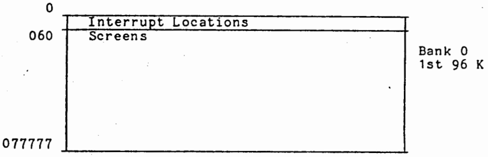
Mini Blocks - 020 (16) words

Micro Blocks - 04 (04) words

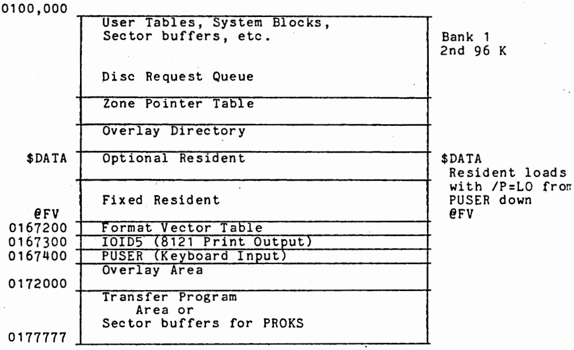
**If there are no 8121 printers, this area will be used for system blocks.

MEMORY LAYOUT

(Physical layout for extended memory systems)

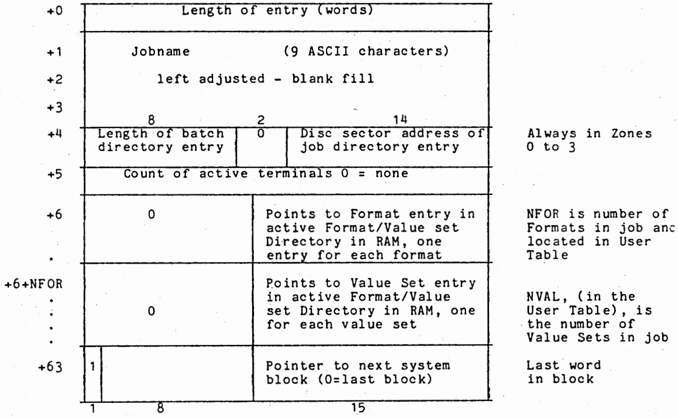


(EXTENDED MEMORY)



ACTIVE JOB DIRECTORY IN RAM

JOBRAM	Pointer to first system block of directory
JOBDES,X1	In user table - points to specific JOBRAM entry



Length is 6+NFOR+NVAL : minimum = 7; maximum = 36.
Therefore, 1-9 entries in block.

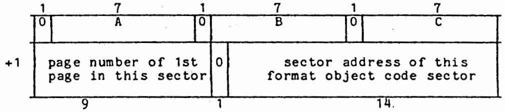
If active terminal count is not zero, job cannot be deleted.

When count is = 0, entry can be deleted. However, deletion will wait until system block would overflow with the addition of another job.

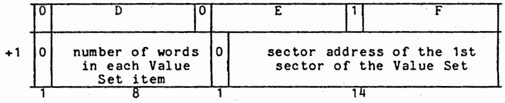
Unused words are all binary zero. When length = 0, the next word > 0 is a valid length.

ACTIVE FORMAT/VALUE SET DIRECTORY IN RAM

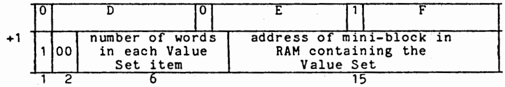
FVSRAM   Pointer to first system block of directory



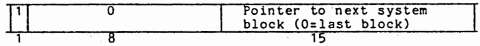
Format Entry (several entries possible for format.)



Value Set Entry (one for each active value



Value Set entry (when it is in a dedicated mini-block)



63rd Word (64th word not used)

001 < ABC < 999 three ASCII digits of format name
001 ≤ DEF ≤ 999 three ASCII digits of value set name

Entries are created when the job directory entry in RAM is built. Several may share entries. A format entry is needed for each sector of format code Four pages can reside in each sector. The "1st page number" will count as 0,4,8,12, etc., for each format and may appear in any random order in the directory.

Unused entries are deleted when the system block would overflow. Entries are on an even boundary. An empty entry is represented by binary zeroes.

PAGE DIRECTORY IN MEMORY (PAGDIR)

PAGDIR	Pointer to first four word entry of directory.
FORPTR,X1	In User Table; points to time word

⟶+0		Link pointer to next 4 word entry	
⟶+1	A	B	C
+2	$0 \leq$ page # ≤ 511	address of system block (PSB)	
+3	Time last referenced		

An entry for each page of format code. Binary zeros for the name = no entry.

NOTE: The page directory consists of a linked list of micro blocks; one micro block for each page in memory. The list is kept in sequence by format and page number.

+0 Pointer is zero if there are no more entries in PAGDIR.

+1 ABC is the format name in ASCII: 001 to 999.

+2 PSB: non-zero = address of system block containing page of code
zero = entry is a page-in request (no block assignment)

+3 Time: non-zero = time this page was last referenced = t-86400 where t is seconds since midnight
zero = page-in request (same as PSB=0)

ABC is the format name in ASCII: 001 to 999.

Up to 21 entries per system block.

IXRRAM

Displ.	Use
0	ASCII INDSET number (with bit 8 set) or zero
1	Sector address of the highest index level.
2	Bits 0-11 Starting column number of the key field in the data batch records. Bits 12-23 Key length in bytes.
3-4	"Who-is-using" bits. Bits 0-23 of word 3 followed by bits 0-7 of word 4 correspond, in that order, to the 32 possible terminals. For example, bit 1 of word 4 corresponds to terminal 25. If a "using" bit is on, the terminal it corresponds to has a record selected in the INDSET given by word 0. Words 5 and 6 specify which record. More than one "using" bit can be on, meaning more than one terminal has the record selected. A terminal may have the record selected, but it might be in a waiting state (with the "WAITING FOR TERM.XXX TO RELEASE INDSET iii" message on his screen) if another terminal currently has exclusive access to the record (see word 7).
5	If any bit in words 3-4 is on, this word has the sector address of the sector containing the selected record.
6	If any bit in words 3-4 is on, this word has the displacement in the sector of the first (header) word of the selected record.
7	User table address of exclusive accessing terminal, or zero if no terminal has exclusive access. When non-zero, the corresponding "using" bit will also be on.

USER TABLE SKELETON DEFINITION

```
*       DCN     01401     IO in PUSER table points here.
USERTB  EQU     0
KSCUR   EQU     1         Current position in KS queue.
KSRES   EQU     2         FWA of keystroke queue (UT ADDR + 0140).
SCR     EQU     3         Execute to place characters on screen.
FWA     EQU     4         First word of screen.
SLWA    EQU     5         Last word + 1 of screen.
PSROT   EQU     6         Cursor SCR first operand.
DEST    EQU     7         Cursor SCR second operand.
SCOL    EQU     010       Cursor col id = - number of columns from right end.
WPO     EQU     011       Current fields governing ATTR during WGT call.
MODE    EQU     012       24 flags. (Defined later).
OPPTR   EQU     013       Pointer to operator directory.
JOBDES  EQU     014       Pointer to job descriptor table.
FORPTR  EQU     015       Pointer to current format.
FSROT   EQU     016       (Pair) pointer to start of current field.
FDEST   EQU     017       First word of current field.
REQ8    EQU     020       Request for LEVEL 8.
RECLOC  EQU     021       Location of current REC in sector.
WIDTH   EQU     022       - (Width of field).
MODPTR  EQU     023       Pointer to in-front modifier.
LCR     EQU     024       Execute to get character from old record in key verify,
                            LCR = SEC ADR for previous REC
ACCP1   EQU     025       Pointer to first four accumulators.  0-3.
ACCP2   EQU     026       4-7.
ACCP3   EQU     027       8-11.
ACCP4   EQU     030       12-15.
ACCP5   EQU     031       16-19.
ACCP6   EQU     032       20-23.
NFOR    EQU     033       Number of formats.
NVAL    EQU     034       Number of value sets.
MFWA    EQU     035       Printer to message line, BIT0=1: lines are off screen.
MSIZE   EQU     036       - Size of screen including dead areas.
MSIZEB  EQU     037       - Size in bytes.
KBTYPE  EQU     040       Keyboard descriptor word from CONFIG.
LASTKS  EQU     041       Last three keystrokes.
TVUSER  EQU     042       Trail verify user pointer
TERMN   EQU     043       Terminal Index 0-31.
BATCHN  EQU     044       ASCII batch ID - 6 characters - even boundry.
BACHN1  EQU     045       Continued.
BINDOC  EQU     046       Binary document no.
TERMAS  EQU     047       Terminal index in ASCII, blank fill.
BATCH   EQU     050       Address in batch directory.
LINKBS  EQU     051       Return address from LINKVS.
BINREC  EQU     052       Binary record number.
MAXREC  EQU     053       Largest binary record number.
NEXTUR  EQU     054       Unconditional keystroke return address.
STATS   EQU     055       Operator statistics table.
STATS1  EQU     056       Second word of operator statistics table.
STATS2  EQU     057       Third word of operator statistics table.
BINCOL  EQU     060       Binary column counter.
ALARM   EQU     061       Audible alarm flag.
SAVPTR  EQU     062       Save pointer for backspacing.
MODEXT  EQU     063       Mode extension flag word. (Defined later).
```

USER TABLE SKELETON DEFINITION - CONTINUED

```
BLOCK7 EQU    064       Terminal block word. (Defined later).
VRECB  EQU    065       Pointer to verify record buffer. (Defined later).
RECHDR EQU    066       Current record header.
RECSIZ EQU    067       Record size.
SECTOR EQU    070       Current sector number.
NEXTCR EQU    071       Return word from BRM NEXTC.
COLUMN EQU    072       Pointer to column count on screen.
ECODE  EQU    073       Error code. (Defined later).
FX2    EQU    074       X2 = Location in format (via BAL).
FX3    EQU    075       X3 = - (columns remaining in field).
VLROT  EQU    076       LCR pointer for verify.
VDEST  EQU    077
*      PZE    0         0100, lost keystroke routine from IOID + 040.
*      BRM    KEYL      0101
*      BRD    $-2       0102
QUE8RA EQU    0103      Terminal restart address when overlay completed.
WPA    EQU    0104      Extra LCR/SCR word pair.
WPA1   EQU    WPA+1
WPB    EQU    0106      Extra LCR/SCR word pair.
WPB1   EQU    WPB+1
WPC    EQU    0110      Extra word pair for scratch.
WPC1   EQU    WPC+1
WPD    EQU    0112      Scratch word pair.
WPD1   EQU    WPD+1
DQSLOT EQU    0114      Current position in disc queue.
OV36BL EQU    0115      Multi-mode block address.
FMTST1 EQU    0116      First 24 bits of format backspacing stack.
FMTST2 EQU    0117      Second 24 bits of format backspacing stack.
MAXDOC EQU    0120      Largest binary document no.
MODE3  EQU    0121      24 single bit flags. (Defined later).
LASSEC EQU    0122      Last sector in chain.
@BUF   EQU    0123      Pointer to format workspace.
@AVE   EQU    0124      Pointer to user's file access table pointers.
CURC   EQU    0125      Counts Level 4 until cursor change needed.
RS     EQU    0126      Digits to right of DEC point LZERO FIELD.
SCPBAD EQU    0127      Used by screen print.
KV     EQU    0130      Byte 0 = SCRAM CHAR, 9-23 = RECLOC for prev. REC.
MODE2  EQU    0131      More single bit flags, like 'MODE'. (Defined later).
CURSES EQU    0132      Three cursor characters (depends on field intensity).
STATS3 EQU    0133      Fourth word of operator statistics table.
@AT    EQU    0134      PTR to word containing current field ATTR. (Defined
                        later).
ATMASK EQU    0135      Mask for MDT in current ATTR (see description above).
MAXCOL EQU    0136      Highest BINCOL so far in format.
SX3    EQU    0137      Save word used by -5C.
       Keystroke Queue From 0140 through 0167
CUR    EQU    0170      Off screen copy of blinking word.
F1GA@  EQU    0171      PTR to field one physical governing ATTR.
MBLOCK EQU    0172      PTR to message line system block.
                        BIT0=1: Lines are in home position.
COMM   EQU    0173      Communications word. (Defined later).
L7SVX2 EQU    0174      Save X2 during level 7 interrupts.
L7SVX3 EQU    0175      Save X3 during Level 7 interrupts.
@SCB   EQU    0176      MSFP pointer to this terminal's submonitor.
MAPWRD EQU    0177      Extended memory map word.
```

USER TABLE DATA STRUCTURES

```
***     MODE    ***************************************************************
*
*       0       1=Normal mode from keyboard, 0=feed through from Level 8.
*       1       1=PROG CTRL off, 0=on.
*       2       1=Release key in process, 0=not.
*
*       3       1=Verify active, 0=inactive.
*       4       1=Ver corr (1 character) in process, 0=not.
*       5       1=Waiting at end of field (must RLS/left zero), 0=not.
*
*       6       1=First pass (disc to screen), 0=not.
*       7       1=ASD on, 0=off.
*       8       1=Special mode.
*                         Note:  Bits 8, 9, 10 = 0 is entry mode.
*       9       1=Verify mode.
*       10      1=Find mode.
*       11      1=Skip/release inhibit (must fill), 0=not.
*
*       12      1=Character input inhibited, 0=allowed.
*       13      1=New record, 0=existing record.
*       14      1=Aux Dup inhibited, 0=not.
*
*       15      1=Index inhibited, 0=not.
*       16      1=Pseudeo new record, 0=not.
*       17      1=No job assigned, 0=assigned.
*
*       18      1=Must enter inhibit, 0=not must enter.
*       19      1=Perform validation checks, 0=not.
*       20      1=First column of record, 0=not first column of record.
*
*       21      1=Waiting for minus key (LZ field), 0=not.
*       22      1=Not first column of field, 0=first column of field.
*       23      1=Left zero field, 0=not left zero field.


*
***     MODE2   ***************************************************************
*
*
*       0       Cursor state flip-flop.
*       1       Verify reconstruct.
*       2       Search-state on.
*
*       3       1=Extended precision.
*       4       1=Correction made in K.V. to current field.
*               If reverification option specified in CONFIG.
*       5       1=Mode X,T (decentralized batch transmission) in process.
*
*       6       Message line home bit -- 0: on screen, 1: off screen.
*       7       1=3270 mode
*       8       1=Backspace field encountered.
*
*       9       NEWFMT 1=format style (always set true).
*       10      TRMBIT current logical term value (1=true).
*       11      EXPBIT current logical expression value (1=true).
*
*       12      IFTHEN 1=Currently between "IF" and "THEN" in format.
*       13      Reserved.
*       14      1=on-line, 0=off-line.
*
*       15      Reserved.
*       16      Reserved.
*       17      Reserved.
*
*       18      Reserved.
*       19      Reserved.
*       20      NDR Bit 1: No data record for this format.
*
*       21      1=INDSET - FIND mode in progress.
*       22      1=In a record, 0=not in a record.
*       23      1=Waiting for new mode character, 0=not.
```

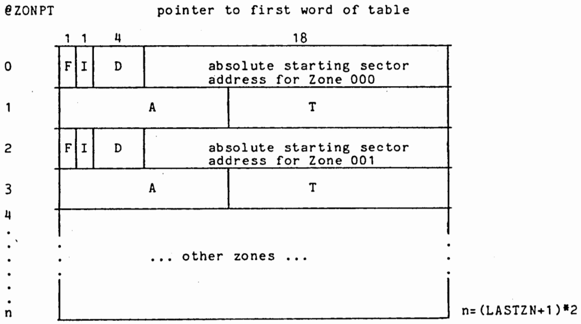
```
*
***     MODE3   ******************************************************************
*
*      0-5      Reserved.
*        6      Noup        Record up and document up not allowed.
*     7-18      Reserved.
*
*       19      1=Doc KV #2 (set in doc key-verify initiation,
*               in overlay 38 entry 6).  This bit remains on
*               during the doc key-verifying. It is turned off when new record is
*               hit and the system reverts back to entry mode.
*
*       20      1=Doc KV #1 (set by KVDOC command in format).  This bit remember
*               that a KVDOC was executed.  At NXTREC, if this bit is on then
*               the document key-verify operations are initiated by calling over-
*               lay 38 entry 6.  The overlay clears this bit.
*
*       21      Reserved.
*       22      1=Out-of-balance (set at SREC in key ver)
*               This is what the OUTBAL Test tests in the format.
*
*       23      1=Batch has marked DOC (set by mark command in format)
*
***     MODEXT  ********************************************************************
*       0       Unconditional release in process.
*       1       Mode key.
*       2       Record up.
*
*       3       Record delete.
*       4       Record insert.
*       5       Home key.
*
*       6       Entry mode bringup.
*       7       Doc up (always ignore search-active).
*       8       Doc down (always ignore search-active).
*
*       9       Immediate-next-record (ignores any search-arg).
*      10       This-record (i.e., No alteration of sector, RECLOC)
*      11       Record down.
*
*      12       PROG key.
*      13       Do-not-update BINREC/BINDOC (Used in conjuction with bits 6, 10
*               and maybe others.)
*      14       Clear key.
*
*    15-23  Reserved.
*
***     ECODE   *****************************************************************
*       0       1=Flashing a message, 0=not flashing a message.
*       1       Flash state:  1=message is up, 0=blank line is up.
*       2
*
*       3
*       4
*       5
*
*       6
*       7
*       8       =1:  message is in keystroke queue, =0: message not yet in queue
*
*    9-23       = message number if message is not yet in queue.
*               = message length if message is already in the queue
*
*       ECODE   = 0 means no message action whatsoever.
*               < 0 means flashing a message.
*               > 0 means steady state message.
*
***     VRECB   ******************************************************************
*
*       0       Record correction (PROG or REC INS).
*       1       Field correction (shift field forward).
*       2       Column correction (shift CORR/RESET).
*    3-23       Reserved.
```

```
 **     COMM      ******************************************************************
        0         1=No COMM operation initiated for terminal 0: something going
        1         1=Selected by CU3270
 *      2         1=Transmit key on keyboard hit.
 *
 *      3         1=Send command in process, i.e., wake terminal without a
 *                keystroke upon completion).
 *      4         1=Waiting for write from host.
 *      5         1=Was written on last select (cleared upon discovery).

        6         Not used.
        7         1=Printer IR task is active at this device.
        8         Not used.
 *
 *   Bits 9-23 Are copied directly from CU3270 COMM word at deselect.
 *      9         1=Erase write.
 *      10        1=Erase all unprotected.
 *      11        1=Copy.
 *
 *      12        1=Write.
        13        1=Read modified.
        14        1=Read buffer.

        15        1=Escape in data.
 *      16        Reserved.
 *      17        Reserved.
 *
 *   18-19        Print line length:
 *                00    unformatted print.
 *                01    40 characters/line.
 *                10    64 characters/line
 *                11    80 characters/line.
 *
 *      20        Not used.
 *
 *      21        Not used.
 *      22        1=Restore keyboard.
 *      23        Not used.
 *
 *
 ***    @AT, ATMASK --- FIELD GOVERNING ATTRIBUTE LOCATION    *******************
 *
 *      @AT       = Pointer to word containing current field governing attribute.
 *                  Not valid if ATMASK = 0.
 *
 *      ATMASK = Mask for MDT bit in current field governing attribute.
                    0:  MDT should not be set on this attribute.

        A complete description of the use of these two cells is included with
        the attribute handling routines -- PUTATR, etc.
 *

 *
 ***    BLOCK7 --- PROKS TERMINAL ACTIVITY BLOCKED   *****************************
 *
 *      BLOCK7 <> 0:  Normal PROKS processing is blocked at terminal.
 *
                Bit 0 = 1: Terminal is ready to be restarted.

                Bit 8 = 1: Terminal is surreptitiously getting keystrokes.

 *              Bits 9-23: Re-start address.
 *
 *      BLOCK7 = 0:   Normal PROKS processing is not blocked.
 *
 *      Bit 8, the "surreptitiously getting keystrokes" bit, is on when normal
 *      format processing is blocked, but a special keystroke processing routine
        is running, getting keystrokes. This bit is used by $DRIVE so that it
        knows that keystrokes are allowed at the terminal even though PROKS
        is blocked.

 *      In all cases, PROKS really means keystroke and format execution
 *      which have run at hardware level 7 in the past.
```

ZONE POINTER TABLE

The Zone Pointer Table contains all of the information necessary to control allocation, deallocation, reading, and writing of zone sectors. It is built at initialization by the subroutine OPENDB. There is one entry in the table for each possible zone beginning with zone 000. The table is variable length and is only as large as is necessary to accommmodate the highest numbered zone defined. Undefined zones within the range of the table are represented by entries of all zeros.

The resident word @ZONPT contains the address of the first word of the Zone Pointer Table. The resident word LASTZN contains the largest zone number defined.



F: 0 = sectors are available in this zone
1 = this zone is 100% full

I: 0 = allocation permitted
1 = allocation inhibited, zone is protected.

D: 4-bit IDOS logical drive number of zone (0 - 11)

A: Number of sectors available for allocation in this zone (12 bits)

T: Total number of sectors in this zone (12 bits)

ZONE LAYOUT

	Sector	
Zone 000	0000	Scratch area used by MAINT
	0001	Allocation Sector - Zone 000
	0002	Master Sector
	0003	CONFIG Sector
	0004 - 0015	Reserved
	0016	Reserved for use by NP/80
	0017	Master Sector (Backup copy)
	0020 - 7677	Directories, Overlays, Data, etc.

Zone 001-377	0000	scratch area used by MAINT	
	0001	Allocation sector for this zone	
	0002 - 0015	Reserved	
	0016	Reserved for use by NP/80	
	0017	Reserved	
	0020 - 7677	Directories, Data, etc.	*1

*1 Directories, format object code, and value sets may be in zones 0-3 only.

SECTOR STRUCTURE

The VISION database has its own internal structure separating it into many smaller "files" such as Job Directory, Batch Directories, batches of data, etc. It has its own allocation table, and allocation and deallocation routines in VISION keep track of the database in a chained structure. This is handled through a standard 4-word header in each sector. This header also identifies the type of information stored in that sector. The only exceptions to this standard header are the Overlay object code sectors, which have no header words, and the Allocation sectors which have only a Here pointer.
The standard 4 word header is as follows:

Word 0: ....ZZZZZZZZSSSSSSSSSSSS

Z.....S Forward pointer indicating the relative sector address of the next sector in this chain. Zeros indicate the last sector of the chain.

Word 1: ....ZZZZZZZZSSSSSSSSSSSS

Z.....S Backward pointer indicating the relative sector address of the previous sector in this chain. Zeros indicate the first sector of the chain.

Word 2: ....ZZZZZZZZSSSSSSSSSSSS

Z.....S Here pointer indicating the sector's own relative sector address.

Word 3: CCCC*DISP*ZZSSSSSSSSSSSS

Code word. If this sector is a format/value/index set sector this word contains the name of the format or value set or index set.

CCCC 4-bit code for sector type:

0000 - Master Directory (remainder of code word also zero)
0100 - Job Directory
0010 - Batch Directory
0101 - Maintenance Sector (or OID)
0110 - Format/Value-Set Directory
0011 - format/value-set/index set object code (by nature of the name)
1000 - data sector which is not full}(remainder of code
0000 - data sector which is full   } word non-zero)

DISP The six most significant bits of the displacement into the Job Directory sector (for Batch Directory sectors) or Batch Directory sector (for data sectors) of the entry to which this chain belongs. These bits are zero for all but Batch Directory and data sectors.

Z.....S The zone and sector address of the entry described under *DISP*. These bits are zero for all but Batch Directory and data sectors.

ALLOCATION SECTOR

his sector is always relative sector 0001 in every zone. Note that the first 16 sectors (0000-0017) of a zone are not allocated by VISION. The relative sector 000 of zone 000 cannot be used, since the here pointer would be zero.

+0	ASCII Day number			Time stamp set by VISION when a change is made
+1	time (T-86400)			
+2		8 bit zone no.	0 0 0 1	Here pointer
+3	usage sequence number			*1
+4	Reserved		number of sectors in zone	
+5		relative sector 020-037		*2
.				
.				
.				
.				
+377		relative sectors 07660-07677		

*1: Each time VISION is brought up this sequence numebr is checked to be identical in all DATzzz files. If they don't match, an incomplete restore may have occurred and MAINT must be run to set them all equal. If they all match, then they are all bumped by one. They will recycle to 0.

*2: 251 word allocation table for this zone. Bits 8 to 23 of each word represent 020 sectors of a zone. Zero bits indicate sectors in use or not assigned to the zone. One bits indicate sectors available for allocation.

MASTER SECTOR

0	0	00003		Pointer to CO
1	0	00000		Back Pointer
2	0	00002		Here Pointer
3	0	00000		Code Word
4	(t-86400) binary seconds since midnight = TIME			
5	ASCII Julian day number (001-366)			
6	0	first sector addr. of FVS dir. (14 bits)		
7	0	first sector addr. of JOB dir. (14 bits)		
010	0	first sector addr. of OID dir. (14 bits)		
011	Reserved			
012	ASCII release level, 1st 3 chars.			
013	ASCII release level, 2nd 3 chars.			
	0 8	9 11	12 23	
014 015 thru 0156	Reserved	number of secs -1	sec. addr. of overlay 1 sec. addr. of overlay 2 sec. addr. of overlay 99	Locations of VISION overlays within Zone 0 (12 bit sector address)
0157	Reserved			
0160	0	zones 000 thru 017		*1
0161	0	zones 020 thru 037		
thru	0	:		
0177	0	zones 0360 thru 0377		
0200	Reserved byte 0	no. zones - 1 byte 1	largest zone no. byte 2	
0201	Reser.	18 bit start. sec. addr. of zone 000		set by MAKESY
0202	Address of Format Vector Table in Ram (@FV)			
0203 thru 0377	Reserved			

*1 A 16 word table maintained by ZONEMA. Bits 8 to 23 of each word represent 020 possible zones. One bits indicate defined zones. Zero bits indic undefined zones.

CONFIG SECTOR (Sector 3 in Zone Header)

+0	Forward Pointer (Points to Keys Sector)	
+1	Backward Pointer (00000002)	
+2	Here Pointer (00000003)	
+3	Code Word (00000000)	
+4	Time of Last Change (t-86400 Seconds)	
+5	Day of Last Change	

	0	1	2 5	6 12	13 23	
+6	KB	AUD	-Reserved- (4 Bits)	Size of Screen in Words/16 (7 bits)	FWA of Screen /16 (11 bits)	*1

+046	VISION P2L2P Table - Maps devices to screens (32 words)	*2
+106	Numbers of disc buffers	*3
+0107	Reconfiguration Password (Scrambled) Default: 8 Blanks	
+0110	Reserved	
+0111	Print/Log Control Password (Scrambled) Default: 12345678	
+0112	Supervisor Password (Scrambled) Default: PASSWORD	
+0113	Reserved for Special Systems (3 words)	

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 17	18 19	20 23	
+0116 CONFIG SYSFIG	VIDEO	EXTMEM	CH	LAMLOG	CR	PR	MAG	DEN	SOR	CCB	DOCS	BEEPER	CCODE	LAM2	RESRVD	DUMP TYPE (3 BITS)	NO. OF TAPE DRIVES (2 BITS)	NO. OF DISC DRIVES (4 BITS)	*4

	0	1	2	3	4	5	6	7	8 9 10	11 12 13	14 15 16 17 18	19 20 21 22 23	
+0117 CNFIG1	RECPRT	ERP	FIND	CFPROP	REV FY	BATCH	RESRVD	RESRVD	TAPE BUFFER SIZE (3 BITS)	RESRVD	NO. OF 8121 PRINTERS (5 BITS)	NO. OF TERMINALS (TRUE VALUE) (5 BITS) IS 32 TERMS WHEN = 0.	*5

CONFIG SECTOR CONTINUED

+0120	Reserved						
+0121	Reserved						
+0122	RESERVED (3 Bits)	MAXBUF (4 Bits)	DYNBAL (3 Bits)	NKS (2 Bits)		LEV78B (5 Bits)	*6
+0123 CNFIG4	No. of Buffered Printers						
+0124 CNFIG5	0 1 2 3 4 5 6 7 8 9	10 L C	11 12 13	14 15 16 17 18 GP DELAY (5 BITS)	19 20 21 22 23 CONTROL UNIT ID (5 BITS)		*8
+0125	Reserved for Future Use (11 words)						
+0140	Jobname of Current Log File (3 words)						
+0143	Batch No. of Current Log File (2 words)						
+0144	PRSTAT						*9
+0146	8121 Printer Specification Words (16 words)						*10
+0166	User Table TERMN (32 words)						
+0226	Spool Parameters (32 words)						
+0266	Spool Conversion Table - CONTBL Table (17 words)						
+0307	Reserved for Future Use (24 words)						
+0337	Reserved for Users (33 words)						

*1 KB: 0=Source, 1=029
AUD: 0=No Audible Alarm
32 Words - One for each possible keyboard device

*2 VISION P2L2P Table
Byte 0: Reserved
Byte 1: Maps devices to screens (index into with device no.)
Byte 2: Maps screens to devices (index into with screen no.)

*3 If # screens < 10; # buffers recommended = # of screens. Min. is 6.
If # screens ≥ 10; # buffers recommended = # of screens /2 +2. Min. is

ADDITONAL FOOTNOTES FOR THE CONFIG SECTOR FOLLOW ON NEXT PAGE

FOOTNOTES FOR CONFIG SECTOR - CONTINUED

*4 CONFIG (SYSFIG)
Video: 0=48, 1=81
EXTMEM: 0=None, 1=Extended memory (IV/90 Model 2)
CH: 0=No channel
LAMLOG: 0=No memory log
CR: 0=No card reader
PR: 0=No printer
MAG: 0=9-track, 1=7-track
DEN: 0=Low, 1=high
SOR: 0=No shift override of field type, 1=allow shift override
CCB: 0=Not concurrent Bisync, 1=concurrent.
DOCS: 0=Count records, 1=count documents.
BEEPER: 0=One beep per error, 1=one beep per keystroke after error.
CCODE: 0=EBCDIC, 1=ASCII
LAM2: 1=Yes, 0=no.
DUMP TYPE 0=8230, 1=8240, 2=8260, 3=Tape

*5 CNFIG1 (SYSFG1)

RECPRT: 0=No 12 channel printer for received data (Bisync)
ERP: 0=Extended error recovery
FIND: 0=Do not count find mode statistics
1=Count find mode statistics
REVFRY: 0=Field reverification, 1=not.
CFPROP: 0=VISION print spooling, 1=Bisync direct print.
BATCH: 0=Do not keep detail operators statistics by batch, 1=do.

Tape Buffer Size: Number of sectors for tape buffer (maximum is 6).

*6 LEV78B, PAGSEC, etc. - Set by $SETPP Supervisory Command.

*7 CNFIG4 - Used by VISION.

*8 CNFIG5 - Used by VISION.

LC: 1=Lower case in default 3270 mode.

*9 PRSTAT
Bit 0: Reserved
Bit 1: 0=No log file, 1=log file exists.
Bit 2: 0=No printer, 1=printer.
Bit 4: Reserved.
Bit 5: Clear request.
Bits 6-18: Reserved.
Bit 19: 1=Log C-Type messages.
Bit 20: 1=Log B-Type messages.
Bit 21: 1=Screen print turned on in log.
Bit 22: 1=Print C-Type messages.
Bit 23: 1=Print B-Type messages.

*10 8121 Printer Specification Word:
Bit 0: 1=Pica; 0=Elite.
Byte 0: Number of printed lines per page - 1.
Byte 1: Number of increments to advance each line.
Byte 2: Total number of lines per page.

JOB DIRECTORY (on Disc)

<table>
<tr><td></td><td></td><td colspan="7">Standard 4 word header</td><td></td></tr>
<tr><td></td><td>+4</td><td colspan="7">Time of last change (t-86400 seconds)</td><td></td></tr>
<tr><td></td><td>+5</td><td colspan="7">Day of last update in ASCII</td><td></td></tr>
<tr><td>0</td><td>+6</td><td colspan="7">Word length of entry (even binary)</td><td></td></tr>
<tr><td>+1</td><td>+7</td><td colspan="7">Job name (1-9 ASCII characters)
left adjusted blank fill.</td><td></td></tr>
<tr><td>+4</td><td>+012</td><td>S</td><td>O</td><td>B</td><td>D</td><td>X</td><td>reserved</td><td>Batch Directory 1st sector address</td><td>*1</td></tr>
<tr><td>+5</td><td>+013</td><td colspan="2">NFOR
of
formats
1-15</td><td colspan="2">NVAL
of
V. Sets
0-15</td><td colspan="1">NACCP
number of
accumulators
0-23</td><td>U</td><td>SIZE
binary record size
in bytes</td><td>*2</td></tr>
<tr><td>+6</td><td>+014</td><td colspan="7">Scrambled password required for access
0 = No password required</td><td></td></tr>
<tr><td>+7</td><td>+015</td><td colspan="5">MAXSIZ
binary maximum
record size (bytes)</td><td colspan="2">starting sector address of
Batch Index chain
Zero = No Batch Index</td><td>*3</td></tr>
<tr><td>+010
.
MAX
+030</td><td>+016</td><td colspan="7">Packed decimal format and value set id numbers. Left to right with formats first. Last half word has binary zero fill (if any). Extra word (to make even length) has binary zero fill (if any).</td><td></td></tr>
<tr><td>.....</td><td>........</td><td colspan="7"></td><td></td></tr>
</table>

An entry in the job directory cannot be moved once defined. All entries are even length.

*1

S = Job has been "selected" by $SELECT command
O = Batches have been ordered for transfer by $ORDER command
B = BUSY - Job is temporarily unavailable due to $PURGE or $ORDER processing
D = Double Accumulators are defined for this job
X = Mode X-T enabled for this job ($XTSET)

*2

NFOR = Number of formats defined for this job (4 bits). Range: 1-15. Same value as NFOR in User Table.
NVAL = Number of Value Sets defined for this job (4 bits). Range: 0-15. Same value as NVAL in User Table.
NACCP = Highest accumulator number defined for this job. (5 bits). Range: 0-23.
U = Special user use (1 bit).
SIZE = Record size for fixed length records (10 bits). 0= variable length records.

*3

MAXSIZ = maximum record size permitted under variable length records (10 bits).

Program MAINT will clear the BUSY flag and delete the Batch Index.

BATCH DIRECTORY (on Disc)

<table>
<tr><td></td><td></td><td colspan="5">Standard 4 word header</td><td></td></tr>
<tr><td>0</td><td>+4</td><td colspan="5">Batch id "number" (1-6 characters)
left adjusted ASCII with blank fill</td><td>*1</td></tr>
<tr><td>2</td><td>+6</td><td colspan="2">4 bits of zero</td><td colspan="3">relative sector address of first data record</td><td></td></tr>
<tr><td>3</td><td>+7</td><td colspan="5">record count - undefined if bit 0=1</td><td>*2</td></tr>
<tr><td>4</td><td>+010</td><td colspan="3">- reserved -</td><td colspan="2">document count</td><td>*3</td></tr>
<tr><td>5</td><td>+011</td><td colspan="5">State and request flags (defined on next page)</td><td></td></tr>
<tr><td>6</td><td>+012</td><td colspan="5">Time last used (t-86400) binary seconds</td><td>*4</td></tr>
<tr><td>7</td><td>+013</td><td colspan="5">Day code (3 ASCII digits)</td><td></td></tr>
<tr><td>010</td><td>+014</td><td colspan="5">Transaction code (3 ASCII characters)</td><td></td></tr>
<tr><td>011</td><td>+015</td><td colspan="2">4 bits of zero</td><td colspan="3">relative sector address of last data record - LASSEC</td><td>*5</td></tr>
<tr><td>012</td><td>+016</td><td>L</td><td colspan="4">Last entry operator id</td><td>*6</td></tr>
<tr><td>013</td><td>+17</td><td>L</td><td colspan="4">Last verify operator id</td><td>*6</td></tr>
<tr><td>014</td><td>+020</td><td>H</td><td>C</td><td>W</td><td>Offset from sec. header</td><td>sector address of next entry</td><td>*7</td></tr>
<tr><td>015</td><td>+021</td><td colspan="5">- Reserved for special user use -</td><td></td></tr>
<tr><td>016</td><td>+022</td><td colspan="5">4 words for each accumulator
12 digit ASCII character right justified
Up to 23 accumulators = 92 words</td><td>*8</td></tr>
</table>

Minimum entry = 14 words if no accumulators
Maximum size = 106 words if 23 accumulators
All entries are fixed size for each job
A maximum of 18 entries per sector (no accumulators)

*1 - 0 = no entry (Purged or never existed)
*2 - 0 = empty batch (no records)
*3 - 0 = no documents (no $DOCHDR records)
*4 - Always negative
*5 - If the whole word is 0, the address is assumed to be unknown.
*6 - L = 1 for last operation on batch, else L = 0
*7 - For ordered job, points to next batch in order chain. H = 1 for head, C = 1, part of order chain, W = used during $ORDER. (Offset is always even, so low order zero is dropped.) Set = 0 when batch created.
*8 - Accumulators are set to ASCII zeros when the batch is created.

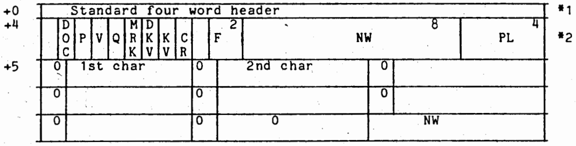
"State and Request Flags" word in Batch Directory Entry
(Word 5 of batch directory entry)

0 1	reserved		
2	1=complete	(COMP)	
3 4	00=Find 10=Verify 01=Entry 11=Verify-Complete		
5	1=Has been written to output device	(WWR)	
6	1=Has been read from input device	(WRD)	
7	1=Has been reformatted	(WRF)	
8	1=Was created by reformat	(CBRF)	
9	1=Display request	(DIS)	
10	1=Purge request	(RPUR)	*
11	1=Write request	(RWR)	*
12	1=Read request	(RRD)	*
13	1=Reformat request	(RRF)	*
14	1=Locked	(LOCK)	*
15	1=Batch in use	(INPR)	*
16	1=Active Log File	(LOG)	*
17	1=Document Marked	(MARK)	
18	1=Indset Batch	(XSET)	*
19 20 21	- Reserved -		
22 23	Reserved for special user use		

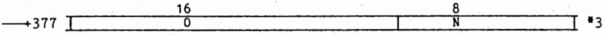
← Last operation performe
on this batch

* If any of these flags ar
on, a "batch active" con-
dition exists and access
to the batch is inhibite

DATA SECTORS



Sector trailer (1 word) when code word is negative:



*1 Special flag in bit 0 of Code word:
0 = sector is full; N = 0400 is implied
1 = sector is not full; word 0377 has N of 004-377

*2 Record header word:

NW:	# of words in entry (2-251)
F:	# of trailer bytes (1-3). Record size in bytes = (NW-1)*3-F.
P:	Program level changed in record
V:	Validation override used
Q:	Unintelligible key used
KV:	Has been key verified
DKV:	Has been document-key verified
CR:	Has been altered since entry
DOC:	This record starts a new document
MRK:	This document has been "marked"
PL:	Program level 0 to 14

*3 Sector trailer word (only present if bit 0 of code word is 1):
N: Number of words being used in this sector (004-377).
Also is relative address in sector where unused area begins.

Algorithm for # of records in a sector:

Total sector length = 256 x 3 = 768
Sector Header = 4 x 3 = 12
Available for data = 252 x 3 = 756

# of characters in record +1 (for trailing NW) rounded up to next multiple of 3 then add 3 (record header) and divide into 756.

NOTE: On jobs with variable length records, the record mix will have to be considered. The system will try to fit each record into the current sector.

MAINTENANCE SECTOR

	Standard 4 word header	
+4	Day number (ASCII) most recent day	
+5 to +0200	Reserved	*1
+0201	S Day number (ASCII)	
+0202	Start time (t-86400) when VISION brought up	*1
+0203	End time set by shutdown (s=1) time of crash; set from Master Sector (s=0)	
+0204	entries for 40 most recent reloads including the one above	
+0371 +0377	7 words reserved for future use	

If day number = binary zero, no entry

*1 One 3 word entry for each loading of VISION. The most recent is first. If end time=0, (S must =0); end time is set from time in Master sector. The stack is pushed down and a new entry is built.

To clear "MAINT flag" to bring up VISION after a crash without running MAINT:

Set word +0212 to a non-zero value

BATCH INDEX

	Standard 4 word header
+4 +5	Batch Name
+6	Displacement in Batch Directory sector
+7	Batch Directory sector address
. . . +0377	more entries

FORMAT/VALUE SET DIRECTORY (on Disc)

<table>
<tr><td colspan="11">Standard 4 word header</td><td></td></tr>
<tr><td>0</td><td colspan="4">A</td><td>0</td><td colspan="3">B</td><td>0</td><td>C</td><td>*1</td></tr>
<tr><td colspan="3">Reserved
6 bits</td><td>S
D</td><td colspan="3">Screen
Code
2 bits</td><td>G</td><td colspan="3">Starting sector address of
format object code</td><td></td></tr>
<tr><td colspan="8">Record size (bytes)
10</td><td colspan="3">Reserved</td><td></td></tr>
<tr><td>0</td><td colspan="4">A</td><td>0</td><td colspan="3">B</td><td>1</td><td>C</td><td>*2</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="3">Starting sector address of
Value Set</td><td></td></tr>
<tr><td colspan="9">Reserved</td><td colspan="2">Number of words
in Value Set item
8 bits</td><td></td></tr>
<tr><td>0</td><td colspan="4">A</td><td>1</td><td colspan="3">B</td><td>0</td><td>C</td><td>*3</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="9">Starting sector address of
Index Set chain</td><td></td></tr>
<tr><td colspan="11">Reserved</td><td></td></tr>
<tr><td>0</td><td colspan="4">A</td><td>1</td><td colspan="3">B</td><td>1</td><td>C</td><td>*4</td></tr>
<tr><td colspan="11">Disc address of FTYPE entry</td><td></td></tr>
<tr><td colspan="11">Not used</td><td></td></tr>
<tr><td>1</td><td colspan="4">A</td><td>0</td><td colspan="3">B</td><td>0</td><td>C</td><td>*5</td></tr>
<tr><td>L</td><td colspan="10">Disc address of FORMS entry</td><td></td></tr>
<tr><td colspan="11">Not used</td><td></td></tr>
<tr><td colspan="11">Binary 0</td><td>*6</td></tr>
<tr><td colspan="11">Not defined</td><td></td></tr>
<tr><td colspan="11">Not defined</td><td></td></tr>
</table>

FOOTNOTES FOLLOW ON NEXT PAGE

Footnotes for Format Value Set Directory (on Disc)

ABC is the 3-digit ASCII name of the format, Index Set, Ftype, or Forms.

Value Set: 000 < ABC ≤ 999
bit 16 = 1 (Value Set)
bit 8 = 1 (Index Set)
bit 8 & 16 = 1 (Ftype)
bit 0 = 1 (Forms)

Formats, Value Sets, Index Sets, Ftypes, Forms, and empty entries can be intermixed in any order.

*1 - 3 word entry for format ABC
SD: 0=single accumulator width, 1=double precision accumulators
G: 0=old style formats, 1=new sytle formats
Screen Code: 0: Quarter screen, 1: Half, 2: Full, 3: Double

Screen code includes the message line. These are the minimum screen sizes for format.

*2 - 3 word entry for Value Set ABC

*3 - 3 word entry for Index Set ABC

*4 - 3 word entry for Ftype ABC

*5 - 3 word entry for Forms ABC

L = Lock bit. Forms cannot be deleted when this is on.
*6 - 3 word empty entry

84 entries per sector. The first word of each entry must contain a valid value.

FORMAT OBJECT CODE SECTOR

	Standard 4 word header -- fourth word is format name in ASCII
+4	Four 63 word pages of compiled calling sequences. page 0
+0103	page 1
+0202	page 2
+0302	page 3

Notes:

The code word is actually the ASCII format name. By the nature of ASCII code, this yields a code word with bits 0-3 = 0011.

Compiled calling sequences may not be split across pages. Therefore, several NOP instructions (06700000) may appear at the end of a page.

The last sector of the format may not be full, i.e., it may contain less than four pages. The remainder of the sector is garbage.

```
*________________________________________________________________________________
***    -W WORD IN FORMAT CODE CALLING SEQUENCES *********************************
*________________________________________________________________________________
*      0       Always 0 so that -W Word will not look like a BAL*
*      1       1=Auto skip
*      2       1=Auto dup
*
*      3       1=Generate an attribute for the field, 0=no attribute.
*
*      4,5     Field intensity (only if bit 3 = 1)
*                00 low intensity
*                10 high intensity
*                11 blank intensity
*
*      6       1=Right justify field
*      7       1=Must enter
*      8       1=Must fill
*
*      9-12    -Number of digits to right of decimal point in left zero field.
*              If not a left zero, field is all 1's.
*
*      13-23   -Width of field
*
```

VALUE SET SECTOR

	Standard 4 word header -- fourth word is Value Set name in ASCII
+4	Time Value Set was complied
+5	Day Value Set was complied
+6 thru +0377	String of Value Set entries

Notes:

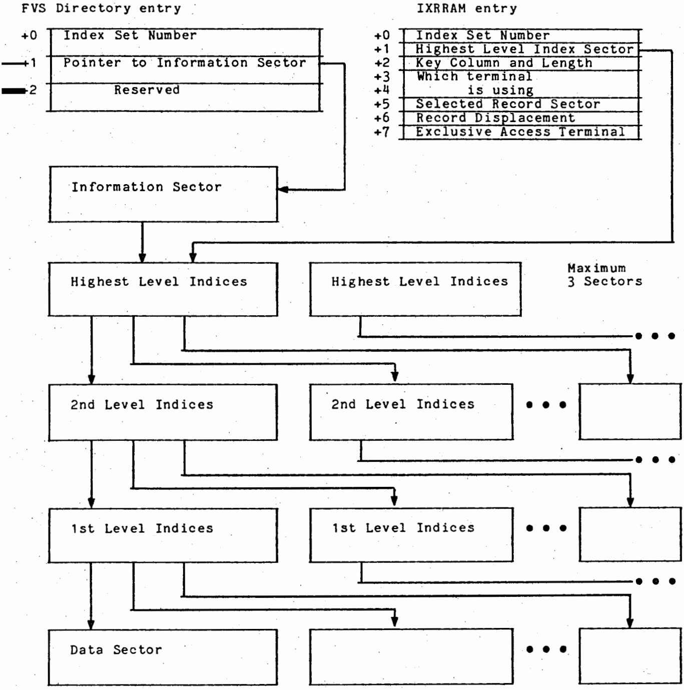
The code word is actually the ASCII Value Set name. By the nature of ASCII code, this yields a code word with bits 0-3 = 0011.

Each Value Set entry is 1 to 250 words long, the length being defined in the Format-Value Set Directory entry.

Each entry is a string of ASCII characters, starting on a word boundary, left justified with garbage fill. Entries may not span across sector boundaries.

Unused words at the end of a sector are filled with binary zeros. Unused words at the end of the last sector of the Value Set are filled with 037777777.

INDEX SET ORGANIZATION



INDEX SETS (on Disc)
Information Sector

	Standard 4 word header	
+4	Reserved	
+5	ASCII INDSET number with bit 8 on	
+6	INDSET number in binary	
+7	Sector address of HIGHEST index level	*1
+10	First column number of key	
+11	Number of columns in key	
+12	Job name of the data batch	
+15	Batch ID of the data batch	
+17	Sector address of first data batch sector	
+20	Sector address containing batch directory entry for data batch	
+21	Displacement of batch directory entry for data batch	
+022 . . . +0377	reserved	

INDEX SECTOR

		Standard 4 word header	
+4	L	Sector address of data batch sector	*2
+5 . . .		Key value of highest key in data sector or previous level index sector (left justified with blank padding to word boundary)	Index entry
. . +0377			

Footnotes for Index Sets

*1 Index level sectors are built by having an index entry for each sector in the data batch. The highest index level must be 3 sectors or less. Index levels will continue to be built from the data batch until a level with 3 or less sectors is reached. Each index level differs from the first only in that the sector address points to a sector in the preceeding index level.

*2 L = 1 if this is the last entry. The index entry length in words is $\frac{(KEYLENGTH - 1)}{3} + 2$

The key value is that of the last (highest) record in the sector.

Note that records cannot be inserted into an index beyond the highest key in the last sector.

File Access Data Structures (On Disc)

type entries: The entries have different formats depending on the type of file described, mag tape, sequential disc, etc. Each disc entry has some common information.

Common information:

+0	File type - 0=MT, 1=SD, 2=IX, 3=PR, 4=CR	
+1	Creator's Operator I.D. in ASCII	
+2	Creation time (T-86400)	
+3	Creation date	
+4	Creator's terminal number	
+5	Use depends on FTYPE (see below)	*1

*1 Magnetic tape files:

+5	Record in bytes, Binary
+6	Number of records per block
+7	0= 9 track, 1= 7 track, <0= Use CONFIG value
+10	0= Low density, 1= High density, <0= Use CONFIG value
+11	0= EBCDIC, 1= ASCII, <0= Use CONFIG value
+12	Deck number in Binary

Sequential disc files:

+5	1st word of job name in ASCII	
+6	2nd word of job name in ASCII	
+7	3rd word of job name in ASCII	
+10	1st word of batch name in ASCII	*A
+11	2nd word of batch name in ASCII	
+12	Not used	

*A =0: Will be specified at open time
<0: Will follow order chains

Index sets:

+5 thru +12	Not used

Printer files:

+5	Names of Forms Control Set in ASCII	*
+6	Line length in Binary	
+7 thru +12	Not used	

*B = 0: Will be specified at open time
=-1: Use default values

Card reader files:

+5	Card width in columns in Binary
+6	EOF mark, 3 ASCII characters
+7 thru +12	Not used

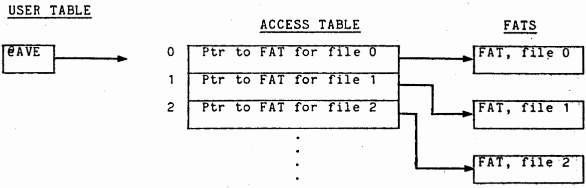
Forms Data Structure (on disc)

	0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3	
	Forward pointer (must be zero)	
	Back pointer (must be zero)	
	Here pointer	
	Forms name - 3 ASCII numbers with bit 0 on	
	Time of creation T-86400	PFTIME
	Date of creation, ASCII JULIAN	PFDAY
	ASCII Operator I.D. at creation	PFOID
	ASCII terminal number at creation	PFTERM
	No. of printed lines/page No. increments to advance/line Total no. of lines per page	*1 PF3IN1
	Channel number mask table Bit 12 is on if channel 12 is defined, Bit 13 is on for channel 11,... ..., Bit 23 is on if channel 1 is defined. (Zero if no channel definition table)	PFCMT
+6	Actual length of channel definition table (Maximum value of 48)	PFTLEN
	Channel definition table Bits 0-07: LINENO Bits 12-23: Each bit if for a channel no. Bit 23 is for channel 1, bit 22 is for channel 2, etc. The last word + 1 contains the following: Byte 0: Zero Byte 1: Form feed channel number Byte 2: Line feed channel number	PFTAB
	Length of the heading in words (Zero if no heading)	PFHLEN
	Length of heading in bytes	PFHLNB
	Off set of variables in page header (Byte/word offset- bits 0-1 are the byte #) (Bits 2-7 variable number, 0-010) There is one word for each of the following: #DAY, #TIMES, #OPER, #JOB, #BATCH, #RECNO, #CONST, #TERMNO, and PAGE NUMBER. This word is -1 if this variable is not used.	PFHVAR
	Page heading - ASCII	PFHEAD

*1 Bit 0 on means use PICA spacing

FILE ACCESS STRUCTURES IN RAM

File access information is kept in the File Access Table, (FAT). Every file which is open has a File Access Table which contains information about that opening e.g., current record number, record buffer location, etc.



FILE ACCESS TABLE

+0	File number, ASCII, with parity bit code	*1
+1	File type code binary	*2
+2	Status flags	*3
+3	Current record length, bytes, Binary	
+4	Maximum record length, bytes, Binary	
+5	Record number, Binary	
+6	Record program level, Binary	
+7	Pointer to record buffer and length code	
+10 thru +17	Use depends on file type	*4

*1 - See FVS Directory parity bits.
*2 - See File Access Data Structures on Disc.
*3 - Status Flags:

	Bit 0=1:	Opened for output.
	Bit 1=1:	Opened for input.
	Bit 2=1:	EOF.
	Bit 3=1:	EOB.
	Bit 4=1:	BOF.
	Bit 5=1:	BOB.
	Bit 6=1:	Variable length records.
(RECSEL)	Bit 7=0:	Record not selected.
	=1:	Record selected.
	Bit 8-23	Not assigned.

*4 See following pages.

*4 SEQUENTIAL DISC

+10	SCR pair for record buffer, 1st half.
+11	SCR pair for record buffer, 2nd half.
+12	Pointer to opening control block for BAM.
+13	LCR/SCR instruction for record buffer.
+14 thru +17	NOT USED

*4 INDEX SETS

+10	Index set number, ASCII, + Bit 8.		
+11	Record header for XD insert.		
+12	Pointer to IXRAM; 0 = No record selected.		
	0-1	2 13	14 23
+13	Byte num.	NOT USED	Word offset of key loc.
+14	Key length in bytes.		
+15	Save area for FX2.		
+16	IX set error flags; Bit 0 = 1 error from ovly		
+17	NOT USED		

*4 PRINTER

+10	SCR pair for record buffer if spooling; word 1.	
+11	SCR pair; word 2.	
+12	Pointer to OCB from BAM if spooling.	
+13	LCR instruction for record buffer.	
+14	Pointer to Forms Control Block.	*A
+15	Maximum record length in words.	
+16	SCR pair for control char at record buffer end.	
+17	SCR pair 2nd word.	

*A

PRINTER FORMS CONTROL BLOCK

PFBFNO	Forms number: =0: No forms >0: Forms with channel definitions <0: Forms with no channel definitions		
PFBHED	Pointer to the forms sector address where the heading may be found. (Will be zero if no heading)		
	Reserved		
	Reserved		
PFBL1R	Return address for level 1 routines		
PFBL2R	Return address for level 2 routines		
PFBL3R	Return address for level 3 routines		
PFBCNO	Channel number or channel bit or scratch word		
PFBLNE	Current line number for this page		
PFBCLN	Current line number for entire document, incremented for every write command except when CR is specified.		
PFBMLN	Maximum number of printed lines per page		
PFBPNO	Current page number		
PFBPSW	The original 8121 printer specification word		
PFBCMT	Channel number mask table Bit 12 is on if channel 12 is defined, Bit 13 is on if channel 13 is defined, ..., Bit 23 is on if channel 23 is defined. (Zero if no channel definitions)		
PFBEND	The LWA+1 of the channel definition table		
		8 15	16 23
PFBFED		FF Channel number (Zero when not defined)	LF Channel number
PFBCIN		Pointer into the channel definition table	
PFBCDT	Channel definition table (Maximum of 48 words)		

*4

CARD READER

+10	Card EOF marker in ASCII
+11	Bit 0 = 1 C.R. not accessed by this user
+12 thru +17	Reserved

*4

MAGNETIC TAPE

+10	Pointer to tape control block	*B
+11	Resource usage flags	*C
+12	Tape's private copy of record number	
+13 thru +17	Reserved	

*B MAG TAPE CONTROL BLOCK

```
TSTAT    | Status word returned by $TAPE                   |
TDECK    | AB                                            CC|
         |                   9                           23|
TBUF     |                   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD|
TREQ     | $TAPE operation request word                    |
TSROT    | Pointer to current record in block (SCR)        |
TDEST    | Pointer to current record in block (SCR)        |
TFAT@    |X                  GGGGGGGGGGGGGGGGGGGGGGGGGGGGGG|
TREC@    |                   MMMMMMMMMMMMMMMMMMMMMMMMMMMMMM|
TRECSZ   |                                                 |
TFLAGS   |0                                              TT|
TBNB     |WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW|
TBLKSZ   | Block size                                      |
TSVX1    | TAPEIT X1 save place                            |
TSVX2    | TAPEIT callers return                           |
TBLKF    | Blocking factor                                 |
TDRIVR   | Logical drive number                            |
```

O: = 1 Tape is open for input. Inhibits read if open output.

```
T:      Code conversion type     00 = EBCDIC
                                 01 = ASCII
                                 10 = 6 Bit
```

X: Tape deck is being rewound.

W: Open output then count of bytes output to current block.
Open input (blocking factor-relative record # in block).

---

*C MAG TAPE RESOURCE USAGE FLAGS

```
*       BIT             FUNCTION                                *
*                                                               *
*       0-3             Tape deck number relative to controller *
*                       Bit 0 = Deck 0, Bit 1 = Deck 1, etc.    *
*                                                               *
*       4-5             Controller type                         *
*         00            7 Track                                 *
*         01            800 BPI                                 *
*         10            1600 BPI                                *
*                                                               *
*       6               User is a tape translate table user.    *
*                                                               *
*       7-13            Reserved                                *
*                                                               *
*       14-17           Block buffer size (TBUF points to it)   *
*         1000          01000 word block buffer                 *
*         0100          00400 word block buffer                 *
*         0001          00100 word block buffer                 *
*         0000          No block buffer                         *
*                                                               *
*       18-19           Record buffer size (TREC@ points to it) *
*         10            0400 word record buffer                 *
*         01            0100 word record buffer                 *
*         00            No record buffer or is in block buffer  *
```

DISC SECTOR POOL AND REQUEST QUEUE

Pending disc requests are kept in two queues: DISCQ1 and DISCQ2. Completed requests are kept in the buffer pool, BFRLST. All queues are linked lists of 4 word blocks:

```
        DISCQ1 is the list of priority one requests; reads and non-standards
               requests.
        DISCQ2 is the list of priority two requests; writes.
        BFRLST is the list of sector buffers currently in the pool.
```

```
Word 0   |Pointer to next 4 word entry. (0 if last entry)|
         |-----------------------------------------------|
Word 1   |Q|N|A|R|S|E|  RC |  Buffer pointer.            |
         | | | | | | |     |  Contents depends on type of |   *1
         |0|1|2|3|4|5|6   8|15    request. (see below)  23|
Word 2   |W| Contents depends on type of request (see     |   *1
         | |                                       below) |
Word 3   |H| Disc Read/Write Request Caller (see below)   |
         | |                                              |
```

Word 1:

```
 Q   = 1 disc I/O request pending, 0=request complete
 N   = 1 non-standard request (see below)
 A   = 1 repeat the write request (a 2nd PUTREL/PUTHLD
       initiated before first PUTREL complete)
 R   = 1 multiple read requests exist for this sector
 S   = Submonitor restart flag. 0 = Restart submonitor, 1 = Don't restart.
 E   = Unrecoverable I/O error.  This bit may be set because of
       a reject (invalid sector address, etc.) or because of 8
       unsuccessful retries on a disc I/O error.  With this bit
       set, PRODSC will cause the error return to be taken on the
       initiating GETSEC, PUTHLD, etc.
 RC  = After 8 retries are counted, this 3 bit counter will over-
       flow setting the error bit (bit 5).
```

Word 2:

```
        W    = 1 write request, 0=read request
```

```
Word 3:  H    = 1 word is User Table address of PROKS caller
              = 0 word is address of Submonitor Control Block of level 8 caller
```

*1 Word 2 and the buffer pointer may have 3 different forms, depending on the type of request.

For standard read and write requests:

```
  0 1            3 4                                                 23
 |W|      0       | 20 bit relative sector address                    |
 Buffer pointer = pointer to "here" cell in sector buffer or if
 0 = buffer not yet obtained.
```

For non-standard requests, there are two formats. The first is for an overlay load:

```
  0 1 2          8 9          11 12                                  23
 |0|0| 7 bits     | number of   | relative starting sector address   |
 | | | reserved   | sectors -1  | of overlay in zone 000             |
 Buffer pointer = location of overlay area.
```

FOOTNOTES FOR DISC SECTOR POOL

The second is for single sector absolute sector I/O (TRAN50)

0	1	2 5	6 23
W	1	4 bit IDOS logical drive	18 bit absolute sector address

Buffer pointer = pointer to user provided buffer.

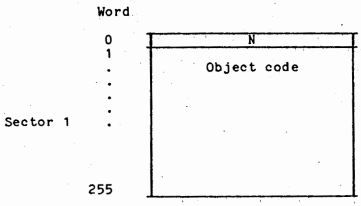
BFRLST NODES

Word 0	Pointer to next 4 word entry.		
	0	8	
Word 1	Not used	Pointer to "here" in disc buffer	*1
Word 2	Time of last operation (T-86400)		
Word 3	Pointer to user		*2

*1 "Here pointer":
0: No sector buffer.
<>0: Pointer to "here" word in sector.
Here* : If 0 then sector buffer is empty.

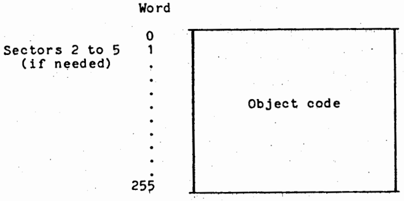
*2 Pointer to user:
0: No current user - sector is free.
>0: Pointer to submonitor control block.
<0: Pointer to user table + high order bits.
(This case is legitimate only when a request has been processed and the terminal has not yet been restarted).

OVERLAY OBJECT CODE



where N is the overlay number (1 to 99)

The 1 to 5 sectors are contiguous



OVERLAYS

Overlays are programs cataloged in the DAT000 file of disc drive 0 that load and execute in the overlay area of RAM. They perform infrequently requested functions (TOTAL key, REC-UP, screen print, new mode, etc.) and, because there is only one overlay area for all terminals to share, they are designed to execute rather quickly. An overlay can be called in several ways:

1. From the resident (TOTAL key, etc.).
2. From another overlay.
3. From a transfer program.

Overlays, regardless of how they are called, are initiated and terminated by the main Overlay Submonitor logic in S96-80. The Overlay Submonitor spends most of it's time waiting for overlay requests, which can be classified as priority overlay requests or terminal overlay requests.

Priority Overlay Requests

A request for a priority overlay is indicated by setting a single-word switch non-zero. There are three priority overlay request words:

FMOREQ	When non-zero, indicates a request for overlay 21 entry 1 to set up a system flashing message. FMOREQ is set by level 4 when a system message number is found in ECODE of a terminal's User Table.
PDOREQ	When non-zero, indicates a request for overlay 17 entry 1 to perform PAGDIR reorganization and Master and Maintainance Sector re-writes. PDOREQ is set by level 4 once every minute.
:REQ8L	When non-zero, it is a pointer to a list of overlay requests which are not associated with any terminal. This is used by the Communications Submonitor to request its overlays and by the Allocation Submonitor to request the overlay to do deallocation.

Terminal Overlay Requests

A request for a terminal overlay is indicated by storing an overlay request word in REQ8 in the terminal's User Table and incrementing the resident word OVRREQ (count of terminal overlay reqests). The format of an overlay request word is:

Q		X		Overlay #	entry #
1	3	1	2	8	8

Q: 0 = queue keystrokes during overlay execution
1 = toss keystrokes during overlay execution
X: 1 = special request

LEVEL 8 PROCESSING

Level 8 processing is controlled by a multi-tasking monitor that supervises t[illegible] execution of several tasks. These tasks, called submonitors, are each controlled by an 8 word Submonitor Control Block (SCB). The OVERLAY, PRINT, and ALLOCATION submonitor control blocks are always present. The DYNAMO, TRANSFER, and Terminal MSFP program submonitor control blocks, however, are dynamically built and linked to the other three SCBs when a call is made for that function controlled by the submonitor (i.e., when DEBUG or a transfer program is running). The SCB is removed from the linked list when the function is complete.

SCB

	0 8	9 23	
0		address of disc queue slot if sector buffer attached	
1	Submonitor state		*1
2		restart address	
3	Pointer to BAM work space block		
4	Use depends on state		*2
5	X1 restoration value		
6	X3 restoration value		
7	ASCII name of Submonitor	Pointer to next SCB (0 = no more SCB's)	*3

* 1 States:
- 0 - waiting for CPU
- 1 - running
- 2 - idle
- 3 - waiting for external restart
- 4 - waiting to call (not currently used)
- 5 - waiting to start disc I/O
- 6 - disc I/O complete
- 7 - waiting for PRINT submonitor

* 2 Displ. 4 word used:
- 4 - address of submonitor being called
- 5 - sector address
- 6 - disc queue entry address of non-standard disc request

*3

Submonitor Name	Title
O	Overlay Submonitor
P	Print Submonitor
A	Alloc/Dealloc Submonitor
D	Dynamo Submonitor
T	Transfer Submonitor
(?)	MSFP User - one character ASCII representation of the binary terminal #

DUMP PROCEDURE

. AUTO to MANUAL.

2. RESET then STEP.

3. LOAD into TIR - 71100001

. MANUAL to AUTO then to MANUAL.

   a. Tape Dump
      1. Mount a scratch tape.

      2. MANUAL to AUTO

The system will dump records 1024 words long (06000 bytes).

   b. Disc Dump
      1. Mount a disc which contains a DUMP47 file.

      2. MANUAL to AUTO
         The system will write all of memory out to the DUMP47 file.

5. When the dump is complete, the system will halt with X3=00000000. If low memory is clobbered and the 71100001 will not execute a dump, the address of the dump routine is also in @RAMDM. Find the address of @RAMDM in your load map and load the contents of that location into location 1.

   To get a formatted dump:

```
// DUMPV
/INPUT=filename@drivenumber (use indicated input instead of DUMP47@0)
/TAPE = TAPE7 or TAPE8 or TAPE16 or T7 or T8 or T16 (dump is on tape
                                                          as indicated)
/ANALYSIS (analysis only, omit RAM)
/RAM      (RAM dump only -- no analysis)
/LOW=l.   (start RAM part of dump at location l)
/HIGH=h   (stop RAM part of dump at location h)
```

   The following parameters only apply to IV/90 MOD II dumps:

```
/PHYSICAL (Print memory; memory with no mapping)
/WINDOW=w (Use window w for mapping. Default is window in effect at
          the time of the dump.)
//
```

   DUMPV requires a 0200 or 0440 (MODII) sector contiguous file called DUMP47. This can be created by running the program MAKD47:

```
// MAKD47
    8230
/0= 8240  @ n
    8260
/BANKS=2  optional (creates 0440 sector file for MODII dumps)
//
```

DYNAMO

DYNAMO is a dynamic memory display, patch, and trace routine that is invoked Mode D. The DYNAMO display occupies the bottom two lines of the Debug displa screen.

DYNAMO has the following features:

Display the contents of any RAM location in octal or ASCII.
Patch RAM, 1, 2, 3 or 4 words at a time.
Place a STOP command at any instruction in RAM.
Place a TRACE command at any instruction in RAM.

Getting started:

When VISION is up and running, first hit the unshifted HOME key to start DYNAMO at the Debug screen. The Debug screen has four lines of dynamic system status information at the top of the screen and DYNAMO uses the bottom two lines. The message line should have a solid block cursor at the left end of the five position location field. The left most 3 places on the line should have "CHG" indicating change mode.

MESSAGE LINE:

```
XXXccclllll aaaaaaa bbbbbbb ccccccc ddddddd
```

where:
XXX is CHG - change mode, enter a location into the lllll field.
DIS - octal display mode, the contents of lllll are displayed at aaaaaaa, the contents of lllll+1 are displayed at bbbbbbb, etc.
ASC - ASCII display mode, aaaaaaa etc is ASCII representation of location lllll. 36 bytes are displayed.
PAT - patch has been made beginning at location lllll.
TRC - a trace or stop has been planted at location lllll.

ccc is count of the number of times the instruction at location lllll has been executed in trace or stop mode.

lllll is the current location in RAM being displayed, patched or traced.

Status Line:

```
YYY RARARARARA RBRBRBRBRB X1X1X1X1X1 X2X2X2X2X2 X3X3X3X3X3
```

See below.

To Display RAM:

Press the HOME key.
Key the 5 octal digit RAM location.
Press the INDEX (F5) key.
The contents of the location will display at aaaaaaa (just to the right the location). Four words are displayed.
To see the contents of the word before location lllll (lower RAM),
Press LEFT arrow.
The four words will display automatically.

To see the contents of higher RAM locations,
Press RIGHT arrow.
The four words will display automatically.

To see the contents of the contents of location lllll (indirect),
Press VALID (F4) key.
The address portion of aaaaaaa is used as the new lllll.
The four words will display automatically.

To restore lllll after an indirect,
PRESS ? (F3) key.
The location before the last indirect is restored and the 4 words display automatically.

To display in ASCII.
PRESS ASD (F1) key.
36 bytes (12 words) beginning at location lllll are displayed.

To STOP Execution At A Specified Location:

The TRC mode is used to stop execution at a specified location.

The stop occurs before the instruction is executed. So you can check the condition codes before a branch, etc.

NOTE: This cannot be used for instructions that are executing at interrupt level 0, 1, 2 or 3 if you want to be able to cancel the STOP. To be able to cancel a STOP the system must be able to process a level 3 interrupt.

Press HOME key.
Key the 5 octal digit location.
Press PROG6 (F11) key.

TRC will appear at the left end of the message line.
STP will appear at the left end of the status line.

When the stop occurs, the status line will show the contents of registers RA, RB, X1, X2 and X3 in octal from left to right. At the far left end of the status line the 3 byte display Scc will display. cc is the octal condition code at the time of the stop. The count ccc on the message line should show 001 the first time.

To cancel the stop and allow the system to continue press RESET (TAB).

To release the stop and catch it again the next time press PROG6.

To Look Around RAM While STOPPED:

When the system is stopped, you can display RAM using the instructions in Section 3. Remember that if you look at the location where the STOP is, you will see the BRM instruction that intercepts execution. In addition the PATCHing of memory or disc can be done while STOPPED. At any time pressing the RESET key will cancel the STOP.

To TRACE Execution At A Specified Location:

The TRC mode is used for tracing. Only one trace can be active.

Press HOME key.
Key the 5 octal digit location.
Press PROG5 (F10) key.

TRC will appear at the left end of the message line.
GO will appear at the left end of the status line.

When the instruction execution occurs, the status line will show the contents of registers RA, RB, X1, X2 and X3 in octal from left to right. At the far left end of the status line the 3 byte display Scc will display. Scc is the octal condition code at the time of the trace. The count ccc on the message line should show 001 the first time and the count up in octal each time the instruction is executed.

To cancel the trace press RESET (TAB).

NOTE: Do not trace instructions that are modified by the system. The debug system removes the original instruction and puts in its place a BRM instruction into the DYNAMO package. The original instruction is executed inside DYNAMO. When RESET is pressed the original instruction is restored.

To PATCH RAM (4 or fewer consecutive words)

Press HOME key.
Key the 5 octal digit lowest location to be patched.
Press the INDEX (F5) key.

The four locations beginning at 11111 will display.
Verify that you are at the correct spot in RAM.

Press HOME key.

Press the RIGHT arrow to position the solid cursor to the octal digits to be changed and key the correct octal digits. The actual changes are not made to RAM until the next step is performed.

Press PROG1 (F6), PROG2 (F7), etc. to patch 1, 2, etc. words.

PAT will display at the left-most end of the message line.

Performance Monitor

ISION may be OPTIONed to included a software Performance Monitor which can be sed to monitor system activity. The report of system activity that is output from the Performance Monitor includes a detailed analysis of such things as keystroke activity, record activity, format page activity, value set/index set activity, disc I/O activity, overlay activity, and Level 7/Level 8 activity.

The two $SETPP subcommands, STARTPM and STOPPM, control the Performance Monitor activity.

STARTPM

STARTPM initiates the Performance Monitor. Any previous Monitor activity is terminated and new statistics begin to accumulate. After the Performance Monitor has been allowed to run for a period of time, the STOPPM subcommand is used to report the results (see below). There are three optional operands that may be specified in parenthesis following STARTPM: OVLOG, RPLOG, and KEYSTROKE.

OVLOG

VLOG specifies that a detailed log of overlay calls, by overlay, is to be aintained in RAM. When STOPPM is executed, a report is printed showing the umber of calls and the mean number of calls per minute for each overlay. A maximum of 64 overlays may be logged.

RPLOG

RPLOG specifies that a detailed log of resident activity is to be maintained in RAM. VISION will be divided into 0100 word blocks of resident code. Then, at each Level 4 interrupt, RP will be sampled to determine which block of resident was executing at the time of the interrupt. When STOPPM is executed, a report is printed showing, for each block of resident logic, the number of times the block was found active and its percentage of CPU utilization. A LOADOV map from OPTION is necessary in order to interpret the results.

KEYSTROKE = software code for control key (modulo 0200)

KEYSTROKE specifies that the keystroke corresponding to the specified software code is to be monitored. When STOPPM is executed, the report will show the number of times the monitored keystroke was processed and its percentage of total keystrokes. The software code for the control key must be specified modulo 0200 (0203 should be specified as 003). If not specified, the default is 000 - record up.

Examples:

```
$SETPP    STARTPM
$SETPP    STARTPM (OVLOG)
$SETPP    STARTPM (KEYSTROKE=205)
$SETPP    STARTPM (RPLOG, OVLOG, KEYSTROKE=0205)
```

STOPPM

The STOPPM command terminates the Performance Monitor and initiates the report writer transfer program to report the results. There are three optional operands that may be specified in parentheses following STOPPM: TITLE, REPORT, and NOREPORT.

PERFORMANCE MONITOR - CONTINUED

TITLE = "...48 Character Identifying Title..."

The specified title will be printed on the first page of the Performance Monitor report. The title should identify the installation and/or conditions of the test; e.g., TITLE="ALL OPERATORS IN VERIFY".

REPORT = Jobname, Batch ID

This operand directs the output of the Report Writer to the specified job and batch. The job must be defined with 133 byte records and the specified batch must already exist. If not specified, the output of the Report Writer is directed to the System Printer.

NOREPORT

NOREPORT specifies that no report is to be produced; i.e., just terminate the Performance Monitor, but do not initiate the Report Writer.

EXAMPLES:

```
$SETPP    STOPPM
$SETPP    STOPPM (NOREPORT)
$SETPP    STOPPM (REPORT=MASTER133,1)
$SETPP    STOPPM (TITLE="ABC", REPORT=X,1)
```

NOTES:

STARTPM and STOPPM may be used in $SETPP with LEV78B:

```
$SETPP LEV78B=12, STARTPM
$SETPP STOPPM (TITLE="XYZ"), LEV78B=3
```

The Performance Monitor should be initiated only after the system has reached its normal activity load; i.e., after all operators have signed on and begun work in a batch. The Performance Monitor should be allowed to run long enough to gather meaningful statistics, probably from a minimum of three minutes to a maximum of about ten minutes. Statistics are accumulated in whole-minute intervals, i.e., if the Performance Monitor is started at 55 seconds, the statistics will reflect system activity for the entire minute.

PERFORMANCE PERAMETERS

LEV78B - Number of terminals that PROKS will attempt to process before returning CPU to submonitors. (LEV78B = N, 0<N≤32)

NKSMIN - Maximum number of keystrokes to process out of a terminal's queue at one time. (NKSMIN = N, 0<N<5)

DYNBAL - Causes system to perform dynamic adjustment to the system's level78 balancing. (DYNBAL = N, 0<N<8, or DYNBAL = NO)

PAGLIF - Sets the age at which an unused page will be discarded by the garbage collector. (PAGLIF = N, 0<N<8 where the "life expectency" is 2 to the Nth, or PAGLIF = NO)

PAGSEC - Causes the system to read into memory more than just the first page in a sector of format object code. (PAGSEC = N, 0≤N≤3 where N is the number of seconds to advance the "time" word in the page directory entry.)

BUFBLK - Enables the system to break down N sector buffers into system blocks. (BUFBLK = N, where N can't be greater than 1/4 the number of sector buffers.)

Note: Show mode K will display the current parameter settings.

3270 ATTRIBUTES AS DISPLAYED ON VIDEO

BIT 6 = 0 ALWAYS			ALPHA BIT 3 = 0		NUMERIC BIT 3 =1	
	DISPLAY	BITS 4&5	UNPROTECTED BIT 2 = 0	PROTECTED BIT 2 = 1	UNPROTECTED BIT 2 = 0	PROTECTED BIT 2 = 1
	NORMAL NON-SEL	00	@	(0^s)	P^s	p
MDT-OFF	NORMAL SEL	01	D^s	d	T^s	t
BIT 7 = 0	HIGH SEL	10	H^s	h	X^s	s
	NON DISPLAY	11	L^s	1	X (Multi-X) 0134	(Multi-X^c) 0174
	NORMAL NON-SEL	00	A^s	a	Q^s	q
MDT-ON	NORMAL SEL	01	E^s	e	U^s	u
BIT 7 = 1	HIGH SEL	10	I^s	i	Y^s	y
	NON DISPLAY	11	M^s	m	$(-^s)$ 0135	$(*^c)$ 0175

SINGLE WORDS IN RAM

These are counters, pointers, addresses, etc., used by VISION to keep track o itself. A description of each word follows:

SYMBOL	DESCRIPTION
ACTO@	Address of the pointer to the Debug Display Area. This is present in a system block on production systems or on a screen on Debug systems.
ASCHR	Current system hour in ASCII, in the form bHH.
ASCMIN	Current system minutes in ASCII, in the form :MM.
ASCSEC	Current system seconds in ASCII, in the form :SS.
ASS4	Address of the last 4-word block assigned.
ASS16	Address of the last 16-word block assigned.
ASS32	Address of the last 32-word block assigned.
ASS64	Address of the last 64-word block assigned.
BATACT	Mask to inhibit the use of any active batch (see state request bits in the batch directory disc format).
BFRLST	Pointer to the start of the sector/buffer list.
BLKCNT	Magnetic tape block count.
CHAR	Last level 7 character.
CHARV	The most recently typed in character in verify mode.
CLOCK	Fires level 4 every 1/5 of a second. Set to -12 at level 4 and incremented by level 0 every 1/60 second.
CLOCKS	One second clock.
CNFIG	Disc address of CONFIG sector.
COMMWD	3270 CONFIG word.
D	Debug level where zero = production/system, non-zero = screen number of Debug display.
DAY	Day number, ASCII
$DBASE	Starting sector address for VISION.
DISCQ1	Pointer to FWA of primary disc request queue.
DISCQ2	Pointer to FWA of secondary disc request queue.
DSETIM	Time of last disc error (T - 86,400).
ECTAB	Counter for all disc errors
EORREQ	Number of end-of-record requests outstanding.
EOVDAT	First data sector of $EOV tape label data.
F90M2?	0 = Not a MOD II, 1 = MOD II.
FETREQ	Number of page fetch requests outstanding.
@FV	New location of Format Vector Table.
FVLEN	Current length of the Format Vector Table.
FVSDIR	Disc address of FVS directory.
FVSRAM	Pointer to the Format/Value set directory in RAM.
HDRDAT	First data sector of $HDR tape label data.
HOUR	Binary system hour.
IO6CNT	IO6 chip failure count.
ISVER3	0 = no index sets, -1 = index sets
IXRRAM	Pointer to the Index Set directory in RAM.
JOBDIR	Disc address of job directory.
JOBRAM	Pointer to the job directory in RAM.
KBCINP	The input character, at level 7, before translation.
KBLOG	The backlog count of keystrokes.
LASTZN	Highest zone in the system.
@LDTAB	Pointer to logical device table.
LL	Screen line length in words - 020 or 040.
LL2	LL + LL.
LOGBAT	The sector address of the log file batch.
LOGJOB	Jobname/batch number of the log file.
....+1	Jobname/batch number of the log file.
....+2	Jobname/batch number of the log file.
....+3	Jobname/batch number of the log file.
....+4	Jobname/batch number of the log file.
LOGSEC	Current log sector.
MAGTAPE	Count of magnetic tape errors.
MAXRAM	Maximum RAM available; determined by initialization.
MINUTE	Current time in minutes.
MP3270	Mapword for selected 3270 screen. 0 = none.
M#OVLY	Highest numbered overlay
N	Count of the number of screens.
NDS	Number of disc drives.

SINGLE WORDS IN RAM - CONTINUED

SYMBOL	DESCRIPTION
NEX4	Address of the next 4-word block available.
NEX8	Address of the next 8-word block available.
NEX16	Address of the next 16-word block available.
NEX32	Address of the next 32-word block available.
NEX64	Address of the next 64-word block available.
NEX256	Address of the next 256-word block available.
NLGSEC	Count of the sectors in the log file.
NPRSEC	Count of the sectors in the print queue.
NSB	Number of screens and number of buffered printers.
OHTOP	Pointer to BAM OCB header chain.
OIDDIR	Disc address of OID directory.
OSOPBD	Operator statistics batch directory sector.
OSOPB1	Operator statistics first data sector.
OVRREQ	Number of overlay requests outstanding.
P2L2P@	Pointer to P2L2P Table.
PAGDIR	Pointer to the page directory in RAM.
PASSWJ	Job define password.
PASSWP	Print/log password.
PASSWS	Supervisor password.
PFCB@	Location of the pointer to the submonitor control block.
PRTACT	Print active/idle flag.
PTHING	Pointer to the system constant (entered during system bring-up).
RESREQ	2770 message count.
SEC95P	95% full point in sectors available.
SECAVL	Total sectors available on all drives.
SECNBR	Number of current level 7 sector.
SECPTR	Pointer to the current level 7 sector buffer.
SECEND	Pointer to the end +1 of the current level 7 sector buffer.
SEMOH	BAM flag.
SRA13	Save RA and X1 for level 3.
....+1	Save RA and X1 for level 3.
SYSFIG	Configuration word for the system (CONFIG sector word 0116).
SYSFG1	Additional configuration information (CONFIG sector word 0117).
SYSFG2	Not used.
SYSFG3	Not used.
SYSNBR	Current system sector in RAM.
SYSPTR	Pointer to the system sector buffer in RAM.
SYSEND	Pointer to the end +1 of the system sector buffer.
TENSEC	Ten second clock.
THING	The system constant area.
....+1	The system constant area.
....+2	The system constant area.
....+3	The system constant area.
TIME	Time = seconds since midnight - 86400.
TIMELW	The master dir. sec. last written to disc.
TP256	Address of next TP area 256-word block available.
USER	Last level 7 user table address.
USER8C	Current level 8 user table address
VOLCNT	Tape volume reel count.
WHAT!	Interrupts unexpected.
ZONTOT	Kept in word 0200 of the master directory sector. Byte 0; Not used, Byte 1: Total zones -1; Byte 2: Largest Zone number.
@$MAXL	Pointer to the size of the LDTAB device.
@OVDIR	Address of overlay directory.
@ZONPT	Address of zone pointer table.
:REQ8L	Pointer to top of system REQ8 list.
$LDINV	Required by IDOS, E3, and above.
SCQG	First entry in submonitor control block.
SCQP	Pointer to last entry in control block.
SCQT	Address of current submonitor control block.
PRSTAT	Printer status word.
LPOUT	Printer selection value.
DIABCF	Address of start of 8121 complete-flags.
DIABND	Number of 8121 printers.
DIABP2	Address of basic 8121 driver, PRIN22.
DIABSU	Unit number of the 8121 sysprint. If the system printer is not an 8121, THEN DIASSU IS -1.

SINGLE WORDS IN RAM - CONTINUED

SYMBOL	DESCRIPTION
DISCRQ	Count of disc requests pending.
DISCLK	Non-zero means channel 2 locked-out for disc, in use by tape.
CHAN2	Zero means channel 2 is idle.
CHAN2R	Non-zero means Level 8 is requesting channel 2.
DISCIO	Non-zero means a disc operation is going.
REQTAB	$JDISC request table.
....+1	$JDISC request table.
....+2	$JDISC request table.
....+3	$JDISC request table.
....+4	$JDISC request table.
....+5	$JDISC request table.
?IGP00	Used to count devices serviced during general poll loop. Initially set to -32.
?IGP01	Last device to use status or TX subroutines starting device number for general poll.
?IGP05	Indicates a device has requested transmission or has changed status.
?IPG06	Second/10 to delay the transmission or EOT during general poll if no devices have changed status or requested transmission.
?NAKS	Count of NAKs sent.
?BFCNT	Buffers pending between level 1 and 7.
?DSRER	Number of Data Set Ready errors.
?OLCNT	?BFCNT served for retransmission.
?REPLY	Text mode response from Host.
?4EROR	Error flag set by level 4.
?ACKOT	Next ACK to be output.
?ACKIN	Expected ACK in.
?ICPSL	Control mode flag from level 1 to level 7.
?7SYNC	SYNC received in level 7.
?7TASK	Current CU3270 task.

RELEASE BA04
MEMORY LAYOUT
(Physical layout for extended memory systems)

0	Interrupt Locations	
060	Screens	Bank 0 1st 96 K
077777		

(EXTENDED MEMORY)

0104,000*	User Tables, System Blocks, Sector buffers, etc. Disc Request Queue	Bank 1 2nd 96 K
	Zone Pointer Table	
	Overlay Directory	
$DATA	Optional Resident	$DATA Resident loads with /P=LO from PUSER down
	Fixed Resident	
@FV		@FV
0167200	Format Vector Table	
0167300	IOID5 (8121 Print Output)	
0167400	PUSER (Keyboard Input)	
	Overlay Area	
0172000	Transfer Program Area or Sector buffers for PROKS	
0177777		

*Note: The first two pages of fast RAM (Bank 1) are not utilized on standalone Mod II systems.

RELEASE BA04
USER TABLE SKELETON DEFINITION

For non-MFE systems, the first seven words of the user table is a short routi that is executed by level 3. The routine calls IKEY to input a keystroke and then places it in the user table's keystroke queue. For MFE systems, MFE passes the keystroke to the VISION level 3 routine which places that keystroke in the keystroke queue for the appropriate terminal.

The keystroke queue consists of single word entries; the first byte of each entry is the (unconverted) keystroke that was entered and the remaining two bytes are the address of the next word in the queue for that user table.

```
USERN    BSS  1        Entry to KEYS routine.
USERN1   BRM  IKEY    Get key input.  (Non-MFE systems, only.)
USERN2   STB* KQBOT   Add keystroke to
USERN3   STB  KQBOT     bottom of the keystroke queue.
USERN4   LDB  LV3SVB  Restore RB.
USERN5   INR  KQLEN   Increment keystrokes in queue.
USERN6   BRD  USERN   Return and debreak.
*
KQLEN    EQU  7    Number of keystrokes in the queue.
KQTOP    EQU  010  Top of keystroke queue; always contains one dummy.
KQBOT    EQU  011  Bottom of keystroke queue.
KQUP     EQU  012  An "UP  KQTOP" instruction to get a keystroke.
*
         PZE  0    Lost keystroke routine.
         BRM  KEYL
         BRD  $-2

FMADR    EQU  016  Pointer to flashing error message block.
SCR      EQU  017  Execute to place characters on screen.
FWA      EQU  020  First word of screen.
SLWA     EQU  021  Last word + 1 of screen.
PSROT    EQU  022  Cursor SCR first operand.
DEST     EQU  023  Cursor SCR second operand.
SCOL     EQU  024  Cursor col id = - number of columns from right end.
WPO      EQU  025  Current fields governing ATTR during WGT call.
FSROT    EQU  026  (Pair) pointer to start of current field.
FDEST    EQU  027  First word of current field.
WPA      EQU  030  Extra LCR/SCR word pair.
WPA1     EQU  WPA+1
WPB      EQU  032  Extra LCR/SCR word pair.
WPB1     EQU  WPB+1
WPC      EQU  034  Extra word pair for scratch.
WPC1     EQU  WPC+1
WPD      EQU  036  Scratch word pair.
WPD1     EQU  WPD+1
VLROT    EQU  040  LCR pointer for verify.
VDEST    EQU  041
FX2      EQU  042  X2 = Location in format (via BAL).
FX3      EQU  043  X3 = - (columns remaining in field).
L7SVX2   EQU  044  Save X2 during level 7 interrupts.
L7SVX3   EQU  045  Save X3 during Level 7 interrupts.
MODE     EQU  046  24 flags. (Definedlater).
OPPTR    EQU  047  Pointer to operator directory.
JOBDES   EQU  050  Pointer to job descriptor table.
FORPTR   EQU  051  Pointer to current format.
REQ8     EQU  052  Request for LEVEL 8.
RECLOC   EQU  053  Location of current REC in sector.
WIDTH    EQU  054  - (Width of field).
MODPTR   EQU  055  Pointer to in-front modifier.
LCR      EQU  056  Execute to get character from old record in key verify,
                     LCR = SEC ADR for previous REC
ACCP1    EQU  057  Pointer to first four accumulators.  0-3.
ACCP2    EQU  060  4-7.
ACCP3    EQU  061  8-11.
ACCP4    EQU  062  12-15.
ACCP5    EQU  063  16-19.
ACCP6    EQU  064  20-23.
NFOR     EQU  065  Number of formats.
NVAL     EQU  066  Number of value sets.
```

```
MFWA      EQU  067  Printer to message line, BIT0=1: lines are off screen.
MSIZE     EQU  070  - Size of screen including dead areas.
MSIZEB    EQU  071  - Size in bytes.
KBTYPE    EQU  072  Keyboard descriptor word from CONFIG.
LASTKS    EQU  073  Last three keystrokes.
TVUSER    EQU  074  Trail verify user pointer
TERMN     EQU  075  Terminal Index 0-31.
BATCHN    EQU  076  ASCII batch ID - 6 characters - even boundry.
BACHN1    EQU  077  Continued.
BINDOC    EQU  0100 Binary document no.
TERMAS    EQU  0101 Terminal index in ASCII, blank fill.
BATCH     EQU  0102 Address in batch directory.
LINKBS    EQU  0103 Return address from LINKVS.
BINREC    EQU  0104 Binary record number.
MAXREC    EQU  0105 Largest binary record number.
NEXTUR    EQU  0106 Unconditional keystroke return address.
STATS     EQU  0107 Operator statistics table.
STATS1    EQU  0110 Second word of operator statistics table.
STATS2    EQU  0111 Third word of operator statistics table.
BINCOL    EQU  0112 Binary column counter.
ALARM     EQU  0113 Audible alarm flag.
SAVPTR    EQU  0114 Save pointer for backspacing.
MODEXT    EQU  0115 Mode extension flag word.(Defined later).
BLOCK7    EQU  0116 Terminal block word.(Defined later).
VRECB     EQU  0117 Pointer to verify record buffer.(Defined later).
RECHDR    EQU  0120 Current record header.
RECSIZ    EQU  0121 Record size.
SECTOR    EQU  0122 Current sector number.
NEXTCR    EQU  0123 Return word from BRM NEXTC.
COLUMN    EQU  0124 Pointer to column count on screen.
ECODE     EQU  0125 Error code. (Defined later).
QUE8RA    EQU  0126 Terminal restart address when overlay completed.
DQSLOT    EQU  0127 Current position in disc queue.
OV36BL    EQU  0130 Multi-mode block address.
FMTST1    EQU  0131 First 24 bits of format backspacing stack.
FMTST2    EQU  0132 Second 24 bits of format backspacing stack.
MAXDOC    EQU  0133 Largest binary document no.
MODE3     EQU  0134 24 single bit flags.(Defined later).
LASSEC    EQU  0135 Last sector in chain.
@BUF      EQU  0136 Pointer to format workspace.
@AVE      EQU  0137 Pointer to user's file access table pointers.
CURC      EQU  0140 Counts Level 4 until cursor change needed.
RS        EQU  0141 Digits to right of DEC point LZERO FIELD.
SCPBAD    EQU  0142 Used by screen print.
KV        EQU  0143 Byte 0 = SCRAM CHAR, 9-23 = RECLOC for prev. REC.
MODE2     EQU  0144 More single bit flags, like 'MODE'.(Defined later).
CURSES    EQU  0145 Three cursor characters (depends on field intensity).
STATS3    EQU  0146 Fourth word of operator statistics table.
@AT       EQU  0147 PTR to word containing current field ATTR.
ATMASK    EQU  0150 Mask for MDT in current ATTR.
MAXCOL    EQU  0151 Highest BINCOL so far in format.
SX3       EQU  0152 Save word used by -5C.
CUR       EQU  0153 Off screen copy of blinking word.
F1GA@     EQU  0154 PTR to field one physical governing ATTR.
MBLOCK    EQU  0155 PTR to message line system block.
                      BIT0=1: Lines are in home position.
COMM      EQU  0156 Communications word. (Defined later).
@SCB      EQU  0157 MSFP pointer to this terminal's submonitor.
```

```
WINDOW   EQU  0160 See below.
SUBSTK   EQU  0161 Subroutine stack for format compiler.
EDTPTR   EQU  0162 Pointer to edit command parameter list.
SPARE2   EQU  0163 Spare user table cell available.
```

WINDOW: terminal's window number.

bit	contents
0	not used
1-8	window number
9-13	logical page number
14	odd parity
15	1=read only
16-18	Bank number
19-23	physical page

TERMN: terminal printer information.

bit	contents
0	1=terminal currently printing; 0=not printing
1-2	not used
3	0=terminal unbuffered 1=terminal is buffered printer
4	0=control dup print request 1=370 print request
5-11	not used
12	1=screen print not allowed at this terminal
13	1=terminal printer is assigned
14-18	Diablo printer unit # assigned to terminal
19-23	binary terminal number

RELEASE BA04
CONFIG SECTOR (Sector 3 in Zone Header)

+0	Forward Pointer (Points to Keys Sector)	
+1	Backward Pointer (00000002)	
+2	Here Pointer (00000003)	
+3	Code Word (00000000)	
+4	Time of Last Change (t-86400 Seconds)	
+5	Day of Last Change	

+6	0 KB	1 AUD	2 5 -Reserved- (4 Bits)	6 12 Size of Screen in Words/16 (7 bits)	13 23 FWA of Screen /16 (11 bits)	*1

+046	VISION P2L2P Table - Maps devices to screens (32 words)		*2
+106	MAXMFE 8 bits	Numbers of disc buffers	*3
+0107	Reconfiguration Password (Scrambled)	Default: 8 Blanks	
+0110	Reserved		
+0111	Print/Log Control Password (Scrambled)	Default: 12345678	
+0112	Supervisor Password (Scrambled)	Default: PASSWORD	
+0113	Reserved for Special Systems (3 words)		

+0116 CONFIG SYSFIG	0 VIDEO	1 EXTMEM	2 CH	3 LAMLOG	4 CR	5 PR	6 MAG	7 DEN	8 SOR	9 CCB	10 DOCS	11 BEEPER	12 CCODE	13 LAM2	14 RESRVD	15 17 DUMP TYPE (3 BITS)	18 19 NO. OF TAPE DRIVES (2 BITS)	20 23 NO. OF DISC DRIVES (4 BITS)	*4

+0117 CNFIG1	0 RECPRT	1 ERP	2 FIND	3 CFPROP	4 REVFY	5 BATCH	6 RESRVD	7 RESRVD	8 9 10 TAPE BUFFER SIZE (3 BITS)	11 12 13 RESRVD	14 15 16 17 18 NO. OF 8121 PRINTERS (5 BITS)	19 20 21 22 23 NO. OF TERMINALS (TRUE VALUE) (5 BITS) IS 32 TERMS WHEN = 0.	*5	

CONFIG SECTOR - CONTINUED

<table>
<tr><td>+0120</td><td colspan="6">Reserved</td><td></td></tr>
<tr><td>+0121</td><td colspan="6">Reserved</td><td></td></tr>
<tr><td>+0122</td><td>RESERVED (3 Bits)</td><td>MAXBUF (4 Bits)</td><td>DYNBAL (3 Bits)</td><td>NKS (2 Bits)</td><td></td><td>LEV78B (5 Bits)</td><td>*6</td></tr>
<tr><td>+0123
CNFIG4</td><td colspan="6">No. of Buffered Printers</td><td>*7</td></tr>
<tr><td>+0124
CNFIG5</td><td>0 1 2 3 4 5 6 7 8 9</td><td>10
L
C</td><td>11 12 13</td><td>14 15 16 17 18
GP DELAY
(5 BITS)</td><td colspan="2">19 20 21 22 23
CONTROL UNIT
ID (5 BITS)</td><td>*8</td></tr>
<tr><td>+0125</td><td colspan="4"></td><td colspan="2">COMWT8 (7 BITS)</td><td></td></tr>
<tr><td>+0126</td><td colspan="6">Reserved for Future Use (10 words)</td><td></td></tr>
<tr><td>+0140</td><td colspan="6">Jobname of Current Log File (3 words)</td><td></td></tr>
<tr><td>+0143</td><td colspan="6">Batch No. of Current Log File (2 words)</td><td></td></tr>
<tr><td>+0144</td><td colspan="6">PRSTAT</td><td>*9</td></tr>
<tr><td>+0146</td><td colspan="6">8121 Printer Specification Words (16 words)</td><td>*10</td></tr>
<tr><td>+0166</td><td colspan="6">User Table TERMN (32 words)</td><td></td></tr>
<tr><td>+0226</td><td colspan="6">Spool Parameters (32 words)</td><td></td></tr>
<tr><td>+0266</td><td colspan="6">Spool Conversion Table - CONTBL Table (17 words)</td><td></td></tr>
<tr><td>+0307</td><td colspan="6">$COMM Controller Assignment Table (CAT) 13 words</td><td>*11</td></tr>
<tr><td>+0324</td><td colspan="6">Reserved for Future Use (11 words)</td><td></td></tr>
<tr><td>+0337</td><td colspan="6">Reserved for Users (33 words)</td><td></td></tr>
</table>

*1 KB: 0=Source, 1=029
AUD: 0=No Audible Alarm
32 Words - One for each possible keyboard device

*2 VISION P2L2P Table
Byte 0: Reserved
Byte 1: Maps devices to screens (index into with device no.)
Byte 2: Maps screens to devices (index into with screen no.)

*3 MAXMFE = 0, non-MFE systems; MAXMFE = Maximum number of terminals signed on to VISION (MFE systems)
If # screens ≤ 10; # buffers recommended = # of screens. Min. is 6.
If # screens ≥ 10; # buffers recommended = # of screens /2 +2. Min. is 10.

ADDITIONAL FOOTNOTES FOR THE CONFIG SECTOR FOLLOW ON NEXT PAGE

RELEASE BA04
FOOTNOTES FOR CONFIG SECTOR - CONTINUED

*4 CONFIG (SYSFIG)
Video: 0=48, 1=81
EXTMEM: 0=None, 1=Extended memory (IV/90 Model 2)
CH: 0=No channel
LAMLOG: 0=No memory log
CR: 0=No card reader
PR: 0=No printer
MAG: 0=9-track, 1=7-track
DEN: 0=Low, 1=high
SOR: 0=No shift override of field type, 1=allow shift override
CCB: 0=Not concurrent Bisync, 1=concurrent.
DOCS: 0=Count records, 1=count documents.
BEEPER: 0=One beep per error, 1=one beep per keystroke after error.
CCODE: 0=EBCDIC, 1=ASCII
LAM2: 1=Yes, 0=no.
DUMP TYPE 0 = No RAM dump.
1 = tape dump.
2 = 8230 disc dump.
3 = 8240 disc dump.
4 = 8260 disc dump.
5 = 8270 disc dump.

*5 CNFIG1 (SYSFG1)

RECPRT: 0=No 12 channel printer for received data (Bisync)
ERP: 0=Extended error recovery
FIND: 0=Do not count find mode statistics
1=Count find mode statistics
REVFRY: 0=Field reverification, 1=not.
CFPROP: 0=VISION print spooling, 1=Bisync direct print.
BATCH: 0=Do not keep detail operators statistics by batch, 1=do.
Tape Buffer Size: Number of sectors for tape buffer (maximum is 6).

*6 LEV78B, PAGSEC, etc. - Set by $SETPP Supervisory Command.

*7 CNFIG4 - Used by VISION.

*8 CNFIG5 - Used by VISION.

LC: 1=Lower case in default 3270 mode.

*9 PRSTAT
Bit 0: Reserved
Bit 1: 0=No log file, 1=log file exists.
Bit 2: 0=No printer, 1=printer.
Bit 4: Reserved.
Bit 5: Clear request.
Bits 6-18: Reserved.
Bit 19: 1=Log C-Type messages.
Bit 20: 1=Log B-Type messages.
Bit 21: 1=Screen print turned on in log.
Bit 22: 1=Print C-Type messages.
Bit 23: 1=Print B-Type messages.

ADDITIONAL FOOTNOTES FOLLOW ON NEXT PAGE

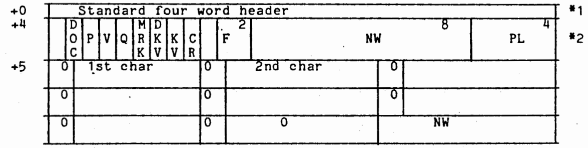
RELEASE BA04
FOOTNOTES FOR CONFIG SECTOR - CONTINUED

*10 8121 Printer Specification Word:
Bit 0: 1=Pica; 0=Elite.
Byte 0: Number of printed lines per page - 1.
Byte 1: Number of increments to advance each line.
Byte 2: Total number of lines per page.

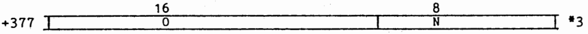
*11 Controller Assignment Table (CAT): Each word corresponds to one controller with the first word corresponding to the 8436 controller, the second word to the 8437 #0 controller, the third to the 8437 #1 controller, and so on. Each word has the following format:

bits	contents
0	0=controller doesn't exist, 1=controller exists
1-7	valid application bits; 0=not valid, 1=valid bit 1 = reserved 2 = 3270 3 = reserved 4 = Bisync 5 = reserved 6 = not used 7 = not used
8-15	current user 0 = none 1 = reserved 2 = 3270 3 = reserved 4 = Bisync 5 = reserved
16-23	reserved

DATA SECTORS



Sector trailer (1 word) when code word is negative:



*1 Special flag in bit 0 of Code word:
0 = sector is full; N = 0400 is implied
1 = sector is not full; word 0377 has N of 004-377

*2 Record header word:
NW: # of words in entry (2-251)
F: # of trailer bytes (1-3). Record size in bytes = (NW-1)*3-F.
P: Program level changed in record
V: Validation override used
Q: Unintelligible key used
KV: Has been key verified
DKV: Has been document-key verified
CR: Has been altered since entry
DOC: This record starts a new document
MRK: This document has been "marked"
PL: Program level 0 to 14

*3 Sector trailer word (only present if bit 0 of code word is 1):
N: Number of words being used in this sector (004-377).
Also is relative address in sector where unused area begins.

Algorithm for # of records in a sector:

Total sector length = 256 x 3 = 768
Sector Header = -4 x 3 = 12
Available for data = 252 x 3 = 756
(words) (chars)

$$NWR = \left[\frac{RS + 3}{3} \right] + 1$$

$$NRS = \left[\frac{252}{NWR} \right]$$

Where, RS = Record Size in characters
NWR = Number of Words per Record
NRS = Number of Records per Sector

$[\ \ ]$ = integer divide (ie, truncate to integer)

NOTE: On jobs with variable length records, the record mix will have to be considered. The system will try to fit each record into the current sector.

RELEASE BA04
OVERLAYS

Overlays are programs cataloged in the DAT000 file of disc drive 0 that load and execute in the overlay area of RAM. They perform infrequently requested functions (TOTAL key, REC-UP, screen print, new mode, etc.) and, because there is only one overlay area for all terminals to share, they are designed to execute rather quickly. An overlay can be called in several ways:

1. From the resident (TOTAL key, etc.).
2. From another overlay.
3. From a transfer program.

OVERLAY REQUESTS QUEUES

There are two types of overlay requests. The first consists of those that are required by terminals (eg, the overlay to process the TOTAL key) and the second consists of those not associated with terminals (eg, Overlay 17, Entry Point 1 which is requested by Level 4 to free page directory blocks.) These two general categories of overlay requests are queued in different ways.

Terminal Overlay Requests

OVRREQ	Number of outstanding terminal overlay requests.
REQ8,X1	Terminal overlay request word.

The format of a terminal overlay request word is:

Q		X		M	Overlay #	entry #
1	3	1	2	1	8	8

Q: 0 = queue keystrokes during overlay execution
1 = toss keystrokes during overlay execution
X: 1 = special request
M: 0 = intercept "MODE" key
1 = pass "MODE" key as data key

Special Overlay Request Queue

:GETOV	Address of top of special overlay request queue. (Next special overlay to be processed.)
:PUTOV	Address of bottom of queue.
:FREOV	Address of free stack for special overlay requests.

Pointer to next queue entry
Ovly # * 256 + Entry Point #

Note: The special overlay request queue always has a dummy entry.

RELEASE BA04
DUMP PROCEDURE

1. AUTO to MANUAL.

2. RESET then STEP.

3. LOAD into TIR - 71100001

4. MANUAL to AUTO then to MANUAL.

   a. Tape Dump
      1. Mount a scratch tape.

      2. MANUAL to AUTO

The system will dump records 1024 words long (06000 bytes).

   b. Disc Dump
      1. Mount a disc which contains a DUMP47 file.

      2. MANUAL to AUTO
         The system will write all of memory out to the DUMP47 file.

5. When the dump is complete, the system will halt with X3=00000000. If low memory is clobbered and the 71100001 will not execute a dump, the address of the dump routine is also in @RAMDM. Find the address of @RAMDM in your load map and load the contents of that location into location 1.

   To get a formatted dump:

```
// DUMPV
/INPUT=filename@drivenumber (use indicated input instead of DUMP47@0)
/TAPE = TAPE7 or TAPE8 or TAPE16 or T7 or T8 or T16 (dump is on tape
                                                     as indicated)
/ANALYSIS (analysis only, omit RAM)
/RAM      (RAM dump only -- no analysis)
/LOW=l.   (start RAM part of dump at location l)
/HIGH=h   (stop RAM part of dump at location h)
```

   The following parameters only apply to IV/90 MOD II dumps:

```
/PHYSICAL (Print memory; memory with no mapping)
/WINDOW=w (Use window w for mapping. Default is window in effect at
          the time of the dump.)
//
```

6. DUMPV requires a 0200 or 0440 (MODII) sector contiguous file called DUMP47. This can be created by running the program MAKD47:

```
// MAKD47
/O= 8230, 8240, 8260, 8270
/BANKS=2  optional (creates 0440 sector file for MODII dumps)
//
```

Debug is a dynamic display of system activity. Normally, the Debug display is kept in a System Block in memory; however, it may be displayed on a screen by entering Mode D.

NOTE: Show Mode G will give a snapshot display of the debug information.

DEBUG DISPLAY FORMAT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	:SS SECOND COUNT	EOR WHEN EOR ACTIVE *	IOx IOB FAILURE COUNT **	-4- WHEN LEVEL 4 ACTIVE *	-I- WHEN INDSET LOOKUP ACTIVE *	xxx NAME OF CURRENT LEVEL 8 SUBMON. ***	-7-, WHEN LEVEL 7 ACTIVE *	OVL WHEN OVERLAY ACTIVE *	xx NAME OF OVERLAY IN RAM (OCTAL)	TRA WHEN TRANSFER PROGRAM ACTIVE *	xxx xxx NAME OF TRANSFER PROGRAM IN RAM		xx NUMBER OF SYSTEM BLOCKS IN PAGDIR	xx NUMBER OF ENTRIES IN FVSRAM	xx NUMBER OF JOBS IN JOBRAM	xx NUMBER OF ENTRIES IN IXRRAM
2	QUE QUEUE LENGTHS →	xx NUMBER OF TERMS IN EOR	xx NUMBER OF SOFT LOST KEYSTROKE	xx NUMBER OF TERMS IN VALUE SET LOOKUP	xx NUMBER OF TERMS IN INDSET LOOKUP	xx NUMBER OF TERMS IN PAGE FETCH	xx NUMBER OF KEYSTROKES QUEUED	xx NUMBER OF OVERLAY REQUESTS QUEUED	xx NUMBER OF DISC REQUESTS QUEUED	LKD WHEN DISC I/O LOCKED	xxx CURRENT OR LAST DISC USER +	Ex COUNT OF DISC ERRORS **	xx DISC OPERATION AT LAST ERROR ++	xx xxx xx DISC STATUS AT LAST I/O ERROR		
3	TOO TERMINAL NUMBERS →	nn T= LAST EOR	nn T= LAST LOST KEYSTROKE	nn T= LAST VALUE SET LOOKUP	nn T= LAST INDEX SET LOOKUP	nn T= LAST PAGE FETCH	nn T= LAST KEYSTROKE PROCESSED	nn T= LAST OVERLAY LOAD	xx CURRENT OR LAST DISC OPERATION ++	D SSS SSS LOGICAL DRIVE AND SECTOR OF LAST DISC I/O			-T- RESERVED FOR TAPE	xxx xxx xxx RESERVED FOR TAPE ERROR STATUS		
4	xx xxx RP AT LAST LEVEL 4 INTERRUPT		xx NUMBER OF HARD LOST KEYSTROKES	xxx NAME OF LAST VALUE SET USED	xxx NAME OF LAST INDEX SET USED	xxx NAME OF LAST FORMAT PAGED	x LAST KEYSTROKE PROCESSED	RESERVED					RESERVED FOR COMMUNICATIONS SUBSYSTEMS			

NOTES

* BLANK WHEN INACTIVE.

** BLANK WHEN ZERO.

*** OVL – OVERLAY SUBMONITOR
TRA – TRANSFER PROGRAM SUBMONITOR
PRT – PRINT SUBMONITOR
ALO – ALLOCATION/DEALLOCATION SUBMONITOR
DYN – DYNAMO SUBMONITOR
COM – COMMUNICATIONS SUBMONITOR
BLANK – NO SUBMONITOR ACTIVE

\+ TERMINAL NUMBER IF LEVEL 7
SUBMONITOR NAME IF LEVEL 8

++ RD – STANDARD READ
WR – STANDARD WRITE
NR – NONSTANDARD READ
NW – NONSTANDARD WRITE
OV – OVERLAY LOAD

RELEASE BA04
DYNAMO

DYNAMO is a dynamic memory display, patch, and trace program that may be used at the DEBUG screen.

To Start DYNAMO

After entering Mode D, enter the password FOURFAZE. The DYNAMO display will appear on the bottom of the screen above the message line. The remainder of the screen will depend on the screen size:

6 lines: The DYNAMO display will appear by itself.
12 lines: An abbreviated form of the instructions for using DYNAMO will appear above the DYNAMO line.
24 lines: The abbreviated form of the instructions will appear above the DYNAMO line and the DEBUG display will remain on the screen.

The DYNAMO line has the following format:

```
XXXccclllll aaaaaaa bbbbbbb ccccccc ddddddd
```

where:

XXX is CHG - change mode, enter a location into the lllll field.
DIS - octal display mode, the contents of lllll are displayed at aaaaaaa, the contents of lllll+1 are displayed at bbbbbbb, etc.
ASC - ASCII display mode, aaaaaaa etc is ASCII representation of location lllll. 36 bytes are displayed.
PAT - patch has been made beginning at location lllll.
TRC - a trace or stop has been planted at location lllll.

ccc is count of the number of times the instruction at location lllll has been executed in trace or stop mode.

lllll is the current location in RAM being displayed, patched or traced.

The contents of the line immediately below the DYNAMO line depends on function being performed.

To Display RAM:

Press the HOME key.
Key the 5 octal digit RAM location.
Press the INDEX (F5) key.

To display in ASCII.
PRESS ASD (F1) key.
36 bytes (12 words) beginning at location lllll are displayed.

To Display MOD II Information

Press the HOME key.
Key the address WWWPP; where WWW = window (8 bits)
PP = page (0-31)
Press the PROG CTL (F2) key.

Display appears as:

1. Mapper RAM
2. Window Register
3. Memory Parity Register
4. Mapper Parity Register

To STOP Execution At A Specified Location:

The stop occurs before the instruction is executed; therefore, you can check the condition codes before a branch, etc.

Press HOME key.
Key the 5 octal digit location.
Press PROG6 (F11) key.

When the stop occurs, the status line will show the contents of registers RA, RB, X1, X2 and X3 in octal from left to right. At the far left end of the status line the 3 byte display Scc will display. cc is the octal condition code at the time of the stop. The count ccc on the message line should show 001 the first time.

To cancel the stop and allow the system to continue press RESET (TAB).

To release the stop and catch it again the next time press PROG6.

NOTE: This may be used to stop trace at all levels above level 3.

To TRACE Execution At A Specified Location:

The trace is a dynamic one which may be stopped at any time by pressing the PROG6 or HOME key.

Press HOME key.
Key the 5 octal digit location.
Press PROG5 (F10) key.

The display for the trace is the same as that for the stop.
To stop the trace press PROG6 or HOME.

To release the stop and catch it again the next time press PROG5.

NOTE: This may be used to stop trace at all levels above level 3.

To PATCH RAM (4 or fewer consecutive words)

Press HOME key.
Key the 5 octal digit lowest location to be patched.
Press the INDEX (F5) key.

The four locations beginning at lllll will display.
Verify that you are at the correct spot in RAM.

Press HOME key.

Press the RIGHT arrow to position the solid cursor to the octal digits to be changed and key the correct octal digits. The actual changes are not made to RAM until the next step is performed.

Press PROG1(F6), PROG2(F7), etc. to patch 1, 2, etc. words.

PAT will display at the left most end of the message line.

RELEASE BA04
SINGLE WORDS IN RAM

These are counters, pointers, addresses, etc., used by VISION to keep track of itself. A description of each word follows:

SYMBOL	DESCRIPTION
ACTO@	Address of the pointer to the Debug Display Area. This is present in a system block on production systems or on a screen on Debug systems.
ASCHR	Current system hour in ASCII, in the form bHH.
ASCMIN	Current system minutes in ASCII, in the form :MM.
ASCSEC	Current system seconds in ASCII, in the form :SS.
ASS4	Address of the last 4-word block assigned.
ASS8	Address of the last 8-word block assigned.
ASS16	Address of the last 16-word block assigned.
ASS32	Address of the last 32-word block assigned.
ASS64	Address of the last 64-word block assigned.
ASS128	Address of the last 128-word block assigned.
BATACT	Mask to inhibit the use of any active batch (see state request bits in the batch directory disc format).
BFRLST	Pointer to the start of the free sector/buffer list.
BLKCNT	Magnetic tape block count.
CHAR	Last PROKS character.
CHARV	The most recently typed in character in verify mode.
CLOCK	Fires level 4 every 1/5 of a second. Set to -12 at level 4 and incremented by level 0 every 1/60 second.
CLOCKS	One second clock.
CNFIG	Disc address of CONFIG sector.
COMMWD	3270 CONFIG word.
COMWT8	Value of COMWT8 performance parameter.
D	Zero = Debug display in system block; non-zero = screen + 1 of Mode-D screen
DAY	Day number, ASCII
$DBASE	Starting sector address for VISION.
DISCQ1	Pointer to FWA of primary disc request queue.
DISCQ2	Pointer to FWA of secondary disc request queue.
DSETIM	Time of last disc error (T - 86,400).
ECTAB	Counter for all disc errors
EORREQ	Number of end-of-record requests outstanding.
EOVDAT	First data sector of $EOV tape label data.
F90M2?	0 = Not a MOD II, 1 = MOD II.
FETREQ	Number of page fetch requests outstanding.
@FV	New location of Format Vector Table.
FVLEN	Current length of the Format Vector Table.
FVSDIR	Disc address of FVS directory.
FVSRAM	Pointer to the Format/Value set directory in RAM.
HDRDAT	First data sector of $HDR tape label data.
HOUR	Binary system hour.
IO6CNT	IO6 chip failure count.
ISVER3	0 = no index sets, -1 = index sets
IXRRAM	Pointer to the Index Set directory in RAM.
JOBDIR	Disc address of job directory.
JOBRAM	Pointer to the job directory in RAM.
KBCINP	The last processed keystroke before translation.
KBLOG	The backlog count of keystrokes.
KFREE	Start of keystroke free cell list.
KFSIZE	Total number of keystroke cells in the system.
KFXTND	-(Number of words to extend KFREE); initially set to -16*NTERMS
LASTZN	Highest zone in the system.
LDTAB@	Pointer to logical device table.
LL	Screen line length in words - 020 or 040.
LL2	LL + LL.
LOCKPR	Printer lockout from magnetic tape.
LOGBAT	The sector address of the log file batch.
LOGJOB	Jobname/batch number of the log file.
....+1	Jobname/batch number of the log file.
....+2	Jobname/batch number of the log file.
....+3	Jobname/batch number of the log file.
....+4	Jobname/batch number of the log file.
LOGSEC	Current log sector.
MAGTPE	Count of magnetic tape errors.
MAXRAM	Maximum RAM available; determined by initialization.

SINGLE WORDS IN RAM - CONTINUED

MFE?	Zero=non-MFE system; non-zero=MFE system
MINUTE	Current time in minutes.
MP3270	Mapword for selected 3270 screen. 0 = none.
M#OVLY	Highest numbered overlay
N	Count of the number of screens.
NDS	Number of disc drives.
NEX4	Address of the next 4-word block available.
NEX8	Address of the next 8-word block available.
NEX16	Address of the next 16-word block available.
NEX32	Address of the next 32-word block available.
NEX64	Address of the next 64-word block available.
NEX128	Address of the next 128-word block available.
NEX256	Address of the next 256-word block available.
NLGSEC	Count of the sectors in the log file.
NPRSEC	Count of the sectors in the print queue.
NSB	Number of screens plus number of buffered printers.
OHTOP	Pointer to BAM OCB header chain.
OIDDIR	Disc address of OID directory.
OSOPBD	Operator statistics batch directory sector.
OSOPB1	Operator statistics first data sector.
OVRREQ	Number of terminal overlay requests outstanding.
P2L2P@	Pointer to P2L2P Table.
PAGDIR	Pointer to the page directory in RAM.
PASSWJ	Job define password.
PASSWP	Print/log password.
PASSWS	Supervisor password.
PFCB@	Location of the pointer to the submonitor control block.
PRTACT	Print active/idle flag.
PTHING	Pointer to the system constant (entered during system bring-up).
PUSER@	Address of PUSER.
SEC95P	95% full point in sectors available.
SECAVL	Total sectors available on all drives.
SECNBR	Number of current PROKS sector.
SECPTR	Pointer to the current PROKS sector buffer.
SECEND	Pointer to the end +1 of the current PROKS sector buffer.
SEMOH	BAM flag.
SRA13	Save RA and X1 for level 3.
....+1	Save RA and X1 for level 3.
SYSF84	Pointer to CAT.
SYSFG3	not used.
SYSFG2	not used.
SYSFG1	Additional configuration information (CONFIG sector word 0117).
SYSFIG	Configuration word for the system (CONFIG sector word 0116).
SYSNBR	Current system sector in RAM.
SYSPTR	Pointer to the system sector buffer in RAM.
SYSEND	Pointer to the end +1 of the system sector buffer.
TENSEC	Ten second clock.
THING	The system constant area.
....+1	The system constant area.
....+2	The system constant area.
....+3	The system constant area.
TIME	Time = seconds since midnight - 86400.
TIMELW	The master dir. sec. last written to disc.
TP256	Address of next TP area 256-word block available.
USER	Last PROKS user table address.
USER8C	Current level 8 user table address
VOLCNT	Tape volume reel count.
WHAT?!	Interrupts unexpected.
ZONTOT	Kept in word 0200 of the master directory sector. Byte 0: not used; Byte 1 = Total zones -1; Byte 2 = Largest zone number.
@$MAXL	Pointer to the size of the LDTAB device.
@OVDIR	Address of overlay directory.
@ZONPT	Address of zone pointer table.
$LDINV	Required by IDOS, E3, and above.

DIABCF	Address of start of 8121 complete-flags.
DIABND	Number of 8121 printers.
DIABP2	Address of basic 8121 driver, PRIN22.
DIABSU	Unit number of the 8121 sysprint. If the system printer is not an 8121, then DIABSU is -1.
PRSTAT	Printer status word.
LPOUT	Printer selection value.
CQG	Pointer to first entry in submonitor control block.
CQP	Pointer to last entry in control block.
CQT	Address of current submonitor control block.

CHAN2	Zero means channel 2 is idle.
CHAN2R	Non-zero means Level 8 is requesting channel 2.
DISCIO	Non-zero means a disc operation is going.
DISCLK	Non-zero means channel 2 locked-out for disc, in use by tape.
DISCRQ	Count of disc requests pending.
REQTAB	$JDISC request table.
...+1	$JDISC request table.
...+2	$JDISC request table.
...+3	$JDISC request table.
....+4	$JDISC request table.
....+5	$JDISC request table.

?IGP00	Used to count devices serviced during general poll loop. Initially set to -32.
?IGP01	Last device to use status or TX subroutines starting device number for general poll.
?IGP05	Indicates a device has requested transmission or has changed status.
?IPG06	Second/10 to delay the transmission or EOT during general poll if no devices have changed status or requested transmission.
?NAKS	Count of NAKs sent.
?BFCNT	Buffers pending between level 1 and 7.
?DSRER	Number of Data Set Ready errors.
?OLCNT	?BFCNT served for retransmission.
?REPLY	Text mode response from Host.
?4EROR	Error flag set by level 4.
?ACKOT	Next ACK to be output.
?ACKIN	Expected ACK in.
?ICPSL	Control mode flag from level 1 to level 7.
?7SYNC	SYNC received in level 7.
?7TASK	Current CU3270 task.

PROGRAMMER WORKSTATION - BD03

Installation of PWS is described in the SRN, Section 13 - pages 43-55.

CONFIGURING COMMUNICATIONS

Communications configuration files contain both protocol and information associated with a host. There should be one communications configuration file for each host. If PWS will communicate with only one host, and the host queue id was chosen to be blanks, then use file SYS.PWSFIG.DATA80@0 for the communications configuration file. Otherwise, create one file for each host/transmission queue. Name the files, SYS.PWSxxx.DATA80@0, where xxx is the host/transmission queue id. Use SYS.PWSFIG.DATA80@0 as a model.

Signon and Signoff Cards.

Put a $SIGNON ($SIGNOFF) record in the configuration file followed immediately by a record containing the signon (signoff) card. BD03 has the ability to specify an automatic signon in the configuration file. To send the signon card immediately after the line has been started, include an $IMMEDIATE SIGNON record in the configuration file. The command "SIGNON Ln" will also cause the signon record to be transmitted.

Line Configuration Parameters.

The following parameters indicate what characteristics the line should have. The parameters and values are listed with a brief explanation of what impact each parameter has. All parameters can be truncated to 2 characters after the dollar sign, except $SIGNON and $SIGNOFF.

Model 20 Line Configuration Parameters.

$MODEL 20

First non-comment card in the configuration file when specifying a Model 20 line.

$*

Indicates a comment record. The record is logged but has no effect on the configuration.

$AUTO ANSWER (default = $MANUAL)

Specifies the modem as auto-answer.

$BLOCK SIZE = value (default = 400)

Specifies the maximum block size between 150 and 512, inclusive.

$COMPRESSION = value (default = 3)

Specifies compression type. Values and meanings are:

0 - No compression.
1 - Trailing blank compression.
2 - Full blank compression.
3 - Full character compression.

$CPU (default = $CPU)

Specifies that the other station is a mainframe.

$EBCDIC (default = $EBCDIC)

Documention only as ASCII lines are not supported.

$IMMEDIATE SIGNON (default = no immediate signon)

Specifies to send a signon when the line is started.

$LOG = value (Line trace must be optioned to use this command.)
Specifies the type of data to log in the line trace.

1 - Log line data.
2 - Log controller request/response tables.
3 - Log line data and controller tables.

$MANUAL ANSWER (default = $MANUAL)
Specifies that the modem is manual answer.

$MESSAGE SIZE = value (default = 120)
Size of received console messages. Must be less than 133.

$POINT TO POINT (Default = $POINT TO POINT)
Documentation only. Model 20 does not support multipoint.

$PRIMARY (default = $SECONDARY)
Indicates that this is the primary station.

$SECONDARY (default = $SECONDARY)
Indicates that this is the secondary station.

$SIGNON
The signon card is in the following record.

$SIGNOFF
The signoff card is in the following record.

$SPANNED (default = no spanned records)
Records can be spanned between transmit buffers.

$TERMINAL TO TERMINAL (default = $CPU)
The other station is a terminal.

$TIMEOUT = value (default = 20)
The time in seconds before the line is considered idle.
Acceptable values are between 5 and 25 seconds, inclusive.

2780 / 3780 Line Configuration Parameters

$2780
First non-comment card in the configuration file when specifying a 2780 line.

$3780
First non-comment card in the configuration file when specifying a 3780 line.

$*
Indicates a comment record. The record is logged but has no effect on the configuration.

$ADDRESS = hh hh hh hh hh
Specify the terminal address. This provides switched network protocol capability on dial-up lines. This is also used to provide the terminal address on a multipoint line. The "hh" is any 2 hexidecimal digits (0-F).

$ASCII (default = $EBCDIC)
Specifies an ASCII line (not supported on multipoint lines).

$AUTO ANSWER (default = $MANUAL)
Specifies the modem as auto-answer.

$BLOCK SIZE = value (defaults = 400 for 2780, 512 for 3780)
Specifies max block size, value must be between 128 and 512.

$COMPRESSION = value (defaults = 1 for 2780, 2 for 3780)
Specifies compression type. Values and meanings are:
1 - Trailing blank compression (3780), no compression (2780).
2 - IBM 3780 compatible blank compression.
3 - Four-Phase compatible full character compression.

$CPU (default = $CPU)
Specifies that the other station is a mainframe.

$EBCDIC (default = $EBCDIC)
Specifies an EBCDIC line.

$EM INSERTION ($2780 only, default = no EM insertion).
Specifies Automatic EM insertion.

$EXTENDED LINE BID RETRY (default = retry line bid 40 times)
Specifies to bid for the line indefinately.

$IMMEDIATE SIGNON (default = no immediate signon)
Specifies to send a signon when the line is started.

$LEASED (default = switched line)
Specifies that PWS is on a private line.

$LOG = value (Line trace must be optioned to use this command).
Specifies the type of data to log in the line trace. The system must be optioned for line trace capability and there must be pages of RAM available for a log area. The values are as follows:

1 - Log line data.
2 - Log controller request/response tables.
3 - Log line data and controller tables.
(It is suggested that 3 be used. default = 0)

$MANUAL ANSWER (default = $MANUAL)
Specifies that the modem is manual answer.

$MULTIPOINT (default = $POINT TO POINT)
Specifies that PWS is on a multipoint line.

$POINT TO POINT (default = $MULTIPOINT)
Specifies that PWS is on a point-to-point line.

$PRIMARY (default = $PRIMARY)
Indicates that this is the primary station.

$SECONDARY (default = $PRIMARY)
Indicates that this is the secondary station.

$SIGNON
The signon card is in the following record.

$SIGNOFF
The signoff card is in the following record.

$SWITCHED NETWORK PROTOCOL (default = not switched network)
Specifies that the mainframe expects switched network protocol. The mainframe always bids to establish the line, regardless of which station is to send first. If this parameter is used with $ADDRESS, the terminal will put the specified address out with the first ACK.

$TERMINAL TO TERMINAL (default = $CPU)
The other station is a terminal.

$TIMEOUT = value (default = 20)
Time in seconds before the line is considered idle. Acceptable values are between 0 and 25 seconds, inclusive. Zero indicates the line is never idle.

$TRANSPARENT (default = no transparency)
Indicates that PWS is to transmit data in transparency.

CREATE OR MODIFY OPERATOR IDS FILE

Use PWS to create or modify file SYS.PWSIDS.DATA80@0, the list of valid operator ID's. User Logon ID OPR is hard coded in the system as a bootstrap. All items except for the User Logon ID itself are optional. The items are positional. Indicate omitted items with two consecutive commas. The format of a user record and meaning of each item are shown:

```
XXXXX     ,AAA,BBB,CCC,DDD,EEE,FFF,GGG,HHH,III,JJJ,KKK,LLL,MMM
```

XXXXX	User Logon ID - Upper case, 3 to 5 characters, leading alpha. Starts in column 1, followed by blanks.
AAA	User Logon Password - Upper case, 1 to 8 characters in length. If omitted, no password will be required for logon of this user.
BBB	System Operator Privileges - Specify 'OPR' if the user is allowed to issue reserved system commands such as START or HOLD, or 'NOOPR' if not. The default is 'NOOPR'.
CCC	Default Filetype - All upper case, 1 to 8 characters in length. Must be one of the valid filetypes defined by the system. If none is specified, the system default is used.
DDD	Default Drive - A single digit, corresponding to a logical drive number. If none is specified, the system default is used.
EEE	Default LINENUM Setting - Specify 'NUM' to set LINENUM on for this user, or 'NONUM' to set LINENUM off. The default is 'NONUM'.
FFF	Default PAGESIZE Setting - Specify the scrolling PAGESIZE to be used whenever a scroll page request is issued. Values may be 'FULL', meaning a full screen, 'HALF', meaning a half screen, or any number between 1 and 99. If none is specified, the system wide default is used.
GGG	Default Communications Destination - Reserved.
HHH	Communications Console Authority - Reserved.
III	Transmit/Print Queueing Priority - Reserved for future expansion.
JJJ	Alarm Beeper Setting - Specify 'BEEP' to enable terminal alarm, or 'NOBEEP' to disable it. The default is 'BEEP'. If system does not have the terminal alarm hardware, this item is ignored.
KKK	Input Erasure Setting - Specify 'ERASE' if the command input area is to be cleared after command execution. Specify 'NOERASE' if the command input area is not to be erased.The default is 'ERASE'.

DEFINING PWS FILE TYPES

PWS files types are defined in module SBDFTP. To add or change a definition SNEDIT the change to the table of valid file types (FILTYP.TABLE). Each entry is four words long. The first three words contain an ASCII string left justified and blank-filled with maximum length of 8. The last word contains a pointer to the set of values which is associated with the type.

The last entry in the table has binary zeroes in place of the string to indicate that it is the last entry. The pointer in the last entry points to the default values returned if the specified file type cannot be found.

```
FILTYP.TABLE  EQU $
       DCA    .MODEL   .      ** MODEL ENTRY **
        PZE    MODEL.VALUES
       DCN    0               ** LAST ENTRY **
       DCN    0
       DCN    0
       PZE    DEFAULT.VALUES
```

A model file type is shown below.

```
*********************************************************************
*
*      FILE TYPE MODEL FILE CHARACTERISTICS
*
*********************************************************************

MODEL.VALUES  EQU $
       DCN    1B0+1B1+1B5+1B13+1B21   Default TABS - one bit position
       DCN    1B6+1B14                defines one tab position for up to
       DCN    1B14+1B17               148 character positions.
       DCN    0
       DCN    0
       DCN    0
       DCN    0
       DCN    1                   Left text margin       (field entry #0
       DCN    7                   First FIELD definition              #1
       DCN    24                  Second FIELD definition             #2
       DCN    80                  Right margin                        #3
       DCN    0      (Up to 7 fields may be specified)                #4
       DCN    0                                                       #5
       DCN    0                                                       #6
       DCN    0                                                       #7
       DCN    3                   Last field entry
       DCA    .000000100.         Default sequence increment (1- )
       DCN    -1                  Upper CASE (0=Both)
       DCN    0                   Left sequence length (0- )
       DCN    8                   Right sequence length (0- )
       DCN    72                  Text length (0 to Record-Sequence lengths)
       DCN    80                  Record length (0-148)
```

DIRECTORY LIST UTILITY

A COBOL utility program is provided for listing of the directory data maintained on PWS format files.

GENERAL:
This program searchs the directories of all assigned logical drives looking for PWS format files. When a files are found, the directory data is extracted, this data is then formatted and a list is output. The source for this program is in file SBDDLS.

EXECUTION:
This program requires MFE for execution.

1) To start execution, issue the command START PWSDIR from the MFE Operator's Console. PWSDIR is a control file which specifies execution of the directory extract program, ABDDXT; a sort; and the directory list utility program, ABDDLS.

   To run the list program alone, issue the command START DIRLIST from the MFE Operator's Console.

   If the directory list program is to accept its parameters from a terminal, the control file should include the following JCL:

```
// ABDDLS
/EXTRACT=filename.
//
```

   If the directory list program is to accept its parameters from the control file and execute without operator intervention, the control file should include the following JCL:

```
// ABDDLS
/RUN=AUTO.
**   AUTO RUN JCL FOLLOWS
CTL: C=aaaaaaaa,U=bbbbbb,N=cccccc,T=dddddddd,D=ee,M=fffffff
/EXTRACT=filename.
//
```

   Field locations and lengths are fixed. The CTL: is a required identifier. The C, U, N, T, D, and A parameters specify criteria, file user, file member name, file type, drive, and date of last access. These are described in more detail below.

2) If the auto run option was not selected, the program will initialize the requestors screen to show:

```
                  PROGRAMMER WORKSTATION

                  DIRECTORY LIST UTILITY

                  EXTRACT PHASE


    TAB OR ENTER REPORT CRITERIA:   STANDARD
      (STANDARD OR SPECIAL)

         USER ID:
         FILE NAME:
         FILE TYPE:
         LOGICAL DRIVE:
         DATE LAST ACCESS:
```

3) To specify a selection of files based upon the criteria elements listed, input 'SPECIAL' into the first field, then input the values desired for the rest. All fields must be entered. To indicate that a field is not to be treated as part of the matching criteria, enter a TAB.

   All fields input must match in the directory data or a failure to match will be assumed.

4) The defined files within this program are as follows:

   /A = Configuration file (used to extract customer name)
   /E = Extract file (pre-digested input to select and list)
   /R = Report file

LOG REPORT UTILITY

A COBOL utility program is provided for listing of the PWS log data maintained on disc.

GENERAL:
Using the log files created by a PWS session or a series of files listed in control file. The data from these files is output in a formatted list. A summary of major event types is provided at the completion of processing. The source for this program is in file SBDRPT.

EXECUTION:
This program requires MFE for execution.

1) To start execution, issue the command START PWSREP from the MFE Operator's Console. PWSREP is a control file which specifies execution of the log report utility program, ABDRPT.

If the log report utility program is to accept its parameters from a terminal, the control file should include the following JCL:

```
// ABDRPT
//
```

If the log report utility program is to accept its parameters from the control file and execute without requiring operator intervention, the control file should include the following JCL:

```
// ABDRPT
/RUN=AUTO.
** AUTO RUN JCL FOLLOWS
CTL: I=xxxxxxxx,F=yyyyyy,P=zzzzzz
//
```

Field locations and lengths are fixed. The CTL: is a required identifier. The I, F, and P parameters specify input type, file list, and print option. These are described in more detail below.

2) If the auto run option was not selected, the program will intialize the requestors screen to show:

```
                    PROGRAMMER WORKSTATION

                     LOG ANALYSIS UTILITY

                     EXTRACT PHASE


    FILE INPUTS VIA DISC/KEYBOARD:  DISCFILE

    ENTER NAME OF CONTROL FILE:     LOGLST

    ENTER PRINT/NOPRNT OPTION:      PRINT
```

3) If a single file is to be processed, then the entry of any value other than DISCFILE will indicate this to the system, and the second display line will change to request the name of the file to be processed.

The PRINT/NOPRNT option allows for suppression of the output listing.

4) The defined files within this program are as follows:

/A = Configuration file - (used to extract customer name)
/B = Log file
/C = Output work file
/D = Input file list file
/R = Report file

PWS LOG MESSAGE ORGANIZATION

Each message in the system console event log has a unique three digit code number. The code numbers are assigned in categories, as follows:

PREFIX	RANGE	DESCRIPTION
1		Accounting Data
	120-	Terminal Logon/Logoff
	140-	Print Subsystem
	160-	Communications Subsystem
2		Error Tracking
	200-	Hardware Errors, Except Communications
	220-	Communications Hardware Errors
	240-	System Software Errors
	260-	Communications Software Errors
3		Operational Messages
	300-	Operator Error
4		PWS/Host Commands and Responses
	400-	PWS Commands
	450-	Host Commands
	460-	Host Responses
5		Security
	500-	Terminal Access Security
	520-	File Access Security
6		Performance
	600-	System Limitation Exceeded
	640-	Performance Monitoring

COMMUNICATIONS LINE TRACE

The line trace capability must be optioned into the system before it can be used. The line trace is initiated by including a $LOG record in the configuration file. When the line is started the trace area is allocated. The trace area is made up of free pages from MFE; it cannot be started if MFE has no pages available. The trace area is returned to MFE, when the user enters a STOP COMMLOG command. The commands that effect the trace while it is running are as follows:

> STOP COMMLOG causes the line trace to stop and all memory allocated for the trace area is returned to MFE.
>
> SUSPEND COMMLOG causes the line trace to stop. The memory is still allocated to the line trace area.
>
> RESUME COMMLOG causes the line trace to start running again after a SUSPEND command.

The line trace is useful for diagnosing communications problems but it should not be run when no problems exist as it impacts system performance. The proceedure for getting a line trace is as follows:

1) Put a $LOG record in the communications configuration file.
2) When the event occurs that you wish to trace, execute a SUSPEND COMMLOG command from the PF6 screen. To write the trace area to disc, enter a DUMP command on the MFE Operator's Console.
3) Once the dump to disc has completed, a STOP COMMLOG command may be executed to return the trace area to MFE.
4) When MFE has been shut down, execute the TRACE utility. See the Communications Services and Utilities manual for details on how to run trace.

CALCULATING PWS MEMORY REQUIREMENTS

BD03 requests memory from MFE as necessary. The memory used by PWS can be calculated by adding up the items listed below. Fractional page results should be rounded up to the next whole page.

- o Resident base is 33 pages. Add 1 page per line printer and 1 page for a card reader.
- o 45 words are used for each printer buffer.
- o Communications requires 4 pages when active. When comm is down, the memory returns to MFE. If both comm protocols are optioned, 2 additional pages are required for a total of 6 pages when comm is active.
- o Line trace logging requires 2 pages when it is active.
- o One fifth of a page is required for each potential terminal. As an example, if the PWS terminal logon limit is 20 terminals, 4 pages of memory are added to the size of the resident base.
- o A pool of pages is preallocated by PWS during initialization. The number of preallocated pages is T/4+2, where T is the terminal logon limit.
- o Memory is dynamically allocated and deallocated by each terminal as functions are performed. Requests are satisfied from the preallocated pool of pages until it is exhausted, then additional pages are requested from MFE. The amount of memory actually used by each terminal (not including waste) will fall somewhere in the following range:

  - completely idle: 0 pages
  - not in edit or view state, optimistic average: 1/8 page
  - not in edit or view state, pessimistic average: 1/4 page
  - view state: 5/8 page
  - edit state, optimistic average: 3/4 page
  - edit state, pessimistic average: 7/8 page
  - max (copy involving an IDOS format file): 1-1/4 pages

FORMATTED DUMP PRINTING UTILITY

The PWS formatted dump program is an IDOS utility which prints an interpretive dump of PWS using the save file generated by LOADER during the PWS generation and the DUMP47 file written by MFE after a system crash.

The formatted dump program is called ABDFDM. It has several options, which are described following:

/A.	Print only the analysis portion of the dump (omit the ram dump).
/R.	Print only the ram portion of the dump (omit the analysis).
/M=	Print the load map. If this option is absent then a full load map will be printed, otherwise ... /M=NAME will print the load map sorted by symbol name. /M=VALUE will print the load map sorted by symbol value. /M without anything following will omit the load map.
/L=addr	The low memory address or starting symbol at which to start printing memory.
/H=addr	The high memory address or ending symbol at which to finish printing memory.
Note:	A symbol is any symbol that appears in the load map and is not in any of the hyperspace segments.
/S.	Dump the hyperspace segments in memory.
/T=	The screen number to dump or 'ALL' to dump all screens signed on to PWS. If a screen is not signed on to PWS it will not be printed even if it was explicitly specified by this option.
/N=	The name of the PWS job running under MFE to be dumped.
/W=	The window of the PWS job running under MFE to be dumped.
/I=file	The name of the MFE memory dump file (DUMP47).
/B=file	The name of the relocatable save file generated by LOADER.

The defaults are:

Print the load map sorted both ways.
Print the analysis.
Print the memory dump.
Low memory address to print is the symbol PWSBOT (/L=PWSBOT).
High memory address to print is 077777 (/H=077777).
Do not print the hyperspaces.
Do not print any screens.
Dump using the last active window.
The input dump file name is DUMP47.
The relocatable save file name is RBDBAS.

Recommended JCL to print the most useful information is as follows:

```
// ABDFDM
/S,T=ALL.
/W=xx.     Where xx = the current window during the crash.
//
```

To take a manual dump:

A. Place the CPU in MANUAL mode.
B. Enter 071100001 (BRM* 01) into the console keys.
C. Press the RESET, STEP, and LOAD switces in this order.
D. Clear HALT (MANUAL mode to AUTO mode).
E. Clear HALT again when machine halts.

The dump has been written to disc (NOTE: If X3 is not = zero the dump may have failed.)

PRINTING AN MFE DUMP:

The processor MFEDMP prints a formatted dump of the MFE and/or the application programs.

MFEDMP is executed as follows.

1. Sign on to MFE as the SYSTEM CONSOLE OPERATOR.
2. Enter START MFEDMP or // MFEDMP into MFE SYSTEM CONSOLE SYSIN.
(you may wish to create a control file that has the MFEDMP option JCL that can be called at run time)

3. The MFEDMP options are:

// MFEDMP	
/RAM OR /ANALYSIS.	If specified this will cause either the formatted section or the octal ram dump to print. If omitted both sections will print.
/INPUT = FILENAME @ DRIVE.	If omitted defaults to "DUMP47 @ 0"
/WINDOW = WWW.	Optional. Defaults to run time window.
/PHYSICAL.	Optional. Print ram in physical page order not logical by window order.
/LOW = LL. (PAGE NO.)	Optional. Low boundry.
/HIGH = HH. (PAGE NO.)	Optional. High boundry.
/NAME = JOBNAME.	Optional. Sets window to "JOBNAME" primary window.
//	END OF JCL

PWS MEMORY LOAD LAYOUT

LOC	DESCRIPTION
0	MFE - see MFE documentation
04000	Screen x and associated TCB (04000 - 04137, 05540 - 5777 TCB)
06000	Segment memory (segments not concurrently resident)
	HYSEG1 Command handling HYSEG2 File management HYSEG3 Misc. + configuration HYSEG4 Communications
024000	SYS - (SBDSYS) Background TCB's DIRECB's and PCB's Monitor(PF4) and Operator(PF6) screens
	CONSTANTS (SBDCON) VARIABLES (SBDVAR)
	SUPERVISOR - (SBDSUP) Tasking
	RESIDENT - non hyperspace modules
@PWSEN	(RESIDENT end rounded up to next page boundary)
	Memory pool - buffers, etc. (Up to 3 pages for each task) FILECB's
PWSTOP	LOGICAL ADDRESS SPACE REMAINING (MAY BE ZERO)
077777	

Note: Starting addresses after SYS vary according to configuration.

TASK CONTROL BLOCK FORMAT DESCRIPTION

WORD	SYMBOL	MEANING
0	TK.LINK	Next free task block
01	TK.PREVIOUS	Previous task block
02	TK.RA	Saved (or returned) RA
03	TK.X1	Saved X1
04	TK.X2	Saved X2
05	TK.X3	Saved X3
06	TK.RB	Saved RB
07	TK.RP	Saved RP (restart address)
010	TK.EVENT1	Wait/post event
011	TK.EVENT2	Wait/post event
012	TK.TIMER1	Decremented every 1/10 sec
013	TK.TIMER2	Decremented every 1/10 sec
014		RESERVED
015	TK.PAGES	Memory mngr assigned pages
020		RESERVED
021	TK.WINDOW	Window number
022	TK.HYPRWIN	Hyperspace windows
026	TK.MOTHERCB	Addr of attaching Task CB - 0 if not a background task
027	TK.COMMCB	Addr of Communications CB - 0 if no communications
030	TK.TERMINAL	Physical terminal number (see WORD EXPANSION)
031	TK.OPTION	Current option selected (see EXPANSION)
032	TK.FLAGS	Flag bits (see EXPANSION)
033	TK.LOCATION	Location (queue) descriptor
034	TK.PAGESIZE	Scroll page size (see EXPANSION)
035	TK.HORIZOFF	Horizontal scroll offset from left
036	TK.ADJUST	Add to record col to get screen col
037	TK.USERID	ID of user who is logged on - ASCII
042	TK.DEFLTTYPE	Default file type - ASCII
045	TK.DEFLTDRIVE	Default file drive
046	TK.CONSAUTH	Console command authorization level
047	TK.TABS	Tab stop bits relative to text
056	TK.FIELDS	Fields table relative to text
066	TK.LASTFIELD	Index to last entry in fields table
067	TK.INCREMENT	Default sequence increment for file - decimal ASCII
072	TK.CASE	Default case switch for file - NEG=UPPER ONLY, POS=BOTH
073	TK.TEXTBUFF	Working store text buffer
0150	TK.SEQBUFF	Working store sequence number buffer
0153	TK.FLAGBUFF	Working store flags buffer
0156	TK.ACCOUNT	Account ID - ASCII
0161	TK.USER	User ID - ASCII
0164	TK.NAME	File name - ASCII
0167	TK.TYPE	File type - ASCII
0172	TK.DRIVE	Logical drive
0173		RESERVED
0174	TK.WORKDIRECB	Address of directory entry CB for work file
0175	TK.WORKFILECB	Address of file CB for work file if open
0176	TK.ORIGDIRECB	Address of directory entry CB for orig file
	TK.IDSDIRECB	Address of ID'S file directory entry CB
0177	TK.ORIGFILECB	Address of file CB for orig file if open
	TK.IDSFILECB	Address of ID'S file file CB

```
WORD  SYMBOL         MEANING

0200 | TK.INPTDIRECB  Address of directory entry CB for input file if open
     |  TK.MERGDIRECB  Address of directory entry CB for merge source
0201 | TK.INPTFILECB  Address of file CB for input file if open
     |  TK.MERGFILECB  Address of file CB for merge source file if open
0202 | TK.OUPTDIRECB  Address of directory entry CB for output file
     |  TK.MERGEXTSEG  Address of file CB for merge extract if allocated
0203 | TK.OUPTFILECB  Address of file CB for output file if open
0204 | TK.CF.DIRECB   Address of directory entry CB for common file rtn
0205 | TK.CF.FILECB   Address of file CB for common file routines
0206 | TK.CF.BUFFER   Address of buffer address for common file routines
0207 | TK.CF.LENGTH   Sindsk record length for common file routines
0210 | TK.FLDNEXT     BP address of start of search
0211 | TK.LINEND      BP address of end of search plus 1
0212 | TK.FLDSTART    BP address of start of current field
0213 | TK.FLDWIDTH    Length of current field in bytes
0214 | TK.OPRSAV      Save for err msg and commands (PF6)
0215 | TK.OPRFLG
0216 | TK.OPRSV2
0217 | TK.OPRPRI
0220 | TK.CMDISP      ASCII inserted log data (SBDM2R/SBDLMR)
     |  TK.RETADDR    Return address (SBDCHR)
     |  TK.ROW        Row on screen (EDGET/EDPUT-SBDLRC)
0221 | TK.SEQINCR     3 wds for key incr (SEQUENCE-SBDECM)
     |  TK.MOVEIT     MOVE instruction (SBDCHR)
     |  TK.RECORDLOC  3 wd block for location (SBDECM+, SEE BELOW)
     |  TK.DIRECBPTR  Addr saves for GETDIRECB
     |  **TKD.NEWDIRECB - DIRECB is new and not yet valid if bit 1 set
0222 |  TK.FILEIDPTR  (RESERV/ACCESS-SBDFAC)
     |  TK.PASSCOUNT  Password retry count (PASCHK-SBDSEC)
0223 |  TK.REQTABLE   DISC request table (SEE BELOW)
0224 | TK.SEQSTART    3 wds for key strt (SEQUENCE-SBDECM)
     |  TK.SAVNEXT    Parser working table (GETUSR-SBDSIN)
0225 | TK.SAVWIDTH
0226 | TK.SAVEND
0227 | TK.SAVSTART
0230 | TK.CBEND      BSS 0
     |  TK.CBLENGTH   EQU TK.CBEND+7/8*8
****************************************************************************
0221 | TK.RECORDLOC  3 wd block for location (SBDECM+)
     |  TK.KEY        for both ext and int locates
     |  TK.MARK
     |  TK.OFFSET
     |
0223 |  TK.REQTABLE   DISC request table
     |  TK.REQSTATUS  for directory routines
     |  TK.REQDEVICE  (RESERV/ACCESS/INFORM/RELEAS-SBDFAC)
     |  TK.REQBUFFER
     |  TK.REQSECTOR
     |  TK.REQQUEUE
```

WORD EXPANSION(S)

TK.TERMINAL **

BITS	SYMBOL	MEANING
0		Always ZERO (to prevent post)
1-7	TKT.TASKMASK	Task number (ZERO BASED INDEX)
8	TKT.ENTERED	Edit work file is currently entered
9	TKT.LOGOFF	Forced logoff of this terminal
10	TKT.MESSAGE	Message pending for this terminal
11	TKT.TABSET	TABSET in process
12	TKT.NOREFRESH	Console/monitor refresh temp disable
13	TKT.MAPPED	This TASKCB is mapped (around screen)
14	TKT.BACKGND	This a background task (not terminal)
15	TKT.NOPASS	Bit set bypasses file password checks
16-23	TKT.TERMASK	Physical terminal number

TK.OPTION **

SYMBOL	VALUE	MEANING
TKO.NULL	EQU 0	Undefined - terminal belongs to MFE
TKO.LOGON	EQU 1	Logo screen is ready to accept a logon
TKO.IDLE	EQU 2	Looking at option selection menu
TKO.VIEW	EQU 3	VIEW a file - any type, no changes
TKO.EDIT	EQU 4	EDIT a file - LE 80 columns, changes
TKO.MON	EQU 5	Looking at PWS system monitor screen
TKO.DIR	EQU 6	PWS file directory display
TKO.CON	EQU 7	COnsole for messages and commands

TK.FLAGS ***

BITS	SYMBOL	MEANING
0	TKF.RESET	RESET requested
1	TKF.CURSORON	Cursor is on screen
2	TKF.LEAVE	Leave input after processing
3	TKF.NODISPLAY	Password checking in process
4	TKF.INHIBIT	Data keys inhibited
5	TKF.ALARM	Alarm on this terminal
6	TKF.OPERATOR	This terminal is sys OPR
7	TKF.LOCKED	Keyboard locked
8	TKF.LOCKSW	Leave locked on final exit
9	TKF.INHIBSW	Leave inhibited on final exit
10	TKF.FIELDS	Fields enabled
11	TKF.TRUNCATE	Truncation allowed
12	TKF.UPPERCASE	Translate input to upper case
13	TKF.LINENUM	Line number at left for EDIT
14	TKF.TOFSEEN	Top of file already seen on screen
15	TKF.BOFSEEN	Bottom of file already seen on screen
16	TKF.NOCURSOR	No cursor on this screen split
17	TKF.DATACURS	Real cursor in data area (vs. cmd area)
18	TKF.FILECHANG	File changed since last save
19	TKF.LINEINS	Line insert mode is active
20	TKF.CHARINS	Char insert mode is active
21	TKF.READCURS	Screen is read only
22	TKF.NORMLEAVE	Normal enter leaves command in place

TK.PAGESIZE ***

SYMBOL	VALUE	MEANING
TKP.FULL	EQU -1	Calculate data lines - 1
TKP.HALF	EQU -2	Calculate data lines / 2

TK.CONSAUTH ***

SYMBOL	VALUE	MEANING
TKC.NONE	EQU 0	No access allowed
TKC.PRESETS	EQU 1	Only preset commands allowed
TKC.ALL	EQU 2	All commands allowed

TK.TEMPMARKS ***

BITS	SYMBOL	MEANING
0	TKT.SAMELINE	Both temp marks are set on same line
1-21	RESERVED	
22-23	TKT.DEFINED	0 = None, 1 = One defined, 2 = Both defined

TK.DIRECTION ***

BITS	SYMBOL	VALUE	MEANING
0-7	TKD.UP	EQU 1	Process upward
	TKD.DOWN	EQU 2	Process downward
8-15	RESERVED		
6-23	TKD.ALL	EQU 1	Process automatically
	TKD.LINE	EQU 2	Process only on current line
	TKD.VERIFY	EQU 3	Process only if verified

(The following words are valid for terminals only)

LOC	SYMBOL	MEANING
-01540	TK.CURSORINX	Cursor location line/column
-01537	TK.DATACURINX	Data cursor line/column
-01536	TK.CMDCURINX	Command cursor line/column
-01535	TK.CURSORPTR	Cursor location byte pointer
-01534	TK.UNDERWORD	Contents word under cursor
-01533	TK.CURSORPTR2	2nd curs loc bytptr (FIND) (0 if none)
-01532	TK.UNDERWORD2	Contents word under 2nd cursor (FIND)
-01531	TK.CURSORPTR3	3rd curs loc bytptr (FIND) (0 if none)
-01530	TK.UNDERWORD3	Contents word under 3nd cursor (FIND)
-01527	TK.CURSORCNT	Cursor blink count
-01526	TK.CURFIELD	Index to current entry in fields table
-01525	TK.XLATEKEY	Addr of key translate table
-01524	TK.KEYSTROKE	Current keystroke
-01523	TK.PREVKEY	PREVIOUS KEYSTROKE
-01522	TK.FUNKEY	
-01521	TK.KEYMODPTR	Pointer to keystroke modifier if any
	TKK.NOTFIRST	Not first keystroke after modifier if bit 0 set
-01520	TK.VIDEOBEG	Addr of video area begin
-01517	TK.VIDEOEND	Addr of video area end(+1)
-01516	TK.LEFTMARG	Current left cursorinx limit
-01515	TK.RIGHTMARG	Current right cursorinx limit
-01514	TK.FIRSTDATA	First logical screen data line (0 base)
	TK.FIRSTLINE	First logical screen line
-01513	TK.LASTDATA	Last logical screen data line
-01512	TK.LINESDATA	Total logical screen data lines
-01511	TK.STATUSLINE	Logical screen status display line
	TK.MSGLINE	Logical screen message line
	TK.LASTLINE	Last logical screen line

LOC	SYMBOL	MEANING
-01510	TK.INPUTLINE	Logical screen command input line
	TK.PROMPTLINE	Logical screen prompt line
-01507	TK.LINES	Total logical screen lines
-01506	TK.TEMPMARKS	Temporary mark status
-01505	TK.ACCEPTOR	Saved context of input acceptor
-01504	TK.OLDSTRPTR	Byte ptr address of old string
-01503	TK.OLDSTRLEN	Length in bytes of old string
-01502	TK.NEWSTRPTR	Byte ptr address of new string
-01501	TK.NEWSTRLEN	Length in bytes of new string
-01500	TK.SEARCHPTR	Current search location, byte ptr
-01477	TK.SEARCHINX	Current search location, line/col
	TKS.FIRSTTIME	Starting line being searched if bit 0 set
-01476	TK.LEFTLIMIT	Leftmost column to start search
-01475	TK.RIGHTLIMIT	Rightmost column to start search
-01474	TK.DIRECTION	Direction of search/type of change
-01473		RESERVED 4 WORDS
-01467	TK.SCREENINX	Screen position line/column
-01466	TK.CONSOLE	
-01465	TK.PF5CB	Address of PF5 control block
-01464	TK.ACCOUNT2	Account ID, ASCII
-01461	TK.USER2	User ID, ASCII
-01456	TK.NAME2	File name, ASCII
-01453	TK.TYPE2	File type, ASCII
-01450	TK.DRIVE2	Logical drive
-01447		RESERVED
-01446	TK.MESSAGE	Buffer for messages/passwords
	TK.MSGLENGTH	EQU $-TK.MESSAGE*3 Message buffer length in bytes
	TK.SAVEFLAGS	Flag save area for password entry
-01445	TK.SAVERIGHT	Right margin save area for password
-01444	TK.SAVELEFT	Left margin save area for password
-01443	TK.SAVECURINX	Cursor position save for password
-01442	TK.SAVEPROMPT	Prompt line save for password entry
-01410		** END OF MESSAGE BUFFER **

FILE CONTROL BLOCK FORMAT DESCRIPTION

WORD	SYMBOL	MEANING
0	FL.USER	Pointer to user table claiming this FCB
01	FL.DIRECB	Pointer to the directory control block
02	FL.ALTFCB	Alternate FCB save location used in block rtns
03	FL.INPUT	Start of the input request table
04	FL.DRIVE1	The drive number
05	FL.INPUT.BUF	Sector buffer for input data stream
06	FL.INPUT.SEC	Sector address for input buffer
07		BSS 2 RESERVED
011	FL.OUTPUT	Start of the output request table
012	FL.DRIVE2	The drive number
013	FL.OUTPUT.BUF	Sector buffer for output data stream
014	FL.OUTPUT.SEC	Sector address for output buffer
015		BSS 2 RESERVED
017	FL.CUR.SEC	Sector address of the current record
020	FL.CUR.REC	Index to current record
021	FL.CUR.KEY	Highest key of current sector
024	FL.FILE.START	Start of the file - sector address
025	FL.FILE.END	End of the file - sector address
026	FL.FILESIZE	Count of sectors in the file
027	FL.STATE	Flip-flop state of file system
	*	= 0 if file not split
	*	= not 0 if file is split
	* FL.SIN.STATE	Bit 0 = state of sindsk file
	*	If bit 0 = 0 then opened for input
	*	If bit 0 = 1 then opened for output
030	FL.CODE.WORD	The PWS code word (inserted in each sector)
031	FL.TYPE	Type indicates if this file has compressed rcds/read only
032	FL.REC.LEN	Record size after compression/decompression.
033	FL.FLAG.BUF	Flag value passed during GET and PUT calls
034	FL.TEXT.BUF	Pointer to text buffer passed in GET and PULL calls
035	FL.SEQ.BUF	Pointer to sequence numbers passed in GET and PULL calls
036	FL.MARK1	State of temporary mark 1
037	FL.MARK2	State of temporary mark 2
	*	The states are < 0 mark is before
	*	= 0 mark does not exist
	*	> 0 mark is after
040	FL.GOV.MARK	Mark that governs the current position
041	FL.NEXT.MARK	Location of the next mark
043	FL.LAST.MARK	Location of the last mark
045	FL.TEMP	Temporary storage used by the file manager
046	FL.TEMP.ADR	Temporary storage used by the file manager
047	FL.TEMP1	Temporary storage used by the file manager
050	FL.TEMP2	Temporary storage used by the file manager
051	FL.TEMP3	Temporary storage used by the file manager
052	FL.DELSTART	Start of dealloc chain
053	FL.DELEND	End of dealloc chain
054	FL.SEQLEFT	Left prefix sequence length
055	FL.SEQRIGHT	Right postfix sequence length
056	FL.TEXTLEN	Text length this file
057	FL.REC.SIZE	Record size for this file
060	FL.LOW.KEY	Lowest key for this file
063	FL.HIGH.KEY	Highest key for this file
066	FL.MARKER	Start of the mark table for this file
067	FL.MARK.TABLE	Mark table (36 words)
0133	FL.REC.WRDLEN	Word length of a record

0134	FL.MARKADR	Start of mark table for this FILECB
0135	FL.LOCTEMP1	LOCATE - temporary location for keys
	FL.COPYTEMP	CPYPSN - temp flag location
	FL.TEMPMRKSEC	SETMRK - sector address to mark
0136	FL.SCO	GETREC PUTREC - SCR pair
	FL.TEMPMRKREC	SETMRK - Record to mark
0137	FL.TEMPMARK	SETMRK - temporary storage
0140	FL.LCO	GETREC PUTREC - LCR pair
	FL.LOCTEMP2	LOCATE - temporary location for keys
	FL.TEMPMARK2	SETMRK - temporary storage
0141	FL.TEMPNEWMRK	SETMRK - temporary storage
0142	FL.SCR	GETREC PUTREC - SCR instruction
	FL.CLRTEMP	CLRMRK - temporary storage
0143	FL.LCR	GETREC PUTREC - LCR instruction
	FL.LOCTEMP3	LOCATE - temporary location for keys
	FL.SKPTEMP	SKPREC - temporary storage*
0144	FL.MBYTES	GETREC PUTREC - num of remaining bytes
0145	FL.SAVEREGA	GETREC - rec size on entry
0146	FL.NXTSCT	GETREC - next sector
0147	FL.CURSCT	GETREC - current sec addr
0150	FL.REMCNT	GETREC PUTREC - remaining buf size
0151	FL.FIRSTSEC	PUTREC - first sector address
0152	FL.LIMIT	GETREC - limiting factor
0153	FL.LASTSEC	PUTREC - last sec addr
0154	FL.DUPCHR	PUTREC - dup char
0155	FL.LASTCHAR	PUTREC - last char stored
0156	FL.TEMPREQ	SETMRK - disk request block
0164	FL.MRKSEC	Pointer to the sector data
0165	FL.MRKREC	Pointer to the record data
0166	FL.TEMP1SEC	Block routines - temp1 sector addr
0167	FL.TEMP1REC	Block routines - temp1 rec offset
0170	FL.TEMP2SEC	Block routines - temp2 sector addr
0171	FL.TEMP2REC	Block routines - temp2 rec offset
0172	FL.FIRSTMARK	Block routines - first temp mark
0173	FL.LASTMARK	Block routines - last temp mark
0174	FL.COUNT	COPY routines down counter
0175	FL.COUNTLOC	Location for counter display

FILE TYPE CONTROL BLOCK FORMAT DESCRIPTION

WORD	SYMBOL	MEANING
0	FT.TABS	Tab stop bits
07	FT.FIELDS	Field/margin columns
017	FT.LASTFIELD	Index to last entry in above
020	FT.INCREMENT	Default sequence increment ASCII DEC
023	FT.CASE	Upper case translate flag
024	FT.SEQLEFT	Length of left sequence if any
025	FT.SEQRIGHT	Length of right sequence if any
026	FT.TEXTLEN	Length of text
027	FT.RECLEN	Total record length

DIRECTORY CONTROL BLOCK FORMAT DESCRIPTION

WORD	SYMBOL	MEANING
0	DE.NEXT	Pointer to next DIRECB in chain
01	DE.ACCOUNT	Account ID, ASCII
04	DE.USER	User ID, ASCII
07	DE.NAME	File name, ASCII
012	DE.TYPE	File type, ASCII
015	DE.DRIVE	Logical drive
016	RESERVED	
017	DE.READPASS	READ password, ASCII
022	DE.WRITEPASS	WRITE password, ASCII
025	DE.NEWUSER	Creation user, ASCII
030	DE.NEWDATE	Creation date, ASCII
033	DE.NEWTIME	Creation time, ASCII
036	DE.MODUSER	Last mod user, ASCII
041	DE.MODDATE	Last mod date, ASCII
044	DE.MODTIME	Last mod time, ASCII
047	DE.LASTUSER	Last access user, ASCII
052	DE.LASTDATE	Last access date, ASCII
055	DE.LASTTIME	Last access time, ASCII
060	DE.RECORDS	Total records
061	DE.SECTORS	Total sectors (-1 for DIR)
062	DE.CRASHFLAG	Crash flag (077777777 if set)
063	DE.LASTSEC	Last sector address
064	DE.FILEFLAG	File flag bits and char
065	DE.SECCOUNT	Sector count - 1
066	DE.FIRSTSEC	First sector address
067	DE.EXTRA	PWS user ID of owner, ASCII
071	DE.MODTASK	Pointer to task with modify access (RESERVE)
072	DE.MODTASK2	Other task with access due to split screen
073	DE.ACCESSES	Count of number of read/only accesses
074	DE.READTASKS	Max of five tasks with read access
0101	DE.FLAGS	Flags (see EXPANSION)
0102	DE.CATSECTOR	Catalogue sector for this entry
0103	DE.DIRSECTOR	Directory sector for this entry

DE.FLAGS

BITS	SYMBOL	MEANING	
0	DEF.EXISTS	Directory entry (and file) exists on disc	
1	DEF.MODIFIED	Contents of file were modified	
2	DEF.EMPTY	No data sectors - only directory sector	
3-15		RESERVED	
6-23	DEF.NOTSET	EQU 0	Undetermined format
	DEF.PWSEDIT	EQU 1	PWS EDIT format
	DEF.SINDSK	EQU 2	IDOS SINDSK format
	DEF.CONTIG	EQU 3	IDOS CONTIGUOUS format

CCBTAB -- Communications Control Block Table

One CCBTAB entry per configured Communications PORT.
Inactive PORTS have zero COMMCB addresses.
Active PORTS contain the COMMCB address for the line.

COMMCB -- Communications Control Block

Word	Symbol	Usage
0	CM.TYPE	Line Protocol 0 - Line not active 1 - Model 20 communications (MLAM) 2 - 2780 communications (LAM) 3 - 3780 communications (LAM)
01	CM.LINENUM	Logical 8437 to which this line is assigned.
02	CM.LINECB	Address of the line control block.
03	CM.LINEID	The ASCII ID of the queue assigned to this line.
04	CM.QUEUECB	Queue CB of the queue assigned to this line.
05	CM.MONITIND	Monitor screen displacement of the Line-Status indicators for this line.
06	CM.MONITXMT	Monitor screen displacement of the Transmit record counts and current file name display.
07	CM.MONITRCV	Monitor screen displacement of the Receive record counts and current file name display.
010	CM.NEWFILEDB	The NEWFILE data block for processing $$NEWFILE records encountered in received punch files.
010	CMN.FILEID	The File ID data block of the file opened by the routine NEWFIL.
013	CMN.USERID	The User ID of the file opened by NEWFIL.
026	CMN.DIRECB	The Directory Entry CB of the file opened by NEWFIL.
027	CMN.ERRFLAGS	Error flags returned by the routine NEWFIL.
030	CMN.TERMIN	Three character file terminator.
031	CMN.ACTFLAGS	Flags indicating actions to perform on the file opened by NEWFIL. (ie. Replace and Queue)
032	CMN.QUEUEID	Upon receiving the entire file, the ID of the queue into which this file is to be enqueued.
033	CMN.QUEUEHD	Address of the head of the queue identified by CMN.QUEUEID.
034	CM.INCLUDEDB	The INCLUDE data block for processing $$INCLUDE records encountered in transmitted files.
037	CMI.USERID	The User ID of the file reserved by the routine INCLUD.
053	CMI.ERRFLAGS	Error flags returned by the routine INCLUD.
056	CM.XFLAGS	Transmit Task Flags. (See Transmit Flags below).
	CMF.ATTEN	Bit 0 - Post the transmit task
	CMF.CANCEL	Bit 1 - Cancel and dequeue the file being transmitted
	CMF.NOXMIT	Bit 2 - Disable transmission, unused
	CMF.STOP	Bit 3 - Stop command entered for this line
	CMF.RESTART	Bit 4 - MLAM - Host restarted, resend current file. LAM - LNELAM problem, resend current file.
	CMF.COMMAND	Bit 5 - Console message awaiting transmission.
	CMF.SIGNON	Bit 6 - Signon awaiting transmission.
	CMF.ABORT	Bit 7 - Abort communications on this line. Caused by an ABORT command or a fatal access-method error.
	CMF.STOPPING	Bit 8 - Stop pending for this line.
	CMF.WAKEUP	Bit 9 - LNELAM error needing immediate attention (LAM only)
	CMF.SIGNOFF	Bit 10- Signoff awaiting transmission.
	CMF.RECEIVE	Bit 11- Contention. Open for receive (LAM only).

057	CM.RFLAGS	Receive Task Flags. (See Receive Flags below).
	CMF.ATTEN	Bit 0 - Post the transmit task
	CMF.NORECV	Bit 2 - Disable receiving, unused
	CMF.STOP	Bit 3 - Stop command entered for this line
	CMF.RESTART	Bit 4 - MLAM - Host restarted, file incomplete. LAM - LNELAM problem, file incomplete.
	CMF.ABORT	Bit 7 - Abort communications on this line. Caused by an ABORT command or a fatal access-method error.
	CMF.STOPPING	Bit 8 - Stop pending for this line.
	CMF.PUNCH.FLS	Bit 15- Punch is flushing for $$NEWFILE card.
	CMF.PRINT.0	Bit 16- Print stream 0 is active.
	CMF.PRINT.1	Bit 17- Print stream 1 is active.
		Bit 18-19 Reserved for print stream 2 - 3.
		Bit 20-21 Reserved for punch stream 3 - 2.
	CMF.PUNCH.1	Bit 22- Reserved for punch stream 1.
	CMF.PUNCH.0	Bit 23- Punch stream 0 is active.
060	CM.XMTTASK	Transmit Task CB Address.
061	CM.RCVTASK	Receive Task CB Address.
062	CM.PU1FILECB	Address of the punch stream File CB. If no punch file is currently opened, this location is zero.
063	CM.RDRNEST	Number of $$INCLUDE files currently opened on the reader.
064-073	CM.RDRFILECB	File CB addresses of all files currently opened on the reader.
076-101	CM.PRTALLOC	Printer allocation words, one per print stream.
0102-105	CM.SPOOLALLOC	Virtual printer number (zero if no virtual printer allocated), one per print stream.
0106	CM.PRINTLAST	Number of the last printer stream serviced.
0107	CM.CBEND	Last word of the Comm CB. Unused.

MLINCB -- MLAM Line Control Block

Word	Symbol	Usage
0	ML.SNDRQT	$XFER Request table for Sending and Control functions.
010	ML.RCVRQT	$XFER Request table for Receiving.
020	ML.INITTED	Initialized flag. (0 = INTM20 not completed.)
	MLC.AUTOANS	Bit 0 - Auto answer configuration
	MLC.TERM	Bit 1 - Terminal to terminal configuration
	MLC.INITTED	Bit 23 - MLAM initialized
021	ML.LOGGING	Logging value.
		0 - No logging.
		1 - Log only line data.
		2 - Log only $XFER request/response tables.
		3 - Log line data and $XFER tables.
022	ML.MAPBUFFER	Mapped buffer flag.
		Bit 0 = True, Mapped buffers in use.
023	ML.XPARENT	Transparency Indicator.
	MLX.OUTXPARENT	Bit 0 - Transmit transparent
	MLX.OUTXLATE	Bit 1 - Translate transparent transmitted data
	MLX.INXLATE	Bit 2 - Translate transparent received data
024	ML.MONIND	Monitor screen offset - line indicator display.
025	ML.CONFIGID	Configuration File Identifier (USER.FILE.TYPE@DRIVE).
043	ML.STATS	MLAM - %STATS Status indicator
044	ML.INDEX	MLAM - %INDEX Current state indicator
045	ML.INSTREAMS	MLAM - %LNSTR Acceptable stream mask
		Bit 16 - Accept printer stream 0
		Bit 17 - Accept printer stream 1
		Bit 22 - Accept punch stream 1
		Bit 23 - Accept punch stream 0
046	ML.OUTSTREAMS	MLAM - %LNSTD Transmit stream number
047	ML.AVAILREC	MLAM - @AVLRC Available records mask
		Bit 15 - Record for console
		Bit 16 - Record for printer stream 0
		Bit 17 - Record for printer stream 1
		Bit 22 - Record for punch stream 1
		Bit 23 - Record for punch stream 0
050	ML.STREAMIN	MLAM - @STRIN Received stream number
051	ML.RECTYPE	MLAM - RECTYP Record type
052	ML.CARRIAGE	MLAM - RECCC Carriage control character
053	ML.RECLENGTH	MLAM - RECLEN Received record length
054	ML.FLUSHFLAG	MLAM - POSTFL Transmit flush flag
055	ML.INBUFFERMLAM receive record buffer.	
0135	ML.OUTBUFFER	MLAM transmit record buffer.

LINECB -- LAM Line Control Block

Word	Symbol	Usage
0	LN.LAMRQT	$XFER table for requests to the 8437.
010	LN.LAMRSP	$XFER table for responses to the 8437.
020	LN.ABTRQT	$XFER table for ABTLNE requests to the 8437.
030	LN.CTLRQT	$XFER table for DEVLNE requests to the 8437.
040	LN.INITTED	LAM initialized flag. (0 = INTLNE not completed.)
041	LN.LOGGING	Logging value. 0 - No logging. 1 - Log only line data. 2 - Log only $XFER request/response tables. 3 - Log line data and $XFER tables.
042	LN.MAPBUFFER	Mapped buffer flag. Bit 0 = True, Mapped buffers in use.
043	LN.XPARENT	Transparency Indicator.
	LNX.OUTXPARENT	Bit 0 - Transmit transparent
	LNX.OUTXLATE	Bit 1 - Translate transparent transmitted data
	LNX.INXLATE	Bit 2 - Translate transparent received data
044	LN.MONIND	Monitor screen offset - line indicator display.
045	LN.COMPRESS	LAM Compression type.
046	LN.COMPRESET	LAM Configured compression.
047	LN.CONFIGID	Configuration File Identifier (USER.FILE.TYPE@DRIVE).
070	LN.LNELAM	LAM - LNELAM Line status indicator
071	LN.INDEX	LAM - $INDEX Current state indicator
072	LN.INSTREAMS	LAM - $LNPUN,$LNPRT Current device status
	LNR.PUNCH	Bit 7 - Punch ready to receive
	LNR.PRINTER	Bit 15 - Printer ready to receive
073	LN.LOGICERROR	LAM - $LOGIC Logic error indicator
074	LN.HOSTRVI	LAM - $LNRVI RVI received indicator
075	LN.HOSTADR	LAM - $LNADR Bid received indicator
076	LN.DATASET	LAM - $LNDSR Data Set Ready indicator
077	LN.RECTYPE	LAM - RECTYP Received record type indicator
0100	LN.RECLENGTH	LAM - RECLEN Received record length
0101	LN.OPEN	LAM - OPEN? Line opened indicator
0102	LN.HUNG	LAM - HUNG? HNGLNE completed indicator
0103	LN.BUFFER	LAM transmit and receive record buffer.

PRNTCB -- Printer Control Block

Word	Symbol	Usage
0	PR.RETURN	Return from interrupt address (not used under MFE)
01	PR.BRM1	BRM to MFE STUB (not used under MFE)
02	PR.CUTWORD	Printer CUT word
03	PR.BRM2	BRM to MPKICK (not used under MFE)
04	PR.FILEID	Printer File ID (ACCOUNT.USER.FILE.TYPE@DRIVE)
022	PR.MONITNAME	Monitor screen displacement for file display
023	PR.FLAGS	Action flags:
	PR.ATTEN	Bit 0 - Attention
	PR.ERROR	Bit 1 - Bad line printer status
	PR.FLASH	Bit 2 - Message is being flashed
	PR.MAX	Bit 3 - Printing at maximum speed
	PR.CANCEL	Bit 4 - Printer canceled (unused)
	PR.SUSPEND	Bit 5 - Printer suspended (unused)
	PR.STOP	Bit 6 - Stop command (unused)
	PR.CLOSE	Bit 7 - Close has been issued
024	PR.SWITCH	Current Allocation Word
025	PR.ID	Printer number (1 - 4)
026	PR.NEXTBUFF	Address of next buffer
027	PR.THISBUFF	Address of record being printed
030	PR.WAITTASK	Address of waiting TASK CB
031	PR.CHARSET	Character set
032	PR.TRANSLATE	Translation table index
033	PR.BUFFTOTAL	Total print record buffers
034	PR.BUFFEMPTY	Number of empth print record buffers
035	PR.CBEND	Unused

CARDCB -- Card Reader Control Block

Word	Symbol	Usage
0	CD.RETURN	Return from interrupt address (not used under MFE)
01	CD.BRM1	First level interrupt (MFE stub - not used under MFE)
02	CD.CUTWORD	Card Reader CUT word
03	CD.BRM2	BRM to CDKICK (not used under MFE)
04	CD.NEWFILEDB	NEWFILE Data Block, for use by NEWFIL
04	CDN.FILEID	File ID Data Block (ACCOUNT.USER.FILE.TYPE@DRIVE)
012	CDN.NAME	File name
024	CDN.TERMIN	File terminator characters
025	CDN.ACTFLAGS	Action flags
027	CDN.QUEUEHD	Queue Head address
030	CD.MONITNAME	Monitor screen offset to file name display
031	CD.FLAGS	Flag word
	CDF.ATTEN	Bit 0 - Attention
	CDF.STOP	Bit 1 - Stop requested (at shutdown)
	CDF.ABORT	Bit 2 - Abort requested
	CDF.ACTIVE	Bit 3 - Reader has a file opened
032	CD.STATUS	Hardware status word
033	CD.BUFFER	Address of the record buffer
034	CD.WAITTASK	Address of the TASKCB
035	CD.TASKECB	Task Event Control Block
036	CD.STATE	Card reader driver routine state index
037- 071	CD.RECORDBUFF	80 character record buffer

SPOLCB -- Virtual Printer Control Block

Word	Symbol	Usage
0	SP.STATE	State of the Virtual Printer
	SPS.AVAILABLE	0 - Not allocated
	SPS.SKIPPING	1 - Skipping (Start of print to start line number)
	SPS.SEARCHING	2 - Searching for $$SPOOL record
	SPS.WRITING	3 - Writing to a disc file ($$SPOOL found)
	SPS.PRINTING	4 - Printing ($$SPOOL not found)
01	SP.FLAGS	Unused
02	SP.COMMFILECB	Common File Control Block
06	SP.INDEX	Virtual printer index
07	SP.ID	ASCII virtual printer number
010	SP.PHYSICAL	Physical printer number - 0 if none allocated
011	SP.TRANSLATE	Translation table index
012	SP.EJECTCOUNT	Count of page ejects
013	SP.RECCOUNT	Count of records
014	SP.RECORDKEY	Key appended to each record (record number)
017	SP.CBEND	Unused

DIRECTORY SECTOR FORMAT DESCRIPTION

WORD	SYMBOL	MEANING
0	DS.HEADER	Sector header (see EXPANSION)
02		RESERVED
010	DS.DBID	Lit flags this as PWS DIRSEC 'PWS' protected HO bits
011	DS.ACCOUNT	Account ID, ASCII
014	DS.USER	User ID, ASCII
017	DS.NAME	File name, ASCII
022	DS.TYPE	File type, ASCII
025	DS.DRIVE	Logical drive
026		RESERVED
027	DS.READPASS	READ password, ASCII
032	DS.WRITEPASS	WRITE password, ASCII
035	DS.NEWUSER	Creation user, ASCII
040	DS.NEWDATE	Creation date, ASCII
043	DS.NEWTIME	Creation time, ASCII
046	DS.MODUSER	Last mod user, ASCII
051	DS.MODDATE	Last mod date, ASCII
054	DS.MODTIME	Last mod time, ASCII
057	DS.LASTUSER	Last access user, ASCII
062	DS.LASTDATE	Last access date, ASCII
065	DS.LASTTIME	Last access time, ASCII
070	DS.RECORDS	Total records count
071	DS.CRASHFLAG	Crash flag (077777777 if set)
072	DS.SECTORS	Total sectors (-1 for DIR)
073	DS.LASTSEC	Last sector address
074	DS.FILEFLAG	File flag bits and char
075	DS.SECCOUNT	Sector count - 1
076	DS.FIRSTSEC	First sector address
077	DS.EXTRA	PWS user ID of owner, ASCII
0101	DS.ACCOUNT2	Account ID, ASCII
0104	DS.USER2	User ID, ASCII
0107	DS.NAME2	File name, ASCII
0112	DS.TYPE2	File type, ASCII
0115	DS.DRIVE2	Logical drive
0116		RESERVED

DS.HEADER

WORD	BITS	MEANING
0	0-11	IDOS constant(=01372) to specify whole sector
	12-23	First 12 bits previous sector address
1	0-5	Last 6 bits of previous sector address
	6-23	Next sector address

HEADER SECTOR FORMAT DESCRIPTION

```
WORD   SYMBOL         MEANING
   0 | HS.HEADER      Sector header (see EXPANSION)
  04 |                RESERVED
010 | HS.TABS.       TASKCB TABS
    | ** 7 word table with each bit corresponding to a tabbed column
017 | HS.FIELDS      TASKCB FIELDS
    | ** 8 word table each word describing a field column number + 1,
    | ** excepting the last entry, the right margin, set to its true value
027 | HS.LASTFIELD   TASKCB last field index (rightmarg)
 030 | HS.INCREMENT  TASKCB default sequence increment
 033 | HS.CASE       TASKCB default CASE switch
     |                 0=input either case
     |                -1=input as upper case
034 | HS.SEQLEFT     FILECB left sequence length
 035 | HS.SEQRIGHT   FILECB right sequence length
 036 | HS.TEXTLEN    FILECB text length
 037 | HS.RECLEN     FILECB total record length
 040 | HS.LOWKEY     FILECB lowest key of the file
 043 | HS.HIGHKEY    FILECB highest key of the file
 046 | HS.MARKPTR    FILECB pointer to first mark
 047 | HS.MARKS      FILECB saved absolute mark table
0113 | HS.END        END
```

SECTOR HEADER WORD	BITS	MEANING
0-1		Standard IDOS sector header (see DIRECTORY SECTOR DESC.)
2	0-23	File header sector address
3	0	If set indicates '* START OF FILE *' sector
	1	'* END OF FILE *' sector
	2-3	RESERVED
	4	Header sector

DATA SECTOR FORMAT DESCRIPTION

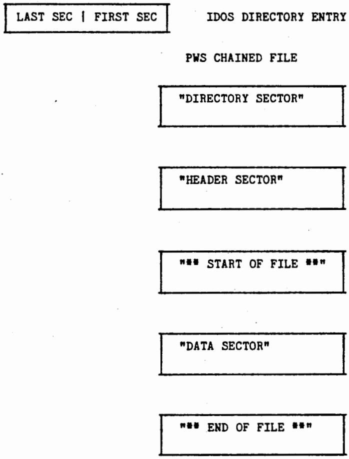
WORD	BITS	MEANING
0-1		Standard IDOS sector header (see DIRECTORY SECTOR DESC.)
2	0-23	Header sector address
3	0	If set indicates '* START OF FILE *' sector
	1	'* END OF FILE *' sector
	2-3	RESERVED
	4	Header sector
4-6		RESERVED
7	0-7	FL.ROOM - room left in sector
	8-15	FL.LAST.REC - index to last record pointer
	16-23	FL.FIRST.REC - index to first record pointer

10-377 Chained data records as described below

0	0-7	FL.SIZE.REC - record size	
	8-15	FL.BACK.REC - index to record back pointer	
	16-23	FL.FORE.REC - index to record forward pointer	
1-3		** 9 digit ASCII sequence number, right justified/zero filled	
4	0	LRF.MODIFIED	record has been modified
	1	LRF.TOP	record is the start of file dummy record
	2	LRF.BOTTOM	record is the end of file dummy record
	3*	LRF.HIGHLIGHT	record between pointers should be highlighted
	4	RESERVED	
	5	LRF.NOTCOMP	record is not compressed
	6	MKB.USER1	** User definable marks **
	7	MKB.USER2	
	8	MKB.USER3	** For all marks, if bit is set then mark points at this record **
	9	MKB.USER4	
	10	MKB.USER5	
	11	MKB.USER6	
	12	MKB.USER7	
	13	MKB.USER8	
	14	MKB.USER9	
	15	MKB.USER0	
	16	MKB.APLSAVE	Internal temporary mark for application save
	17	MKB.FIOSAVE	Internal temporary mark for file IO save
	18	MKB.TOP	Top of file mark
	19	MKB.BOTTOM	Bottom of file mark
	20	MKB.TEMP1	First temporary mark
	21	MKB.TEMP2	Second temporary mark
	22	MKB.CURLINE1	Split 1 current line mark
	23	MKB.CURLINE0	Split 0 current line mark
5-XX		** DATA RECORD - SEE DATA COMPRESSION DESCRIPTION **	

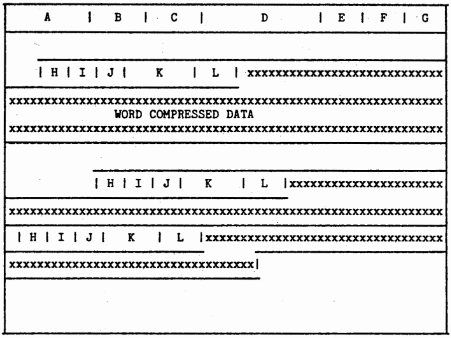
*Used only in memory - not stored on disk

FORMAT OF PWS CHAINED FILE



PWS files are chained with the standard IDOS chained file structure with internal enhancements defined in the file itself. See the descriptions of the Directory, Header, and Data sector formats.

PWS DATA SECTOR FORMAT



Where: A = Standard IDOS chained sector header
B = PWS file header sector address
C = Sector flags
D = Reserved words in header
E = Room left in sector
F = Last sector offset
G = First sector offset
H = Record size
I = Back record pointer
J = Forward record pointer
K = 9 digit ASCII sequence number
L = Record flags
x = Word compressed data

For more detailed descriptions see the DATA SECTOR DESCRIPTION.
Each data sector is comprised of chained data records linked with forward and backward pointers, defining word offsets from the beginning of the sector. valid word offsets range from 010 - 0372, allowing for the 8 data sector header words and a 5 word record header for each record.

COMPONENT LIST

Most PWS file names are of the form tBDxxx where t = type of file:

S for MACROL source
P for Pascal source
M for MACROL macro library source
R for Relocatable
L for compiled MACROL macro library
A for Absolute
C for Control files
G for directory files
xxx = unique three characters identifying the file

BD03 is released on a WRTAPE, containing the following categories:

C	Control Files
A	Absolute Utility files
R	Relocatables
S	Source
G	Development Control Files and Utilities
D	Canned Demo Files

The categories below represent internal component categorization:

INT -- INITIALIZATION
SYS -- MAIN SYSTEM RESIDENT
TRM -- TERMINAL HANDLING
COM -- COMMAND PROCESSORS
BGD -- BACKGROUND PROCESSING
CMM -- COMMUNICATIONS INTERFACE
SPC -- SPECIAL ASSEMBLY OR COMPLILE REQUIREMENTS
CTL -- CONTROL FILES
MAC -- PWS MACRO LIBRARY FILES
PRD -- FILES NEEDED ON NEW CUSTOMER PACK
UPG -- FILES NEEDED TO UPGRADE EXISTING CUSTOMER PACK
TAP -- FILES NEEDED ON TAPE FOR TRANSFER TO CUSTOMER PACK
WRK -- ALL ELSE NEEDED FOR WORKING SYSTEM DURING DEVELOPMENT
UTL -- UTILITIES
PMP -- FILES TO BE PLACED ON THE PRODUCT MASTER PACK
DMO -- FILES USED FOR THE CANNED PWS DEMO

SEVERAL CATEGORIES ASSUME STRUCTURED NAMES E.G., S FOR SOURCE, R FOR RELOCATABLE, ETC. BOTH THE SOURCE AND THE RELOCATABLE NAME ARE INCLUDED IN THE DIRECTORY.

DIRECTORY FILE

GBDDIR,GBDDIR,C;. CTL,PMP.	PWS COMPONENT FILE
GBDINS,GBDINS,C;. CTL.	PMP INSTALLATION DIRECTORY FILE

SYSTEM CONTROL FILES

CBDCPS,CBDCPS,C;. CTL,PMP.	COPY FROM PMP TO CUS - >DO THIS FIRST<
CBDCPN,CBDCPN,C;. CTL,PMP.	COPY FROM PMP TO NEW SYSTEM
CBDCPU,CBDCPU,C;. CTL,PMP.	COPY FROM PMP TO UPGRADE
CBDCPT,CBDCPT,C;. CTL,PMP.	COPY FROM PMP TO TRANSFER TAPE
CBDCTN,CBDCTN,C;. CTL,TAP.	COPY FROM TRANSFER TAPE TO NEW SYSTEM
CBDCTU,CBDCTU,C;. CTL,TAP.	COPY FROM TRANSFER TAPE TO UPGRADE
CBDQFF,CBDQFF,C;. CTL,PRD,UPG,PMP.	FORMAT QUEUE FILE
CBDINS,CBDINS,C;. CTL.	DO PMP INSTALLATION
PWSGEN,PWSGEN,C;. CTL,PMP.	INVOKE PWSGEN

PRODUCTION PACK FILES

PAINT ,PAINT ,A;. PRD,UPG,PMP.	PAINT ABSOLUTE
ABDPWS,ABDPWS,A;. PRD,UPG.	PWS ABSOLUTE
RBDBAS,RBDBAS,C;. PRD,UPG.	LOADER SYMBOL SAVE FILE

```
   ABDRPT,ABDRPT,A;. PRD,UPG,PMP.    LOG REPORT PROGRAM ABSOLUTE
   ABDDXT,ABDDXT,A;. PRD,UPG,PMP.    DIRECTORY LIST EXTRACT ABSOLUTE
   ABDDLS,ABDDLS,A;. PRD,UPG,PMP.    DIRECTORY LIST SELECT/PRINT ABSOLUTE
   ABDFDM,ABDFDM,A;. PRD,UPG,PMP.    FORMATTED DUMP PROGRAM ABSOLUTE
   SBDLST,SBDLST,A;. PRD,UPG.        CONFIGURATION LIST FILE
   SBDPRM,SBDPRM,A;. .               CONFIGURATION EQUATE FILE
   PWSLOG,PWSLOG,A;. PRD,PMP.        IDLE SCREEN LOGO
   PWSFIG,PWSFIG,A;. PRD,PMP.        COMMUNICATION CONFIGURATION
   PWS   ,PWS   ,C;. PRD,UPG,PMP.    PWS START PROCEDURE
   PWSDIR,PWSDIR,C;. PRD,UPG,PMP.    PWS DIRECTORY UTILITY INIT PROCEDURE
   DIRLIS,DIRLIS,C;. PRD,UPG,PMP.    PWS DIRECTORY UTILITY SELECT PROCEDURE
   PWSREP,PWSREP,C;. PRD,UPG,PMP.    PWS LOG REPORT UTILITY PROCEDURE
   CBDQFF,CBDQFF,C;. PRD,UPG,PMP.    QUEUE FILE FORMATTER
   SORT60,SORT60,C;. PRD,UPG,PMP.    LARGE 8260 DIRECTORY SORT EXAMPLE
   DIRGET,DIRGET,A;. PRD,UPG,PMP.    LARGE 8260 DIRECTORY GET UTILITY
   DIRPUT,DIRPUT,A;. PRD,UPG,PMP.    LARGE 8260 DIRECTORY PUT UTILITY
   MLAMEP,MLAMEP,G;. PRD,UPG,WRK.    MLAM 8437 ABSOLUTE
   LAMEP ,LAMEP ,G;. PRD,UPG,WRK.    LAM 8437 EBCDIC POINT-TO-POINT ABSOULTE
   LAMAP ,LAMAP ,G;. PRD,UPG,WRK.    LAM 8437 ASCII POINT-TO-POINT ABSOLUTE
   LAMEM ,LAMEM ,G;. PRD,UPG,WRK.    LAM 8437 EBCDIC MULTIPOINT ABSOLUTE
   TRACE ,TRACE ,G;. PRD,UPG,WRK.    LINE TRACE INTERPRETER
   DTCOMM,DTCOMM,G;. PRD,UPG,WRK.    8437 DEBUGGING TOOL

PWS CANNED DEMO FILES
   JOBCRD,JOBCRD,D;. DMO.            BCS JOB CARD FOR CANNED DEMO
   BKDEMO,BKDEMO,D;. DMO.            SAMPLE COBOL SOURCE - ORIGINAL COPY
   SEND  ,SEND  ,D;. DMO.            JCL TO COMPILE SAMPLE COBOL SOURCE
   RETDSK,RETDSK,D;. DMO.            JCL TO RETRIEVE A FILE FROM BCS
   SMPDAT,SMPDAT,D;. DMO.            TEST DATA FOR SAMPLE COBOL PROGRAM
   PWSIDS,PWSIDS,D;. DMO.            CANNED DEMO ID'S FILE
   PWSOFF,PWSOFF,D;. DMO.            BCS SIGNOFF CARD
   PWSTXT,PWSTXT,D;. DMO.            IVWORD TEXT AREA WITH PWS DEMO DOCS

WORK STUFF - OTHER STUFF NEEDED TO HAVE A WORKING MASTER PACK
   ABDIVL,ABDIVL,A;. PMP.            INTERACTIVE VIDEO LIBRARY
   SELECT,SELECT,G;. WRK.            SELECT ABSOLUTE
   PWSQUE,PWSQUE,G;. WRK.            PREALLOCATED QUEUE SAVE FILE
   RDNX02,RDNX02,G;. WRK.            $XINIT
   SDNX04,SDNX04,G;. WRK.            $XFER TABLES SOURCE
   RBDX04,RBDX04,G;. WRK.            $XFER TABLES
   RDNX06,RDNX06,G;. WRK.            $XOPEN
   RDNX07,RDNX07,G;. WRK.            $XCLOSE
   RDNX08,RDNX08,G;. WRK.            $XLOG
   RDNX11,RDNX11,G;. WRK.            $XFER
   RDNX12,RDNX12,G;. WRK.            $XFER LEVEL 1 ROUTINES
   REHEL0,REHEL0,G;. WRK.            ELOG - LOG INITIALIZATION
   REHEL1,REHEL1,G;. WRK.            ELOG - LOG ROUTINE
   REHEL2,REHEL2,G;. WRK.            ELOG - LOG STUBS
   REB001,REB001,G;. WRK.            CLUAT TABLE
   PASCAL,PASCAL,G;. WRK.            PASCAL COMPILER
   RCULBB,RCULBB,G;. WRK.            PASCAL RECURSIVE LIBRARY
   PSCLIB,PSCLIB,G;. WRK.            PASCAL LIBRARY
   RDSUFA,RDSUFA,G;. WRK.            MFE NAME-TO-WINDOW ROUTINE

MACRO LIBRARY FILES
   MBDDAT,LBDDAT,C;. MAC.            DATA DEFINITIONS
   MBDEXT,LBDEXT,C;. MAC.            EXTERNAL DEFINITIONS
   MBD490,LBD490,C;. MAC.            PSEUDO-OPS IV/90 VERSION
   MBDSYS,LBDSYS,C;. MAC.            SYSTEM MACRO CALLS
   MBDMDS,LBDMDS,C;. MAC.            PWS APPLICATION MACRO CALLS
```

RESIDENT SYSTEM FILES.

SBD201,RBD201,S;. CMM.	LAM
SBD202,RBD202,S;. CMM.	LAM
SBD203,RBD203,S;. CMM.	LAM
SBD204,RBD204,S;. CMM.	LAM
SBD301,RBD301,S;. CMM.	MLAM
SBD303,RBD303,S;. CMM.	MLAM
SBD304,RBD304,S;. CMM.	MLAM
SBD305,RBD305,S;. CMM.	MLAM
SBDAPL,RBDAPL,S;. SYS.	MFE APL ROUTINES
SBDBKG,RBDBKG,S;. BGD.	BACKGROUND MISCELLANEOUS TASK
SBDCAT,RBDCAT,S;. SYS.	DIRECTORY CATALOG
SBDCCD,RBDCCD,S;. BGD.	CONTROL CARDS
SBDCF3,RBDCF3,S;. SPC,PMP.	SEGMENT THREE OPTIONAL CONFIGURATION
SBDCFL,RBDCFL,S;. SYS.	COMMON FILE SYSTEM
SBDCHG,RBDCHG,S;. SYS.	EDITOR CHANGE COMMAND
SBDCHR,RBDCHR,S;. TRM.	FUNCTION KEYS - CHAR INSERT/DELETE
SBDCMC,RBDCMC,S;. SPC,PMP.	COMMUNICATION CONFIGURATION FILE
SBDCML,RBDCML,S;. CMM.	COMMUNICATIONS LOG ROUTINES - SEGMENT
SBDCMR,RBDCMR,S;. CMM.	COMMUNICATIONS LOG ROUTINES - ROOT
SBDCMS,RBDCMS,S;. CMM.	COMMUNICATIONS SHARED ROUTINES
SBDCON,RBDCON,S;. SYS,PMP.	SYSTEM CONSTANTS
SBDCRD,RBDCRD,S;. BGD.	CARD READER
SBDCRS,RBDCRS,S;. TRM.	FUNCTION KEYS - CURSOR MOVEMENT
SBDCST,RBDCST,S;. TRM.	CURSOR LOCATION SETTING
SBDDSC,RBDDSC,S;. SYS.	DISK HANDLING
SBDECM,RBDECM,S;. TRM.	EDITOR COMMANDS
SBDEDT,RBDEDT,S;. TRM.	FUNCTION KEYS - FIELD/LINE/BLOCK EDIT
SBDEFA,RBDEFA,S;. SYS.	EDIT FILE ACCESS
SBDEMK,RBDEMK,S;. TRM.	FUNCTION KEYS - SET/LOCATE REF MARKS
SBDESL,RBDESL,S;. TRM.	EDIT SCROLLING
SBDFA2,RBDFA2,S;. SYS.	FILE ACCESS LOWER LEVEL ROUTINES
SBDFAC,RBDFAC,S;. SYS.	FILE ACCESS SHARED/EXCLUSIVE
SBDFCL,RBDFCL,S;. SYS.	FILE ACCESS COMMON FOR BACKGROUND
SBDFKY,RBDFKY,S;. TRM.	FUNCTION KEY DISPATCHER
SBDFL1,RBDFL1,S;. SYS.	FILE MANAGEMENT - SPLIT AND RELINK
SBDFL2,RBDFL2,S;. SYS.	FILE MANAGEMENT - GET AND PUT ROUTINES
SBDFL3,RBDFL3,S;. SYS.	FILE MANAGEMENT - OPEN/CLOSE ROUTINES
SBDFL4,RBDFL4,S;. SYS.	FILE MANAGEMENT - SERVICE SUBROUTINES
SBDFL5,RBDFL5,S;. SYS.	FILE MANAGEMENT - EDIT OPEN/CLOSE
SBDFL6,RBDFL6,S;. SYS.	FILE MANAGEMENT - MARKS
SBDFL7,RBDFL7,S;. SYS.	FILE MANAGEMENT - LOCATE
SBDFL8,RBDFL8,S;. SYS.	FILE MANAGEMENT - SKIP RECORDS
SBDFL9,RBDFL9,S;. SYS.	FILE MANAGEMENT - RESEQUENCE
SBDFLA,RBDFLA,S;. SYS.	FILE MANAGEMENT - SINDSK INTERFACE
SBDFLB,RBDFLB,S;. SYS.	FILE MANAGEMENT - COPY SUBROUTINES
SBDFLC,RBDFLC,S;. SYS.	FILE MANAGEMENT - EXTRACT DEL INS SEG
SBDFLD,RBDFLD,S;. SYS.	FILE MANAGEMENT -
SBDFND,RBDFND,S;. SYS.	FIND FOR EDITOR
SBDFTP,RBDFTP,S;. SYS,PMP.	FILE TYPE DEFINITIONS
SBDHSG,RBDHSG,S;. SYS.	HYPERSPACE DEFINITION
SBDIDR,RBDIDR,S;. INT.	DIRECTORY CATALOG INIT
SBDIL6,RBDIL6,S;. SYS.	LEVEL 6 INTERRUPT HANDLING
SBDINT,RBDINT,S;. SYS.	INTERRUPT HANDLING
SBDINZ,RBDINZ,S;. INT.	SYSTEM INITIALIZATION
SBDIZT,RBDIZT,S;. SPC,PMP.	OPERATOR SCREEN TABLES
SBDKEY,RBDKEY,S;. TRM.	KEYBOARD TRANSLATION TABLE
SBDL37,RBDL37,S;. CMM,PMP.	8437 LOADER
SBDLMI,RBDLMI,S;. CMM.	LAM INITIALIZATION
SBDLMR,RBDLMR,S;. CMM.	LAM RECEIVE ROUTINES
SBDLMT,RBDLMT,S;. CMM.	LAM TRANSMIT ROUTINES
SBDLNS,RBDLNS,S;. TRM.	SCREEN LINE HANDLING SUBROUTINES
SBDLOG,RBDLOG,S;. BGD.	EVENT LOGGING

SBDLRC,RBDLRC,S;. SYS.	LOGICAL RECORD ACCESS GET/PUT PWS
SBDM2I,RBDM2I,R;. REL,PMP.	
SBDM2R,RBDM2R,S;. CMM.	MLAM RECEIVE TASK
SBDM2T,RBDM2T,S;. CMM.	MLAM TRANSMIT TASK
SBDMEM,RBDMEM,S;. SYS.	MEMORY MANAGEMENT
SBDMIZ,RBDMIZ,S;. INT.	MEMORY MANAGEMENT INITIALIZATION
SBDMLP,RBDMLP,S;. BGD.	MULTIPLE PRINTER DRIVER
SBDMRG,RBDMRG,S;. TRM.	EDITOR MERGE COMMAND
SBDMSG,RBDMSG,S;. SPC.	ERROR MESSAGES
SBDOIZ,RBDOIZ,S;. INT.	OPERATOR SCREEN INIT
SBDOL0,RBDOL0,S;. COM.	QUEUE OPERATOR COMMANDS
SBDOL1,RBDOL1,S;. COM.	OPERATOR SYSTEM COMMANDS
SBDOL2,RBDOL2,S;. COM.	OPERATOR DEVICE COMMANDS
SBDOL3,RBDOL3,S;. COM.	OPERATOR PRINT ALLOC COMMANDS
SBDOPR,RBDOPR,S;. TRM.	OPERATOR COMMANDS
SBDOPT,RBDOPT,S;. TRM.	FUNCTION KEYS - OPTION SWITCHING
SBDPER,RBDPER,S;. SYS.	INTERNAL PERFORMANCE MONITOR
SBDPF5,RBDPF5,S;. TRM.	DIRECTORY LIST
SBDPF6,RBDPF6,S;. TRM.	CONSOLE VIEWING
SBDPRT,RBDPRT,S;. BGD.	PRINTER
SBDPS2,RBDPS2,S;. SYS.	STRING PARSING AND LOOKUP CONTINUED
SBDPSR,RBDPSR,S;. SYS.	STRING PARSING AND LOOKUP
SBDQUE,RBDQUE,S;. SPC.	QUEUE/DEQUEUE
SBDQWR,RBDQWR,S;. COM.	QUEUE WRITE TO DISC
SBDRDC,RBDRDC,S;. SYS.	FILE RENAME/DELETE/COPY
SBDRFS,RBDRFS,S;. TRM.	CONSOLE/MONITOR SCREEN REFRESH
SBDSOD,RBDSOD,S;. SYS.	SCREEN ZERO DISPLAY
SBDSCL,RBDSCL,S;. TRM.	FUNCTION KEYS - DATA SCROLLING
SBDSEC,RBDSEC,S;. SYS.	FILE SECURITY CHECKS
SBDSET,RBDSET,S;. SYS.	EDITOR SET STUFF
SBDSIN,RBDSIN,S;. TRM.	SIGN ON/OFF
SBDSPL,RBDSPL,S;. BGD.	VIRTUAL PRINTER SPOOLING
SBDSPT,RBDSPT,S;. TRM.	FUNCTION KEYS - SCREEN SPLITTING
SBDSUB,RBDSUB,S;. TRM.	TERMINAL SUBROUTINES
SBDSUP,RBDSUP,S;. SYS.	SUPERVISOR
SBDSYS,RBDSYS,S;. SPC,PMP.	OPTIONAL SYSTEM
SBDTAB,RBDTAB,S;. TRM.	TAB KEY
SBDTIM,RBDTIM,S;. SYS.	TIMER ROUTINES
SBDTMP,RBDTMP,S;. SYS.	TIMELY ROUTINES
SBDTOP,RBDTOP,S;. SYS.	END OF LOAD MODULE
SBDTPL,RBDTPL,S;. BGD.	8121 IDOS/MFE SUBSTITUTES
SBDTYP,RBDTYP,S;. BGD.	8121 DRIVER
SBDUIO,RBDUIO,S;. SYS.	USER INTERFACE SUBROUTINES
SBDVAR,RBDVAR,R;. SYS.	OVERLAY STUFF
SBDVEC,RBDVEC,R;. SYS.	HYPERSPACE VECTOR TABLE

SYSTEM GENERATION CONTROL FILES

CBDPGN,CBDPGN,C;. CTL,PMP.	PWSGEN PROGRAM
CBDLMF,CBDLMF,C;. CTL,PMP.	LOAD MFE VERSION

DEVELOPMENT CONTROL FILES

CBDASM,CBDASM,G;. CTL.	ASSEMBLE ALL OF PWS
CBDBLD,CBDBLD,G;. CTL.	BUILD ALL PWS ABSOLUTES AND RELOCS
CBDFS1,CBDFS1,G;. CTL.	MICROFICHE BUILD - GENERATE SPOOL FILES

```
   CBDFS2,CBDFS2,G;. CTL.          MICROFICHE BUILD - CREATE TAPES
   CBDMAC,CBDMAC,G;. CTL.          COMPILE ALL MACRO LIBRARIES
   CBDMUT,CBDMUT,G;. CTL.          LOAD UTILITIES
   CBDQLD,CBDQLD,G;. CTL.          QUICK RELOAD SAME CONFIG OF MFE VERSION
   CBDSEQ,CBDSEQ,G;. CTL.          RESEQUENCE ALL SOURCE MODULES
   CBDSPC,CBDSPC,G;. CTL.          ASSEMBLE / COMPILE SPECIAL MODULES
   CBDTAP,CBDTAP,G;. CTL.          MAKE MASTER DISTRIBUTION TAPE
   CBDXRF,CBDXRF,G;. CTL.          LOAD AND GENERATE CROSS REFERENCE

UTILITY PROGRAMS
   SBDCAL,RBDCAL,S;. UTL.          MFE TIME SUBROUTINE FOR COBOL
   SBDDGT,RBDDGT,S;. UTL.          DIRGET UTILITY
   SBDDLS,RBDDLS,S;. SPC.          DIRECTORY LIST UTILITY (COBOL)
   SBDDPT,RBDDPT,S;. UTL.          DIRPUT UTILITY
   SBDDXT,RBDDXT,S;. UTL.          DIRECTORY EXTRACT UTILITY
   SBDFD1,RBDFD1,S;. UTL.          FORMATED DUMP - SAVE FILE ACCESS
   SBDFD2,RBDFD2,S;. UTL.          FORMATED DUMP - DUMP PRINT
   SBDFD3,RBDFD3,S;. UTL.          FORMATED DUMP - DUMP FILE INTERFACE
   SBDFD4,RBDFD4,S;. UTL.          FORMATED DUMP - DUMP FILE ACCESS
   SBDFD5,RBDFD5,S;. UTL.          FORMATED DUMP - CONTROL BLOCK DEFINE
   SBDFD6,RBDFD6,S;. UTL.          FORMATED DUMP - SINGLE RAM WORD DEFINE
   SBDFDA,RBDFDA,S;. SPC.          FORMATED DUMP - MAIN PROGRAM (PASCAL)
   SBDFDB,RBDFDB,S;. SPC.          FORMATED DUMP - PROCEDURES (PASCAL)
   SBDFDC,RBDFDC,S;. SPC.          FORMATED DUMP - PROCEDURES (PASCAL)
   SBDIVL,RBDIVL,S;. UTL.          IVL INTERPRETER FOR PWSGEN
   SBDMAN,RBDMAN,S;. UTL.          PAINT UTILITY
   SBDRPT,RBDRPT,S;. SPC.          STATISTICS PROGRAM (COBOL)
                        END OF PWS DIRECTORY
```

SYNTAX SUMMARY

RESERVED WORDS

The following is a list of reserved words:

ACCEPT
ACCESS
ACTUAL +
ADD
ADVANCING
AFTER
ALARM +
ALL
ALPHABETIC
ALSO
ALTER
ALTERNATE
AND
APOSTROPHE +
APPLY +
ARE
AREA
AREAS
ASCENDING
ASSIGN
AT
AUDIBLE +
AUTHOR

BATCH +
BEFORE
BEGINNING +
BLANK
BLOCK
BOTTOM
BY

CALL
CANCEL
CD
CF
CH
CHARACTER
CHARACTERS
CLOCK-UNITS
CLOSE
COBOL
CODE
CODE-SET
COLLATING
COLUMN
COMMA
COMMUNICATION
COMP
COMPUTATIONAL
COMPUTE
CONFIGURATION
CONTAINS
CONTROL
CONTROLS
COPY
CORR
CORRESPONDING
COUNT
CURRENCY

DATA
DATE
DATE-COMPILED
DATE-WRITTEN
DAY
DE
DEBUG-CONTENTS
DEBUG-ITEM
DEBUG-LINE
DEBUG-NAME
DEBUG-SUB-1
DEBUG-SUB-2
DEBUG-SUB-3
DEBUGGING
DECIMAL-POINT
DECLARATIVES
DELETE
DELIMITED
DELIMITER
DEPENDING
DESCENDING
DESTINATION
DETAIL
DISABLE
DISPLAY
DIVIDE
DIVISION
DOWN
DUPLICATES
DYNAMIC

EGI
ELSE
EMI
ENABLE
END
END-OF-PAGE
ENTER
ENVIRONMENT
EOP
EQUAL
ERROR
ERROR-ITEM +
ERROR-ITEM-1 +
ERROR-ITEM-2 +
ERROR-ITEM-3 +
ESI
EVERY
EXCEPTION
EXIT
EXTEND

FD
FILE
FILE-CONTROL
FILLER
FINAL
FIRST
FOOTING
FOR
FROM

GENERATE
GIVING
GO
GREATER
GROUP

HEADING
HIGH-VALUE
HIGH-VALUES

I-O
I-O-CONTROL
IDENTIFICATION
IF
IN
INDEX
INDEX-SET +
INDEXED
INDICATE
INITIAL
INITIATE
INPUT
INPUT-OUTPUT
INSPECT
INSTALLATION
INTO
INVALID
IS

JUST
JUSTIFIED

KEY
KEY-IN +
KEYBOARD +

LABEL
LAST
LEADING
LEFT
LENGTH
LESS
LIMIT
LIMITS
LINAGE
LINAGE-COUNTER
LINE
LINE-COUNTER
LINES
LINKAGE
LOCK
LOW-VALUE
LOW-VALUES

MEMORY
MERGE
MESSAGE
MODE
MODULES
MOVE
MULTIPLE

MULTIPLY

NATIVE
NEGATIVE
NEXT
NO
NOT
NUMBER
NUMERIC

OBJECT-COMPUTER
OCCURS
OF
OFF
OMITTED
ON
OPEN
OPTIONAL
OR
ORGANIZATION
OUTPUT
OVERFLOW

PAGE
PAGE-COUNTER
PERFORM
PF
PH
PIC
PICTURE
PLUS
POINTER
POS +
POSITION
POSITIVE
PRINTING
PROCEDURE
PROCEDURES
PROCEED
PROGRAM
PROGRAM-ID
PROGRAM-LEVEL +
QUEUE
QUOTE
QUOTES

RANDOM
RD
READ
RECEIVE
RECORD
RECORDS
REDEFINES
REEL
REFERENCES
RELATIVE
RELEASE
REMAINDER
REMOVAL
RENAMES
REPLACING
REPORT
REPORTING
REPORTS
RERUN
RESERVE
RESET
RETURN
REVERSED
REWIND
REWRITE
RF
RH
RIGHT
ROUNDED
RUN

SAME
SCREEN +
SD
SEARCH
SECTION
SECURITY
SEGMENT
SEGMENT-LIMIT
SELECT
SEND
SENTENCE
SEPARATE
SEQUENCE

SEQUENTIAL
SET
SIGN
SIZE
SORT
SORT-MERGE
SOURCE
SOURCE-COMPUTER
SPACE
SPACES
SPECIAL-NAMES
STANDARD
STANDARD-1
START
STATUS
STOP
STRING
SUB-QUEUE-1
SUB-QUEUE-2
SUB-QUEUE-3
SUBTRACT
SUM
SUPPRESS
SYMBOLIC
SYNC
SYNCHRONIZED

TABLE
TALLYING
TAPE
TERMINAL
TERMINATE
TEXT
THAN
THEN +
THROUGH
THRU
TIME
TIMES
TO
TOP
TRAILING
TYPE

UNIT

UNSTRING
UNTIL
UP
UPON
USAGE
USE
USING

VALIDATE +
VALUE
VALUES
VARYING

WHEN
WITH
WORDS
WORKING-STORAGE
WRITE

ZERO
ZEROES
ZEROS
+
-
*
/
**
>
<
=

+ Four-Phase Extension

PICTURE CHARACTERS

X	Any character
A	Alphabetic character or space
9	Numeric character
(n)	The preceding character is repeated n times
S	Operational sign
P	Scale factor
V	Assumed decimal point

EDITING CHARACTERS

0 B . , /	Insertion characters (zero, blank, decimal point, comma, slash)
CR DB -	Sign symbols printed if value minus (space if value positive)
+	Plus or minus sign printed
$	Dollar sign
$ + -	More than one $ + or - specifies floating
Z	Numeric with leading zero suppression
*	Leading zero replaced with asterisk

EXTERNAL NAMES

SD-a	Chained Disc Files (Organization is SEQUENTIAL)
MT-a	Magnetic Tape
CR-a	Card Reader
PR-a	Line or Character Printer
DC-a	Contiguous Disc Files (Organization is RELATIVE)
RF-a	Contiguous Disc Files (Organization is RELATIVE)
DI-a	DISAM Disc File (Organization is INDEXED)
MK-a	MKAM Disc File (Organization is INDEXED)
D4-a	DATA IV/70 Non-indexed Disc File (Organization is BATCH)
BI-a	DATA IV/70 Indexed Disc File (Organization is INDEX-SET)

NOTES:

Syntactically valid program may be either semantically invalid or incorrect.

The various FD clauses may be in any order.

Record description clauses for a single entry may be in any order except for REDEFINES which must be first if present.

NOTATION:

GO TO data-name	Reserved words are capitalized. Required reserved words (keywords) are underlined. Optional reserved words are not underlined. Generic names for user-supplied words are lower-case.
{ choose-1 } { choose-2 }	Vertical stacking within brackets or braces indecates a choice of two or more options one of which is selected.
[optional part]	Brackets indicate that the enclosed words and phrases may b used or omitted.
[repeated]...	An ellipsis (3 periods) indicates that the preceding group may be repeated as many times as necessary.

COBOL LANGUAGE FORMATS

IDENTIFICATION DIVISION Format:

```
IDENTIFICATION DIVISION.
PROGRAM-ID.   program-name.
[AUTHOR.          [comment-entry ... .]]
[INSTALLATION.  [comment-entry ... .]]
[DATE-WRITTEN.  [comment-entry ... .]]
[DATE-COMPILED. [comment-entry ... .]]
[SECURITY.        [comment-entry ... .]]
```

ENVIRONMENT DIVISION Format:

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
                  { FOUR-40       }
                  { FOUR-50       }
SOURCE-COMPUTER.  { FOUR-70       }  [WITH DEBUGGING MODE ].
                  { FOUR-90-MOD2  }

                  { FOUR-40       }
                  { FOUR-50       }
                  ( FOUR-60       )
                  ( FOUR-65       )
                  ( FOUR-70       )
OBJECT-COMPUTER.  { FOUR-90-MOD1  }
                  { FOUR-90-MOD2  }
                  { FOUR-90-MFE   }

  [, SCREEN  SIZE IS lines,linesize
    [, WITH AUDIBLE  ALARM  ]]
    [, MEMORY  SIZE integer  WORDS ]
    [, PROGRAM COLLATING SEQUENCE IS alphabet-name].


 SPECIAL-NAMES.

                                { NATIVE      }
                                { STANDARD-1  }
                                { EBCDIC      }
    [ alphabet-name IS  { UKWTRD      } ]
                                { PRINT-96    }
                                { PRINT-64    }
                                { HONEYWELL   }


    [ CURRENCY SIGN  IS literal ]
    [ DECIMAL-POINT IS COMMA  ]
    [ QUOTE IS APOSTROPHE  ].

[INPUT-OUTPUT SECTION.

FILE-CONTROL. entry.
[I-O-CONTROL. entry.]]
```

FILE CONTROL FORMAT:

Format For SD, MT, PR, And CR Files:

```
       [FILE-CONTROL.
             { SELECT        file-name
               ASSIGN TO     external-file-name ...
             [ ORGANIZATION IS SEQUENTIAL ]
             [ ACCESS MODE  IS SEQUENTIAL ]
             [FILE STATUS  IS file-status].}...]
```

Format For DC And RF Files:

```
       [FILE-CONTROL.
            { SELECT        file-name

              ASSIGN TO  { DC-a }
                         { RF-a }


              ORGANIZATION IS RELATIVE

            [ ACCESS MODE IS { SEQUENTIAL  [RELATIVE KEY IS relative-key]}
                             { RANDOM  RELATIVE KEY IS relative-key      }

             [FILE STATUS IS file-status].}...]
```

Format For DISAM (DI) and MKAM (MK) Files:

```
       [FILE-CONTROL.
            { SELECT        file-name
              ASSIGN TO  { DI-a }
                         { MK-a }
              ORGANIZATION IS INDEXED

                             { SEQUENTIAL }
            [ ACCESS MODE IS { RANDOM     }]
                             { DYNAMIC    }

              RECORD KEY IS  record-key [ WITH DUPLICATES ]
            [{ KEYFIELD IS     } data-name-1 [data-name-2] ...]
             { KEYFIELDS ARE  }
            [[ ALTERNATE RECORD KEY IS alternate-key [WITH DUPLICATES ]]...
            [{ KEYFIELD IS     } data-name-3 [data-name-4] ...]
             { KEYFIELDS ARE  }
            [ INDEX-SET KEY IS index-key] [ SAME KEY AREA ]]
            [ FILE STATUS IS file-status]
            [ PROGRAM-LEVEL IS program-level].}...]
```

Format For DATA IV BATCH (D4) Files:

```
       [FILE-CONTROL.
            { SELECT        file-name
              ASSIGN TO     D4-a ...
              ORGANIZATION IS BATCH
            [ ACCESS MODE IS  SEQUENTIAL ]
              BATCH KEY IS    batch-key
            [ BATCH STATUS IS batch-status]              [ FILE STATUS IS
file-status ].}...]
```

Format For DATA IV INDEX-SET (BI) Files:

```
       [FILE-CONTROL.
            { SELECT           file-name
              ASSIGN TO        BI-a ...
              ORGANIZATION IS   INDEX-SET

            [ ACCESS MODE IS { RANDOM      }
                             { SEQUENTIAL  }

              INDEX-SET KEY IS index-set-key
              RECORD KEY IS record-key
            [ INDEX-SET STATUS IS index-set-status]
            [ FILE STATUS IS     file-status ].}...]
```

I-O-CONTROL Format:

```
      [I-O-CONTROL.
            [ RERUN ON BF-r   EVERY { END OF REEL        } OF file-name-1]...
                                    { integer RECORDS    }
            [ SAME AREA FOR file-name-2 [,file-name-3]... ]...
            [ APPLY keyboard-variables TO KEYBOARD
                [ WITH LENGTH ] [ INVALID KEY ]].]
```

DATA DIVISION format:

```
         DATA DIVISION.
      [ FILE SECTION.
           {file-description-entry.
            record-description-entry.}...]
      [ WORKING-STORAGE SECTION.
           {record-description-entry.}...]
      [ SCREEN SECTION.
           {record-description-entry.}...]
      [ LINKAGE SECTION.
           {record-description-entry.}...]
```

```
FILE DESCRIPTION FORMAT:

       FD  file-name
           [ BLOCK CONTAINS integer-1 { RECORDS     }]
                                      { CHARACTERS }
           [ RECORD CONTAINS [integer-2  TO ] integer-3 CHARACTERS]
             LABEL  { RECORD IS    }{ STANDARD }
                    { RECORDS ARE }{ OMITTED   }
           [ DATA  { RECORD IS    }
                   { RECORDS ARE } data-name ... ]
           [ CODE-SET IS alphabet-name].

       01 record-name record-description-entry.

RECORD DESCRIPTION FORMAT:

General Format 1:

       01 { data-name  [descriptive-clause...] }
          { FILLER }                            .
      [sub-level-no { data-name [descriptive-clause...] }
                    { FILLER }                           .]...

General Format 2:

       77  data-name descriptive-clause... .

Descriptive-Clause Format (File Section):
             [ REDEFINES  data-name]
             [ BLANK WHEN ZERO ]
           [ { JUSTIFIED }
             { JUST      } RIGHT ]
             [ OCCURS  integer TIMES [ INDEXED BY index-variable...] ]
           [ { PICTURE }
             { PIC     } IS  data-picture ]
           [ [ USAGE IS ] DISPLAY ]
            [[ SIGN IS] { LEADING SEPARATE CHARACTER          }
                        { TRAILING  [ SEPARATE CHARACTER ] } ]
           [ { SYNCHRONIZED }  { RIGHT }
             { SYNC         }  { LEFT  }  ]

Descriptive-Clause Format (Working-Storage):

             [ REDEFINES  data-name]
             [ BLANK WHEN ZERO ]
           [ { JUSTIFIED }
             { JUST      } RIGHT ]
             [ OCCURS  integer TIMES [ INDEXED BY index-variable...] ]
           [ { PICTURE }
             { PIC     } IS  data-picture ]
                         { COMPUTATIONAL }
           [ [ USAGE IS ] { COMP          }
                         { DISPLAY       } ]
                         { INDEX         }
            [[ SIGN IS] { LEADING SEPARATE CHARACTER          }
                        { TRAILING  [ SEPARATE CHARACTER ] } ]
           [ { SYNCHRONIZED } [ RIGHT ]
             { SYNC         } [ LEFT  ]  ]
            [ VALUE IS literal ]
```

Descriptive-Clause Format (Screen Section):

```
          [ REDEFINES  data-name]
          [ BLANK WHEN ZERO ]
        [ { JUSTIFIED }  RIGHT ]
          { JUST }
          [ OCCURS  integer TIMES [ INDEXED BY index-variable...] ]
        [ { PICTURE }  IS  data-picture ]
          { PIC     }
        [ [ USAGE IS ] { DISPLAY }  ]
                       { KEY-IN  }
         [[ SIGN IS] { LEADING SEPARATE CHARACTER         }   ]
                     { TRAILING  [ SEPARATE CHARACTER ] }

        [ { POSITION }  IS line-no,column-no ]
          { POS      }
        [ { SYNCHRONIZED } [ RIGHT ]   ]
          { SYNC         } [ LEFT  ]
```

Descriptive-Clause Format (Linkage Section):

```
          [ REDEFINES  data-name]
          [ BLANK WHEN ZERO ]
        [ { JUSTIFIED }  RIGHT ]
          { JUST      }
          [ OCCURS  integer TIMES [ INDEXED BY index-variable...] ]
       [ { PICTURE }  IS  data-picture ]
         { PIC     }
                      { COMPUTATIONAL }
       [ [ USAGE IS ] { COMP          }  ]
                      { DISPLAY       }
                      { INDEX         }
        [[ SIGN IS] { LEADING SEPARATE CHARACTER          }  ]
                    { TRAILING  [ SEPARATE CHARACTER ] }
       [ { SYNCHRONIZED } [ RIGHT ]   ]
         { SYNC         } [ LEFT  ]
```

PROCEDURE DIVISION format:

```
  PROCEDURE DIVISION [USING data-name-1 [data-name-2] ...].
[ DECLARATIVES.
[ section-name-1 SECTION.
       USE AFTER KEYBOARD INPUT.
 [paragraph-name.
        use-after-keyboard-procedure.]...]

[ section-name-2 SECTION.

       USE FOR DEBUGGING ON {procedure-name-1  [procedure-name-2]...}
                            { ALL PROCEDURES                        }.

 [paragraph-name.
       debugging-procedure.]...]

                                                  { file-name-1...}
       USE AFTER STANDARD{ EXCEPTION } PROCEDURE ON { INPUT          }  .
                         { ERROR     }              { OUTPUT         }
                                                  { I-O            }

[paragraph-name.
       error-recovery-procedure.]...]

 END DECLARATIVES.]
 paragraph-name.
       main program
```

PROCEDURE DIVISION STATEMENT FORMATS:

```
        ACCEPT record-name
```

Format 1:

```
        ADD { numeric-data-name-1 }[ numeric-data-name-2 ]
            { numeric-literal-1   }[ numeric-literal-2   ]...
           TO numeric-data-name-m   [ ROUNDED ]
          [,ON SIZE ERROR imperative-statement ]
```

Format 2:

```
        ADD  { numeric-data-name-1 } { numeric-data-name-2 }
             { numeric-literal-1   } { numeric-literal-2   }...
           GIVING  { numeric-data-name-m  }  [ ROUNDED ]
                   { numeric-edited       }
          [ ON SIZE ERROR imperative-statement ]


        ALTER procedure-name-1 TO [PROCEED TO ] procedure-name-2

        CALL  "external-name" [ USING  { parameter-name      }
                                       { entry-point-literal }...]


                            [ { REEL }  [ WITH NO REWIND ]  ]
        CLOSE  { file-name  [ { UNIT }                      ] }...
                            [          { NO REWIND }        ]  .
                            [   WITH   { LOCK      }        ]

        DELETE  file-name RECORD [ INVALID KEY imperative-statement].


                 { data-name            }
        DISPLAY  { figurative-constant  }...
                 { literal              }
```

Format 1:

```
DIVIDE { numeric-data-name-1 } INTO numeric-data-name-2
       { numeric-literal     }
  [ ROUNDED ] [ON SIZE ERROR  imperative-statement]
```

Format 2:

```
DIVIDE {numeric-data-name-1} INTO {numeric-data-name-2}
       {numeric-literal-1  }      {numeric-literal-2  }
    GIVING {numeric-data-name-3} [ ROUNDED ]
           {numeric-edited     }
   [ON SIZE ERROR  imperative-statement]
```

Format 3:

```
DIVIDE {numeric-data-name-2} BY {numeric-data-name-1}
       {numeric-literal-2  }    {numeric-literal-1  }
    GIVING { numeric-data-name-3 }  [ROUNDED]
           { numeric-edited      }
   [ON SIZE ERROR  imperative-statement]
```

```
EXIT [PROGRAM].
```

Format 1:

```
GO TO.
```

Format 2:

```
GO TO  procedure-name-1  [, procedure-name-2 ...
    DEPENDING ON   integer-data-name .
```

```
IF {relational-condition,}  {imperative-statement-1}
   {class-condition      }  { NEXT SENTENCE        }'
  [ ELSE  {imperative-statement-2.} ]
          { NEXT SENTENCE         }
```

Format 1:

```
                                              { ALL character-1      }
INSPECT data-item  TALLYING count  FOR { LEADING character-1 }
                                              { CHARACTERS           }
[ { BEFORE } INITIAL character-2 ]
  { AFTER  }
```

Format 2:

```
                                      {  ALL character-4      }
INSPECT  data-item   REPLACING  {  LEADING character-4 }
                                      {  FIRST character-4    }
                                      {  CHARACTERS           }
    BY character-3 [ { BEFORE } INITIAL character-5 ]
                     { AFTER  }
```

Format 3:

```
                                              { ALL character-1      }
INSPECT data-item  TALLYING count  FOR { LEADING character-1 }
                                              { CHARACTERS           }
[ { BEFORE } INITIAL character-2 ]
  { AFTER  }
                  { ALL character-4      }
    REPLACING     { LEADING character-4 }  BY character-3
                  { FIRST character-4    }
                  { CHARACTERS           }
[ { BEFORE } INITIAL character-5 ]
  { AFTER  }
```

```
KEY-IN  screen-data-name

  [ AND  VALIDATE

  [ length, type, [picture]]]

  [ BEGINNING  AT character-pos ]


MOVE { identifier-1 } TO identifier-2 [, identifier-3]...
     { literal      }
```

Format 1:

```
                { numeric-data-name-1 } _

MULTIPLY        { numeric-literal-1   } BY numeric-data-name-2
  [ ROUNDED ] [ON SIZE ERROR  imperative-statement]
```

Format 2:

```
MULTIPLY { numeric-data-name-1 } BY { numeric-data-name-2 }
         { numeric-literal-1   }    { numeric-literal-2   }
    GIVING { numeric-data-name-3 } [ ROUNDED ]
           { numeric-edited      }
   [ON SIZE ERROR imperative-statement]


       { INPUT  }
OPEN {{ OUTPUT }  file-name[file-name-2]...}...
       { I-O    }


PERFORM  procedure-name-1 [ { THRU    } procedure-name-2 ]
                            { THROUGH }

  [ { integer-data-item } TIMES ]
    { numeric-literal   }
```

Format 1 (sequential access):

```
READ [ REVERSED ] file-name [NEXT] RECORD [ INTO  identifier]

   [AT END  imperative-statement]
```

Format 2 (random access):

```
READ file-name RECORD [ INTO identifier][ KEY IS alternate-key]

   [ INVALID KEY imperative-statement ]


REWRITE  record-name [ FROM identifier ]

  [ INVALID KEY imperative-statement ]
```

Format 1:

```
        SET { index-data-item }...  TO  { index-variable  }
                                        { index-data-item }
```

Format 2:

```
        SET {positive-integer-name}...  TO  index-variable
```

Format 3:

```
                                       { index-data-item            }
         SET { index-variable }...  TO { positive-integer-name      }
                                       { positive-integer-literal   }
                                       { index-variable             }
```

Format 4:

```
                                 { UP BY   }{ integer-name    }
        SET {index-variable}...  { DOWN BY }{ integer-literal }
```

```
        START file-name [ KEY IS data-name ]
```

```
               {literal}
        STOP [ { RUN    } ]
```

Format 1:

```
        SUBTRACT {numeric-data-name-1}[  numeric-data-name-2]...
                 {numeric-literal-1  }[' numeric-literal-2  ]
           FROM  numeric-data-name-m [ ROUNDED ]
          [ ON SIZE ERROR imperative-statement ]
```

Format 2:

```
        SUBTRACT {numeric-data-name-1}[  numeric-data-name-2]...
                 {numeric-literal-1  }[' numeric-literal-2  ]
           FROM {numeric-data-name-m} GIVING {numeric-data-name-n}
                {numeric-literal-m  }        {numeric-edited     }

         [ ROUNDED ] [ON SIZE ERROR  imperative-statement]
```

Format for Printer Files:

```
        WRITE record-name [ FROM identifier]

         [ { BEFORE } ADVANCING { integer LINES } ]
           { AFTER  }           { PAGE          }
```

Format for Tape and Disc Files:

```
        WRITE record-name [ FROM identifier]

           [ INVALID KEY imperative-statement]
```

Format of Relational Condition:

```
                                 { GREATER THAN }
                                 { EQUAL TO     }
         identifier-1 IS [ NOT ] { LESS THAN    } identifier-2
                                 { >            }
                                 { =            }
                                 { <            }
```

Format of Class Condition:

```
        { display-data-item } IS [ NOT ] { NUMERIC    }
        { key-in-data-item  }            { ALPHABETIC }
```

Format of Combined Condition:

```
  [ NOT ] {relational-condition} { { AND } [ NOT ] {relational-condition} }...
          {class-condition     }   { OR  }         {class-condition     }
```

PROGRAM CHANGES REQUIRED TO USE COBOL'74

The changes required to an existing COBOL program originally written with NTP/200 (ANSI COBOL) to compile on COBOL'74 are primarily changes involving files. A list of the Keywords that have been deleted or changed and a list of the required new ones follows this brief summary of the changes. Following the lists of keywords, required and optional changes to each division are described in detail. This is followed by a description of the job control changes involved.

The Keywords deleted include NOMINAL KEY, ACTUAL KEY, FILE LIMIT, PROCESSING MODE. The file type ISAM has been eliminated. The file type, Relative Filled, (RF) has been added. The verb EXAMINE has been replaced by the verb INSPECT.

Keywords now required include ORGANIZATION for all disc files except for non-indexed sequential files. ALTERNATE RECORD KEY is required for DISAM files with secondary keys and DUPLICATES is required if a secondary key has duplicates.

A COMMENT statement must have an * in column 7. No other character will suffice. Statements using REMARKS or NOTE are no longer allowed.

KEYWORDS THAT HAVE BEEN ELIMINATED BY COBOL'74

ACTUAL KEY
AT
BACKWARD
EXAMINE
FILE LIMIT
FOR MULTIPLE REEL
FOR MULTIPLE UNIT
FOUR-90 M1
FOUR-90 M2
IS-a (ISAM files)
NOMINAL KEY
NOTE
NUMBER OF SECONDARY
PROCESSING MODE
REMARKS
SEEK
SUPPRESS
TALLY
THEN
UNIT (in I-O-CONTROL)
UNTIL FIRST

KEYWORDS ADDED BY COBOL'74

ALTERNATE RECORD KEY (was NOMINAL KEY)
BATCH (organization)
BATCH KEY
BATCH STATUS
CHARACTERS
DEBUG-ITEM
DEBUG-LINE
DEBUG-NAME
DEBUG-SUB-1
DEBUG-SUB-2
DEBUG-SUB-3
DEBUG-CONTENTS
DYNAMIC
ERROR-ITEM
EXCEPTION
EXIT PROGRAM
FOUR-90-MOD2 (was FOUR-90 M2)
FOUR-90-MFE
INDEX (organization)
INDEX SET (organization)
INDEX-SET KEY
INDEX-SET STATUS
INSPECT
NATIVE
ORGANIZATION
RELATIVE (organization)
RELATIVE RECORD KEY (was ACTUAL key)
REVERSED
SIGN IS LEADING SEPARATE CHARACTER
SIGN IS TRAILING SEPARATE CHARACTER
STANDARD-1
THROUGH
USE FOR DEBUGGING ON ALL PROCEDURES
WITH DEBUGGING MODE
WITH DUPLICATES

NAME CHANGES

Was for COBOL'68	IDOS COBOL'74	MFE COBOL'74
CBLERR	CBLERR	CBLMER
CBLGEN	C74GEN	C74GEN
COBLIB	C74LIB	C74LB2
COBOL	COBL74	COBL74
DATA IV/70	DATA IV	DATA IV
FLDCOB	CBEFLD	C74FL2
LOADOV	LOADER	LOADER
P:OGTA	PBEOGT	PBEOGT
P:OG7T	PBEOG7	PBEOG7
P:OKAA	PBEOKA	PBEOKA
R:OGTA	RBEOGT	RBEOGT
R92AL1	STLLIB	MFELIB

IDENTIFICATION DIVISION CHANGES

Required changes

REMARKS has been eliminated. The asterisk in column 7 must be used for comment

ENVIRONMENT DIVISION CHANGES

Required Changes

FOUR-90M1 has been eliminated from both the SOURCE-COMPUTER and OBJECT-COMPUTER paragraphs. FOUR-90M2 has been changed to FOUR-90-MOD2.

If an INPUT-OUTPUT section is included in the program, a FILE-CONTROL paragraph must be included even if the program uses no files.
Each file type has been assigned a specific file organization type which must be specified in the File-Control paragraph. Sequentially organized files (SD, MT, PR, and CR) files may be specified implicitly. All other types must be explicitly specified.

FILE STATUS is now only two characters long. In addition, it is no longer used for setting and resetting Batch Directory flags for DATA IV files.

RF Files: A new file type has been added that is compatible with the COBOL'68 DC files. COBOL'68 DC files are not compatible with COBOL'74 DC files. To use a COBOL'68 DC file with a COBOL'74 program, the file must be converted into an RF (Relative Filled) file by changing the IDOS directory entry. FOREWORD TXAREA files may be used as RF files without conversion or change. RF files are defined as ORGANIZATION IS RELATIVE.

COBOL'74 DC Files: ORGANIZATION IS RELATIVE is a new requirement for DC files. RELATIVE KEY must be specified instead of ACTUAL KEY for DC files in random access mode. COBOL'74 DC files contain area assigned by the system to flag the presence of records in record slots. This space amounts to one word of storage for each 24 records in the file. Because COBOL'68 DC files do not have this space, they are incompatible with COBOL'74 DC files.

DISAM Files: ORGANIZATION IS INDEXED is a new requirement for DISAM files. RECORD KEY rather than NOMINAL KEY is used to access DISAM files by primary key. ALTERNATE RECORD KEY is used to access DISAM files by alternate keys that are unique. ALTERNATE RECORD KEY WITH DUPLICATES is required for access by alternate keys that are not unique. KEY IS <u>record-key</u> is used in place of NOMINAL KEY with the START verb in sequential access to change the sequence. INDEX-SET KEY is used with alternate key with duplicates to specify a record other than the first record with the same key.

ISAM Files: ISAM (IS-a) has been eliminated as a file type.

DATA IV Batch (D4) Files: ORGANIZATION IS BATCH is a new requirement. BATCH KEY is required in place of NOMINAL KEY to specify the job, batch and program level when opening a D4 batch. It is required to specify the program level when closing a batch. BATCH STATUS takes the place of FILE STATUS for the purpose of setting and resetting batch directory flags in the BAM interface.

DATA IV Index-Set (BI) Files: ORGANIZATION IS INDEX-SET is a new requirement. INDEX-SET KEY is required in place of NOMINAL KEY to specify the index set and program level when opening a BI batch. It is required to specify the program level when closing a batch. INDEX-SET STATUS takes the place of FILE STATUS for the purpose of setting and resetting batch directory flags in the BAM interface.

I-O-CONTROL: The option "UNIT" has been deleted.

Optional Changes

In the OBJECT-COMPUTER paragraph, a program collating sequence clause has been added. This has no effect on FOUR-Phase COBOL programs since only the NATIVE collating sequence (ASCII) is used.

FOUR-90-MFE has been added as an OBJECT-COMPUTER selection.

The pound sterling has been added as choice of literal under SPECIAL-NAMES. WITH DEBUGGING MODE has been added. It is a compile time switch that tells the compiler whether or not to compile source code identified by a 'D" in column 7 or source code in the USE FOR DEBUGGING Section of the Declaratives. If this clause is omitted, these statements are treated as comments by the compiler. Debugging is explained in the discussion of Declaratives in Section 4.

DYNAMIC mode for DISAM and MKAM files allows both random and sequential access

DATA DIVISION CHANGES

Required Changes

Except for Division headers, section names, paragraph names, Declaratives header and End Declaratives, FD level indicator, and 01 and 77 level numbers, no other items are allowed in the A margin. Previously, other level numbers and statements following an FD keyword or level number on the same line were allowed.

The target of a REDEFINES must be the originally defined field. (Prior releas allowed the target field to be a field with a REDEFINES attribute.

FILE STATUS must be defined as a two-character item rather than a two-word item in Working-Storage.

DC Files: The data name given in the RELATIVE KEY clause in the Environment Division must be defined in Working-Storage as an unsigned unscaled integer. ACTUAL KEY was previously used.

The DC file structure is changed from prior releases of COBOL. One word of d is added to a block for each 24 records or fraction of 24 records in the block. If disc sectors are tightly packed, these overhead bytes may cause th size of the file to expand dramatically. Two solutions are: change the record length or blocking factor. Unload the old DC file using a previous release of COBOL and reload using release BE01 or subsequent.

DISAM Files: The data name given in the INDEX-SET KEY clause in the Environment Division must be defined in Working-Storage as a group item consisting of two Computational items. NOMINAL KEY was previously used for this function.

DATA IV D4 Files: The data name given in the BATCH KEY clause in the Environment Division must be defined in Working-Storage as a group item 18 bytes long to specify the job name, batch name, and program level. NOMINAL KEY was previously used fpr this function.

DATA IV BI Files: The data name given in the INDEX-SET KEY clause in the Environment Division must be defined in Working-Storage as a group item 6 bytes long to specify the index set number and program level. NOMINAL KEY was previously used for this function.

Optional Changes

CODE-SET has been added. It specifies the character set to be used to represent data on a non-mass-storage media such as a printer. The six options are: NATIVE, STANDARD-1, EBCDIC, PRINT-96, PRINT-64, UKWTRD, HONEYWELL.

File Section: The record description has a SIGN clause to allow further specification of numbers specified as signed numeric in the PICTURE clause. SIGN clause allows specifying the operational sign as a leading or trailing separate character. By default, it is imbedded. This clause, when used at the group level, specifies the sign location of each signed numeric item within the group. If SEPARATE is specified, a storage location is allocated for the sign. Code conversion as needed is handled automatically when a number with a separate character position for the sign is used in computations.

The stroke character (/) has been added as an editing character.

The "B" character is now allowed in the picture character-string of an alphabetic item.

The pound-sterling character is now allowed as either a fixed or floating data-picture character.

CHANGES IN PROCEDURE DIVISION

Required Changes

The Declaratives part must begin with a paragraph name. Likewise, the main program must also start with a paragraph name. These paragraph names are required even though they are never referenced.

The word NOT is no longer permitted to precede the first identifier in a relation condition or class condition.

The EXAMINE statement has been changed to an INSPECT statement.

The storage area TALLY no longer exists.

The START statement has been changed replacing the keywords USING KEY with the keywords KEY IS.

READ BACKWARD has been changed to READ REVERSED (DISAM).

COPY statement rules have been liberalized. COPY restrictions have been removed. SUPPRESS and AT drive-n have been deleted from the COPY statement. Member has also been deleted. COPY library (member) has been changed to COPY library.

Opening a DC file for OUTPUT clears the file. DC files are now sensitive to the presence or absence of a record in a slot.

The NOTE statement has been deleted.

The word THEN is no longer supported in IF statements.

IF statements may now be nested.

Optional Changes

USE FOR DEBUGGING has been added to the Declaratives part providing a convenient tool for debugging. User-written procedures in this section are executed whenever a specified procedure is about to be executed. Thus, the number of times a given procedure is executed can be tabulated. DEBUG-ITEM and ERROR-ITEM are new special registers that can be examined for further analysis during debugging. ERROR-ITEM contains the 3-byte code formerly in the second word of the file status buffer.

The USE AFTER STANDARD ERROR has been changed to allow the use of EXCEPTION in place of ERROR. It also allows the use of INPUT, OUTPUT, and I-O in place to filename.

The EXIT PROGRAM option has been added to the EXIT statement.

The word THROUGH has been added as an alternative to the word THRU

The OPEN statement has been changed to allow more than one file to be opened with the same OPEN statement.

An optional MOVE INTO statement has been incorporated into the READ statement.

An optional MOVE FROM statement has been incorporated into the WRITE and REWRITE statements.

The keyword NEXT has been added to the READ statement and is required when retrieving DISAM records sequentially from a file that has been declared DYNAMIC access.

PAGE has been added as an option to the WRITE statement to advance the printer to the next page.

JOB CONTROL LANGUAGE CHANGES

LOADOV cannot be used. LOADER must be used instead.

The IDOS release AD-32 must be used with COBOL'74. The IDOS release AD-32 utilities (UGEN) requires about 5000 sectors. If a 8230 disc is to be used, don't use UGEN. Instead, use COPY to copy only the utilities needed.
All COBOL programs must be recompiled to run under BE02.

COBOL now runs under MFE if desired. To run under MFE, a COBOL program must be changed and recompiled to specify FOUR-90-MFE as the object computer and to interact with MFE to sign on and sign off terminals. The load step must be changed to specify C74LB2 and MFELIB rather than C74LIB and STLLIB. Also, RDS-SY, MFETOP, and SETUP must be specified.

Until MIDOS is updated to the IDOS AD32 level, it cannot be used with COBOL'74.

REFERENCES

COBOL'74 Language Definition Manual, Document SIV/70-45-10.

COBOL'74 Programmer's Guide, Document SIV/70-45-9.

COBOL COMPILER TEMP FILES

Compiler passes:
1. Listing/Syntax Analysis
2. Symbol Table Build
3. Storage Generation
4. Attribute Merge
5. Binary Tree
6. Data Division Map
7. Code Generation
8. Final Assembly
9. Cross Reference
10. Error Report

The following TEMP:x files are used by the compiler, where x is the file letter in the first column. The second column describes the file usage and lists what pass creates it (e.g. out 2) and what passes read it (e.g. in 3).

G	Production Numbers, Procedure Division out 1; in 2, 4, 5, 9.
P	Production Numbers, Other Divisions out 1; in 2, 3.
C	Internal Source out 1; in 2.
S	Constant String out 2; in 3.
T	Attribute File out 2; in 3.
A	Symbol Table Indices out 2; in 3.
B	Attribute Pointers out 2; in 4, 6, 9.
E	Edited Pictures out 2; in 3.
F	Data Item Attributes out 3; in 4, 6, 9.
W	Procedure Division Literals out 2; in 7.
D,Z	01 Text - (Changes to Z) out 3; in 8.
M	Data Items for Procedure Division out 2; in 4, 9.
H	Merged Attributes out 4; in 5.
J	Binary Tree File out 5; in 7.
L	Ltext out 7; in 8.
D	02 Text - (Second Value for D) out 7; in 8.
N	Error file out 1, 2, 3, 4, 5, 7; in 10.

ERROR FILE FORMAT

TEMP:N is the file which all passes log errors to, and from which the error printout is formatted. The format of each error record is:

Field	Length	Note
Error Number	4	Documented in Programmer's Guide, Appendix A
Line number	6	Source program line number
Severity	1	0 to 4 for Warning, Syntax, Error, Fatal, or Recovery respectively
Pass	1	Compiler Pass number which logged error
Other info	30	Fields needed to fill in error message
Zero	3	

ABEDMP is a debug utility which may be used to format some of the compiler TEMP files to the printer. The format is:

```
    // ABEDMP
    /x              where x is the file letter
    //
```

PASCAL ERROR MESSAGES

The compiler, executing as a PASCAL program, may take a PASCAL error under unexpected circumstances. The format of the error message is:

```
              PASCAL OBJ LIB ERROR xxx
                   where xxx equals:
```

DSC	File or disc error from IDOS routines $FOPEN or RSCR (May indicate no more room on disc)
EOF	Attempted to read past end of file (Compiler may be looking for required syntax in source file)
OVD	Disc error during overlay manipulation from PASCAL $OVRLY
OVR	Dynamic storage area of memory overwritten by a PASCAL overlay read
RAM	No memory left in heap storage for sector buffer, record buffer, or a new procedure
REC	No memory left in stack for recursive procedure call
TAP	TAPE7, 8, or 16 was used for a file name. PASCAL does not provide tape support.

COMPILER TABLE SIZES

The following describes the current table sizes used by the COBOL compilers. With BE03-A and later the table sizes under the MFE compiler were expanded. The table type is listed on the left followed by the size for each type of the three compilers and the name of the PASCAL source modules which specify table sizes.

	COMPILER SOURCE MODULE AND SIZE FOR					
TABLE	MFE		96K		72K	
SYMBOL TABLE	SBE2TC SBE4TC SBE6BA SBE9TC	1500	SBE2TA SBE4TA SBE6AA SBE9TA	1400	SBE2TB SBE4TB SBE6AA SBE9TB	850
ATTRIBUTE TABLE	SBE4TC SBE9TC	915	SBE4TA SBE9TA	850	SBE4TB SBE9TB	500
NODE TABLE	SBE5BA SBE7TC	125	SBE5AA SBE7TA	100	SBE5AA SBE7TB	100
LITERAL POOL WORD LIMIT	SBE7TC	7500	SBE7TA	4000	SBE7TB	1200
LITERAL POOL CHARACTER LIMIT	SBE7TC	22500	SBE7TA	12000	SBE7TB	3600

LAM / 8437

Release BF03

LAM Internal Errors ($LOGIC)

The following error codes are returned to the application in $LOGIC if an internal error occured during the execution of the last LAM routine.

$LOGIC	Routine	Error
1	INTLNE	An attempt to initialize LAM was made before LAM was terminated (via a TRMLNE call).
2	INTLNE	SYSGN parameters passed to LAM do not agree with the load module in the 8437.
3	OPNLNE	LAM was not initialized (INTLNE) before the open call.
4	PUTLNE	LAM was not opened to transmit before the write call.
5	GETLNE	LAM was not opened to receive before the read call.
6	INTLNE	SYSGN parameters passed to LAM specified an invalid block size. (127 < Block Size < 513)
7	OPNLNE	LAM was opened before the open call.
8	GETLNE	A read request was received in the 8437 before the previous request completed.
9	PUTLNE	A write request was received in the 8437 before the previous request completed.
10		LAM internal error.
11		LAM internal error.
12		8437 received an invalid request table.
13		Not used.
14	CLSLNE	A call to close was made while receiving data. The application should call ABTLNE to abort receiving.
15	ABTLNE	A call to abort was made before LAM was initialized.
16	TRMLNE	A call to terminate was made before LAM was initialized.
17	OPNLNE	An invalid device type was specified while opening LAM to transmit.
18	HNGLNE	A call to hang the line was made before LAM was aborted.
19	TRMLNE	A call to terminate LAM was made before the HNGLNE call completed.
20	HNGLNE	A call to hang the line was made before LAM was initialized.
21	OPNLNE	A call to open the line was made before the abort call completed.
22		LAM internal error.
23	PUTLNE	No characters in the record written to LAM.
24	RNGLNE	A call to send a BEL was made while not in a point-to-point, terminal-to-terminal configuration.

LAM Status Indicators

The status indicators are updated at level 1 by the 8437-resident software every time a request or response table is sent to the IV/xx and when the software performs a status update. The indicators contain 1 when true, 0 when false. These are by no means real-time indicators. For example, $LNCRR will indicate that carrier is up the entire time while receiving because it was up when the last block was received and the transmit logic does not sense the carrier state, as it has no relevance to transmitting.

Indicator	Meaning
LNELAM	LNELAM error indicator (see below).
$LNBEL	A BEL sequence was received.
$LNCHK	A NAK was transmitted or received in response to a block.
$LNDSR	Data Set Ready is true.
$LNADR	A line bid or select was received.
$LNDIB	There is data in the 8437 buffers.
$LNRVI	An RVI was received.
$LNBID	LAM is sending line bids (never true on multipoint systems).
$LNWAK	LAM is WACKing text blocks.
$LNHNG	LAM received a DLE-EOT.
$LNRTY	A prior LAM call has not completed.
$LNNLA	Line idle for 20 seconds.
$INDEX	Indicates the current operation of the 8437 (see below).
$LOGIC	Indicates details of logic errors (see above).

$INDEX	Meaning
0	Read line bid.
1	Write an ENQ.
2	Receive ENQ error.
3	Waiting for Data Set Ready.
4	Write ACK to line bid or text.
5	Read text.
6	Write text.
7	Read response to a text block.
8	Write NAK.
9	Read response to a line bid.
10	Write an EOT then read line bid.
11	Write an EOT the write a line bid.
12	Write a WACK.
13	Write a TTD.
14	Send a DLE-EOT then retry the connection.
15	Line bid received.
16	Write a line bid.
17	Send a BEL then read line bid.
18	Send an RVI.
19	Dial a number.

LNELAM Indicator values

Value	Meaning
1	LAM sent NAKs to a text block and then received an EOT.
2	Data Set Ready dropped.
3	LAM received an EOT in response to a text block.
4	No response to line bids.
5	No response to text blocks sent so LAM sent an EOT.
6	LAM received the wrong ACK 15 times, then sent an EOT.
7	The application passed LAM a record with transparent data when not in transparent mode.
8	LAM received an EOT following an ETB block.
9	LAM received a block larger than 513 characters.
10	LAM received NAKs to line bids 15 times.
11	LAM received NAKs to text then received an EOT.
12	No Sync characters were received in three seconds.
13	LAM received NAKs to text 15 times, then sent an EOT.
14	Not used.
15	No line activity for 20 seconds.
16	Contention for the line on a secondary terminal.

Value	Description
17	Record overflow during expansion. Record exceeded 135 bytes.
18	More than 513 characters received without an ETB/ETX and without any Sync characters in 3 seconds.
19	More than 513 characters received with an ETB/ETX.
20	Data received for an unavailable device and LAM sent an EOT.
21	LAM received data while opened to transmit under Switched Network Protocol.
22	LAM received an unrecognizable device code.
23	Unused.
24	LAM received an ENQ-EOT sequence.
25	Dynamic terminal reconfiguration requested and the other station wants the line. LAM remaines secondary until the next INTLNE call.
26	Dynamic terminal reconfiguration requested and the other station wants the line. LAM switched from primary to secondary until the next INTLNE call.
27	Dynamic terminal reconfiguration requested and the other station acknowledged LAM's line bid. LAM switched from secondary to primary until the next INTLNE call.
28	Dynamic terminal reconfiguration requested and the other station acknowledged LAM's line bid. LAM remains primary until the next INTLNE call.

Request Table Format

The following is the layout of the request and response tables transfered between the IV/xx and the 8437.

	0 1 2 3 4 5 6 7	8 9 1 0 1 2 3 4 5	6 7 8 9 2 0 1 2 3
+0	CMP IOC	LAM Status Indicators	
+1	$INDEX	Unused	Logical 8437
+2	Command	Modifier	Data Desc.
+3	Device Dest.	Transparency	Terminator
+4	Request Table Address (used by $XFER)		
+3	Error Type	<- Post Processing Routine ->	Error Code
+6	Associated Buffer Address		

Word	Bits	Definition
0	0	Completion bit. 1 = completion table.
	2	I/O complete bit. 1 = table transfered to the 8437.
	8-23	LAM Status indicators (in tables from the 8437 only). The bits are as follows: 8 = $LNDSR 11 = $LNDIB 15 = $LNRVI 18 = $LNBEL 9 = $LNCRR 12 = $LNWAK 16 = $LNADR 10 = $LNCHK 13 = $LNBID 17 = $LNHNG
1	0-7	$INDEX (in tables from the 8437 only).
	16-23	Logical 8437 number.

Word	Byte	Description
2	0-7	Command: 0 - Write 1 - Read (the Command Modifier is not used for Read commands) 2 - Control 3 - Status
	8-15	Command Modifier: For Command = 0 (Write): 0 - Write one record. 1 - Flush all buffers. 2 - Write one record then flush all buffers. 011 - Write ACU table for DIALNE. 0200 - Write initialization table. For Command = 2 (Control): 0 - Open for transmit. 1 - Open for receive. 2 - Close. 3 - Abort. 4 - Hang (disconnect). 5 - Terminate - retain the line. 6 - Terminate - drop the line. 7 - Transmit a BEL sequence. For Command = 3 (Status): 0 - Device update request. 1 - Error report. 2 - Terminal ID report.
	16-23	Data Descriptor: For Command = 0, Modifier = 0, 1 or 2 (Write): 0 - No compression, no conversion to EBCDIC 1 - No compression, with conversion to EBCDIC 2 - Blank compression, with conversion to EBCDIC 3 - Full compression, with conversion to EBCDIC For Command = 0, Modifier = 011 (Write DIALNE ACU): 1 - Status: dial in progress. For Command = 1 on completions from the 8437 (Read): The record terminator (carriage control) character. For Command = 2, Modifier = 0 (Open for transmit): 0 - Not transparent 1 - Transparent For Command = 2, Modifier = 1 (Open for receive): 0 - Received records will be decompressed. 1 - Received records will remain compressed.
3	0-7	Device Destination: For Command = 2, Modifier = 0 (Open for transmit): 0 - Transmit punch data. 1 - Transmit print data. For Command = 1 on completions from the 8437 (Read): 0 - Punch record received. 1 - Print record received. 0377 - End of file received.
	8-15	Transparency Conversion (for Open to transmit or receive): 0 - Do not convert transparent data. 1 - Convert received transparent data to ASCII. Convert transmitted transparent data to EBCDIC.
	16-23	Length of the received record on completions from the 8437, Command = 1 (Read). Contains the truncated length of the record received. Compressed trailing blanks are included in this count.

Word	Byte	Description
4	0-23	Reserved for $XFER
5	0-23	Address of the post-processing routine (Command = 0, 1 or 2).
	0-7	Contains the error type on requests from the 8437 only, Command = 3. Zero indicates no error.
	16-23	Contains the specific error code on requests from the 8437 only, Command = 3.
6	0-23	Address of the associated buffer as follows: The transmit record buffer (Command = 0, Modifier = 0 or 2). The ACU table (Command = 0, Modifier = 011). The initialization table (Command = 0, Modifier = 0200). The receive record buffer (Command = 1).

LAM SYSGN

The following is a summary of the LAM SYSGN questions and there effect on the operation of LAM.

- 2780 Terminal ?
  - Y = 2780.
  - N = 3780.
- Compression Type ?
  - 1 = No compression. Truncation for 3780 or 2780 with Auto EM (below).
  - 2 = Blank compression. Uses IBM 3780 GS sequences to compress 2 or more duplicate blanks.
  - 3 = Full compression. Uses Four-Phase SUB sequences to compress 3 or more duplicate characters.
- Auto-EM Insertion (2780 only) ?
  - Y = EM characters are inserted into records with trailing blanks.
  - N = No EM characters are inserted. (Must be N if compression ¬ 1).
- Point-to-point ?
  - Y = Point-to-point between only 2 terminals or 1 CPU and 1 terminal.
  - N = Multipoint. The host has several terminals on the same line.
- Inquiry Mode (3780 Multipoint only) ?
  - Y = 3780 inquiry will be used in transmit operations.
  - N = No inquiry will be used.
- Extended Line Bid Retry ?
  - Y = LAM will resend a line bid forever when starting a transmission.
  - N = LAM will resend line bids 40 times when starting a transmission.
- Terminal to CPU ?
  - Y = LAM will put no device selection on transmitted data and is the primary station.
  - N = LAM will put device selection on all transmitted data.
- Primary Station (terminal-to-terminal only) ?
  - Y = LAM will bid every 1 second for the line until it is acknowledged.
  - N = LAM will bid every 3 seconds for the line and will relinquish the line if the other station bids.
- Switched Line (point-to-point only) ?
  - Y = LAM will transmit a DLE-EOT when the application calls HNGLNE.
  - N = LAM will never transmit a DLE-EOT.
- Terminal ID ?
  - Y = LAM will transmit the terminal ID on point-to-point lines. LAM will recognize its terminal ID on a multipoint line.
  - N = LAM has no terminal ID.
- Manual Answer Modem (switched line only) ?
  - Y = LAM will bring up DTR immediatly after initialization.
  - N = LAM will bring up DTR in response to DSR or Ring Indicator.

Standard Blocksize ?
Y = LAM will transmit 512-byte blocks for 3780, 400-byte blocks for 2780. LAM will receive up to 513-byte blocks.
N = LAM will transmit blocks the specified size. LAM will receive up to 513-byte blocks.

EBCDIC ?
Y = LAM will use EBCDIC line-control characters.
N = LAM will use ASCII line-control characters.

Translate Transparent Text ?
Y = LAM will translate received transparent text to ASCII.
LAM will translate transmitted transparent text to EBCDIC.
N = LAM will not translate transmitted or received transparent text.

Standard Records/Block (Normal Data) ?
Y = Transmit up to 7 records/block (2780) or 256 records/block (3780).
N = LAM will transmit the specified number of records per block.

Standart Records/Block (Transparent Data) ?
Y = Transmit 1 record per block.
N = LAM will transmit the specified number of records per block.

Immediate Wack Option (3780 Multipoint only) ?
Y = LAM will respond to selects with WACK if not opened to receive.
N = LAM will delay before sending a WACK if not opened to receive.

Switched Network Protocol (Auto answer only) ?
Y = LAM gives the host control of the line when it answers the phone.
N = LAM does not wait for the host to establish the line.

Expand Horizontal Tabs to Spaces ?
Y = HT characters received are expanded to the number of spaces specified by the previously received Horizontal Tab buffer.
N = HT characters are passed to the application in the data record.

LAM 8437 Dump Analyzer

The following is a summary of the JCL which is entered into LAMDMP. LAMDMP can interpret either the contents of the 8437 RAM or an IDOS file created by DTCOMM. The default values are logical controller 0 and the EBCDIC character set.

```
// LAMDMP                  To start execution of the dump analyzer.
/ INPUT = file @ drive.    Analyze 8437 dump file on the disc (overrides /L).
  or
/ LOGICAL = controller.    Analyze 8437 RAM in the Logical 8437 specified.
/ CODE = ASCII/EBCDIC.     The dump is to be interpreted as ASCII or EBCDIC.
//
```

LAM Calling Sequences

```
Initialization
    BRM     INTLNE
     MZE    Common Area       Pointer to the LAM Common Area.
     BRA    Retry
     BRA    Error Return      Error code in $LOGIC
    ---     Normal Return
 Open
    BRM     OPNLNE
     PZE    Flag Address      Flag contents: 0 = receive, 1 = transmit
     BRA    Retry Return
     BRA    Error Return      Error code in $LOGIC
    ---     Normal Return
```

Close

```
    BRM     CLSLNE
     PZE    Value Address   Value contents: 0 = close, ¬0 = open to receive.
     BRA    Retry Return
     BRA    Error Return    Error code in $LOGIC
    ---     Normal Return
```

Read

```
    BRM     GETLNE
     PZE    Record Buffer   The record buffer is a 45-word area.
     BRA    Error Return    Error code in $LOGIC
     BRA    Overflow        An Overflow record was received by LAM
     BRA    End-of-File     LAM received an End of File and closed.
```

Write

```
    BRM     PUTLNE
     PZE    Record Buffer   The record buffer is a 45-word area.
    or
     MZE    Record Buffer   Transmit the record then flush all buffers.
    or
     DCN    0               Flush all buffers immediately.
     BRA    Retry Return
     BRA    Error Return    Error code in $LOGIC
    ---     Normal Return
```

Abort

```
    BRM     ABTLNE
     BRA    Reject Return   $LNDIB is true and a GETLNE call is outstanding.
     BRA    Retry Return
     BRA    Error Return    Error code in $LOGIC
    ---     Normal Return
```

Hang up the line

```
    BRM     HNGLNE
     BRA    Retry Return
     BRA    Error Return    Error code in $LOGIC
    ---     Normal Return
```

Terminate the line

```
    BRM     TRMLNE
     DCN    Soft terminate  If the location is non 0, soft terminate the line.
     BRA    Error Return    Error code in $LOGIC
    ---     Normal Return
```

Update the Device Status

```
    BRM     DEVLNE
     PZE    Status words    Pointer to 2 status words.  The first is for the
                            punch, the second for the printer.  1 = device is
                            ready, 0 = device is not ready.
    ---     Normal Return
```

Transmit a BELL Sequence

```
    BRM     RNGLNE          Only works on terminal-to-terminal configurations.
     BRA    Retry Return
     BRA    Error Return    Error code in $LOGIC
    ---     Normal Return
```

MLAM / 8437

Release BG03

HASP Workstation (MODEL 20) Protocol Notes

See section 'AT' of the SE Handbook for further protocol notes. Section AT also has information on the 8436 version of MLAM; this section (BG) contains information on the 8437 version of MLAM. Protocol conventions for Model 20 are established by each individual vendor of HASP Workstation software rather than by a "Component Description" such as is done for 3270, 3770 or 3780. Thus it is sometimes possible for one vendor or another to slip something thru that eventually become a challenge to other vendors to cope with. In particular, many host systems will ignore device Wait-A-BiTS (such as for PTR 1: 87-CF) but not Full System Wait-A-BiTS (for all devices: CF-CF), and therefore it becomes necessary for the SE to inform the offending vendor that they are in error. The best documentation Four-Phase has on this subject is the MLAM PROGRAMMERS GUIDE (SIV/70-53-2A). A Host User may refer to the HOUSTON AUTOMATIC SPOOLING AND PRIORITY SYSTEM - VERSION 3 manual that IBM publishes (S/360D-05.1.014) or the appropriate Internal Logic manual for the Comm spooler in use (JES, RSCS, RES, etc.).

External Interface Codes

MLMERR error codes (Note: values are decimal!)

1	Invalid record type given with SNDLNE
2	Invalid record type given with SNDEOJ
3	$XCLOS reject exit ($XCLOS error code in RA)
4	$XLOG reject exit ($XLOG error code in RA)
5	SNDEOJ called before initializing with INTM20
6	SNDLNE called before initializing with INTM20
7	RCVLNE called before initializing with INTM20
8	Compression Count or Type invalid (Count/type in RA)
9	INTM20 called twice
10	TRMM20 called before initializing with INTM20
11	HNGM20 called before initializing with INTM20
12	Called SNDEOJ while still doing SNDLNE retries
13	Called SNDLNE while still doing SNDEOJ retries
14	?CONSZ is greater than LCB console message size
15	$XOPEN reject exit ($XOPEN error code in RA)
16	$XFER reject exit ($XFER error code in RA)
17	?MXBSZ buffer size is outside range of 150 to 512
18	No response from 8437 for 5 seconds to INTM20 call
19	No response from 8437 for 5 seconds to HNGM20 call
20	No response from 8437 for 5 seconds to SFTM20 call

$INDEX Codes indicating type of line activity (Note: values are decimal!)

0	Awaiting ENQ on idle line
1	Awaiting ENQ or ACK on idle line
2	Reading ACK/NAK/Text Data
3	Write Text Data
4	Write ACK
5	Write NAK
6	Write ENQ
7	Write "Request Permission to Send to Device" message
8	Write "Permission Granted to Send to Device" message
9	Write "Lost Text" message
10	Write "WABT" message
11	Write "Cancel WABT" message

$STATS Codes indicating error conditions, if any (Note: values are decimal!)

0	No abnormal condition
1	Transparency Check: transparent data detected on receive in nontransparent data block.
2	Data Set Ready dropped ($LNDSR went zero)
3	Lost Data on Send (Lost Text Message received)
4	Lost Data on Receive (Lost Text Message received)
5	Oversized message block received (greater than configured buffer size)
6	Not currently used
7	Not currently used
8	Modem CTS signal not responding to RTS
9	Unknown SCB in Text (garbaged data block)
10	Unknown RCB in Text (garbaged data block)
11	ENQ received -- line possibly restarted

$ Indicators Used by MLAM/8437 (Zero means indicator off)

$LNDSR	Data Set Ready -- modem is ready for 8437 to speak to it
$LNCRR	Carrier received (CD high)
$LNPMP	Waiting for Permission to send from Device)
$LNWBT	WABT received, we cannot send
$LNNLA	No line activity
$LNDIB	Data in input buffer (a record is available to receive)
$LNCHK	NAK sent or received
$LNIDL	Idle ACKing on line
$LNDTR	Data Terminal Ready -- 8437 ready to speak to modem

Request Table Format

The following is the layout of the request and response tables transfered between the IV/xx and the 8437.

				1	2
	0 1	2	3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3
+0	CMP	IOC		$INDEX	$STATS
+1	Available Record Bits				Logical 8437
+2	Command			Modifier	Stream ID.
+3	$LNxxx Indicator Mask				Record Length
+4	Request Table Address (used by $XFER)				
+3	Post Processing Routine Address				
+6	Associated Buffer Address				

Word	Bits	Description
0	0	Completion bit. 1 = completion table.
	2	I/O complete bit. 1 = table transfered to the 8437.
	8-15	$INDEX. Set on completions from the 8437.
	16-23	$STATS. Set on completions from the 8437.

Word	Bits	Description
1	0-15	Available Record Bits (requests and completions from 8437 only).
	0	Console message bit. 1 = console message available.
	1-7	Reserved.
	8	Printer 1 record bit. 1 = record available for printer 1.
	9	Printer 2 record bit. 1 = record available for printer 2.
	10-13	Reserved
	14	Punch 2 record bit. 1 = record available for punch 2.
	15	Punch 1 record bit. 1 = record available for punch 1.
	16-23	Logical 8437 number.
2	0-7	Command. 0 = Control 1 = Read (from the 8437 to the IV/xx) 2 = Write (from the IV/xx to the 8437) 3 = Status
	8	Convert. 1 = Convert transparent transmissions to EBCDIC.
	9	Transparent. 1 = Transparent in transparency.
	10-11	Reserved.
	12-15	Command Modifier. 0 = Initialization. 1 = Status request. 2 = SNDLNE. Transmit a record. 3 = SFTM20. Soft terminate. 4 = SNDLNE. Flush transmit buffers. 5 = HNGM20. Hang up the phone. 6 = TRMM20. Terminate communications. 7 = SNDLNE. Send end of job. 8 = RCVLNE. Receive a record. 9 = DIALNE. Dial the phone.
	8-15	Carriage control character (on receive completion).
	16-23	Acceptable record mask (on receive request).
	16	Printer 1 bit. 1 = can receive to printer 1.
	17	Printer 2 bit. 1 = can receive to printer 2.
	18-21	Reserved.
	22	Punch 2 bit. 1 = can receive to punch 2.
	23	Punch 1 bit. 1 = can receive to punch 1.
	16-19	Stream identifier (0-7, on transmit request, receive completion).
	20-23	Record type (on transmit request, receive completion). 0 = 80 byte record from the card reader. 1 = 80 byte record to the punch. 2 = 133 byte record to the printer. 3 = console message. 4 = Signon, uncompressed. 5 = Signoff.
3	0-8	$LNxxx Indicators (on completions from the 8437 only). 0 = $LNDSR 3 = $LNWBT 6 = $LNCHK 1 = $LNCRR 4 = $LNNLA 7 = $LNIDL 2 = $LNPMP 5 = $LNDIB 8 = $LNDTR
	9-15	Reserved.
	16-23	Record length (on read completions).

Word	Bits	Description
4	0-23	Address of the request table (filled in by $XFER).
5	0-23	Address of the post processing routine.
6	0-23	Address of associated data buffer.

Initialization Table Layout

Byte 0	INITED	$FF = Initialization complete
1	LOGSWITCH	$00 = No line trace logging
2	IDLE.TIME	Idle Line Time x 10 (5 -25 secs)
3	WABT.ACK.TIME	WABT delay time x 10 (.5 - 2.5 secs)
4	COMP.TYPE	Compression Type 0 = No compression 1 = Compress trailing blanks only 2 = Compress all blanks 3 = Compress all characters
5	BLK.COMP.CNT	Min # of blanks to compress (2 - 31)
6	DUP.COMP.CNT	Min # of chars to compress (2 - 31)
7	CON.MSG.SIZE	Console message size (up to 120)
8	MAX.BLK.SIZE	Line block size (150 - 512)
9	SPANNED.REC	Allowing spanning on Xmit if nonzero
10	TERM.TO.TERM	0 = T-2-CPU, nonzero = T-2-T
11	MODEM.TYPE	0 = Manual Answer, nonzero = Auto Answer
12	PRIM.SEC	0 = Primary, nonzero = Secondary
13	CVT.RCV	Convert Receive Data to ASCII if nonzero
14	MODEM.RATE	0 = Select Low Modem Rate
15	DELAYED.ENQ	0 = no delay, nonzero = delay line startup ENQ until SEND request received
16	CONTROL.TRANS	0 = send WABTs, Permission messages in nontransparent mode, nonzero = send in transparent mode.

MLAM Translation Tables

It is most common for S.E.'s to patch the translation tables in the 8437 in order to conform to special system requirements. With MLAM, the translation tables are located in the following locations:

$200 - $2FF	ASCII to EBCDIC
$300 - $3FF	EBCDIC to ASCII punch / console
$400 - $4FF	EBCDIC to ASCII printer

One takes the character to translate from and adds the value of that character to the beginning address of the appropriate translate table. The byte located at that location is the translated chracter. For example:

A Vision user wishes to have a logcial OR bar sent from the Host (EBCDIC $4F = 0117) print on a standard line printer as an exclaimation mark (ASCII $21 = 041). The S.E. would make the following change on the user's pack:

```
// DTCOMM
/ INPUT  = MLAMD4.
/ UPDATE = 041 @ $44F.   Translate $4F to 041.
/ OUTPUT = MLAMD4.
//
```

I.2 CFG327

```
// CFG327
/INPUT = xxxxxx@y  Input file (optional, default is DEFAULT@0)
/OUTPUT = xxxxxx   Output file (optional, default is CFGFIL)
/AUTOMATIC = xx.   40 or 80   Screen size of configuring system.
//
```

If /A is specified, the program operates as though SHIFTED DOWN ARROW were pressed repeatedly.

CFG327 Keyboard Entries:

CFG327 page 1:

```
+----------------------------------------------------+
+                                                    +
+ LINE LENGTH OF THIS SYSTEM 80;                     +
+                                                    +
+ INPUT FILE DEFAULT@0;  OUTPUT FILE CFGFIL          +
+                                                    +
+                                                    +
+                                                    +
+                                                    +
+ PRESS SHIFTED DOWN ARROW TO ADVANCE TO NEXT PAGE +
+----------------------------------------------------+
```

LINE LENGTH OF SYSTEM	Enter 40 or 80.
INPUT FILE	Enter input configuration file and drive.
OUTPUT FILE	Enter output configuration file.

CFG327 page 2:

```
+----------------------------------------------------+
+ APPLICATION NTP100; REMOTE; COBOL @ FIELD? N/A;    +
+ MEMORY SIZE IS 72K BYTES; SCREEN SIZE IS 80x24;    +
+ TRANSMISSION CODE(EBCDIC/ASCIIA/ASCIIB)? EBCDIC;   +
+ LINE SPEED IS 4800 BPS; CONTROL UNIT ADDRESS 40;   +
+ MAXIMUM MESSAGE LENGTH (WITH ORDER BYTES) 2000;    +
+ DEBUG? N; STRING EDITOR DEBUG? N/A                 +
+ LOG LENGTH N/A;  ; CHECKPOINT? N; CKPT DEV N/A;    +
+ TAB TO COLON? N;  STORE AND FORWARD? N;            +
+ LINE DISCIPLINE BSC ;  IS NCP GENNED FOR NRZI N;   +
+                                                    +
+ PRESS SHIFTED DOWN ARROW TO ADVANCE TO NEXT PAGE +
+----------------------------------------------------+
```

APPLICATION	Enter 100 or 150.
REMOTE	Enter REMOTE or LOCAL.
COBOL @ FIELD	Enter Y or N.
MEMORY SIZE	Enter 24, 48, 72, or 96.
SCREEN SIZE	Enter 80x24, 80x12, or 40x12.
TRANSMISSION CODE	Enter EBCDIC, ASCIIA, or ASCIIB.
LINE SPEED	Enter 24, 48, 72, or 96.
CONTROL UNIT ADDRESS	Enter local control unit address or bisync control unit polling address or SNA physical unit address.
MAXIMUM MESSAGE LENGTH	Enter length of longest message expected.
DEBUG	Enter Y or N.
STRING EDITOR DEBUG	Enter Y or N.
LOG LENGTH	Enter 0 to use all available memory. Compute log area size to allow for NTP/150 overlays.
CHECKPOINT	Enter Y or N.
CKPT DEV	Enter 8230, 8240, 8250, Or 8260.
TAB TO COLON	Enter Y or N.
STORE AND FORWARD	Enter Y or N.
LINE DISCIPLINE	Enter BSC or SDLC. (N/A for LOCAL)
NRZI (Non Return to Zero Inverted)	Enter Y or N to match the specification in the NCP 3704 or 3705 sysgen.

(CFG327 cont.)

CFG327 page 3:

```
+----------------------------------------------------+
+ KEYBOARD TYPE 0 IS PRINTER ONLY;  TYPE - = DONE   +
+ KEYBOARD TYPE 1 IS 7226;                           +
+ LOWER CASE? N; NUMERIC LOCK? Y;                    +
+ KEYBOARD TYPE 2 IS 7200;                           +
+ LOWER CASE? Y; NUMERIC LOCK? N;                    +
+ TERMINAL  1 USES KEYBOARD TYPE 1 (0,1,2,-);        +
+ IN SCREEN POSITION 01;     WITH POLL ADDRESS 40;  +
+ AUDIBLE ALARM? Y;   INITIAL INTERCEPT FLAG? N/A;  +
+ PRINTER TYPE 81XX  (XX=NONE, 21=8121, LP=OTHER);  +
+         AT HARDWARE ADDRESS N/A;                   +
+                                                    +
+ PRESS SHIFTED DOWN ARROW TO ADVANCE TO NEXT PAGE  +
+----------------------------------------------------+
```

KEYBOARD TYPE 1	Enter 7200, 7203, 7226, 7227, 7228, or U0-U9.
LOWER CASE	Enter Y or N.
NUMERIC LOCK	Enter Y or N.
KEYBOARD TYPE 2	(same as for keyboard type 1)

Repeat the following entries for each terminal:

TERMINAL n USES KEYBOARD TYPE	Enter 0 for printer 1 for keyboard type 1 2 for keyboard type 2 - for no more terminals.	
SCREEN POSITION	Enter 01 - 32 (01 - 24 LOCAL).	
POLL ADDRESS	Enter local device address BSC device polling address SNA logical unit address	(LOCAL) . (BSC) (SDLC)
AUDIBLE ALARM	Enter Y or N.	
INITIAL INTERCEPT	Enter Y or N.	
PRINTER TYPE	Enter 8121 or 81LP.	
AT HARDWARE ADDRESS	Enter octal 000 - 037 030 - 031	(8121) (81LP)

CFG327 page 4:

```
+----------------------------------------------------+
+ IS LOWER CASE USED BY PRINTERS? N/A;               +
+ WHAT IS THE 8121 DEFAULT FOR LINES/PAGE? N/A;      +
+ WILL SB3270 OPEN FILES? N/A;                       +
+ NUMBER OF DISC RETRIES? N/A;                       +
+                                                    +
+ LOAD MODULE NAME? SIM327;                          +
+                                                    +
+ PRESS SHIFTED UP ARROW TO REVIEW CONFIGURATION     +
+    OR PRESS SHIFT DOWN ARROW TO FINISH             +
+----------------------------------------------------+
```

LOWER CASE BY PRINTERS	Enter Y or N.
DEFAULT LINES/PAGE	Enter 01 - 99.
SB3270 OPEN FILES	Enter Y or N.
DISC RETRIES	Enter 0 - 9.
LOAD MODULE NAME	Enter any valid IDOS file name.

II.1 ESCAPE KEY FUNCTIONS

Key	Function	Password	Basic	Debug	S&F	Printer
1	Toggle CG	FOURFAZE	X			
B	Exit to DOS	FOURFAZE		X		
C	Take checkpoint	FOURFAZE		X		
D	Dynamic Dump	FOURFAZE		X		
L	Set number of lines per page	none				X
M	Set address of Dynamic Dump Dump	FOURFAZE		X		
P	Store password	none	X			
V	Store value string in address	FOURFAZE		X		
W	Set printer spacing	none				X
Z	Print screen.	none	X			X
ESC	Store attribute character	FOURFAZE		X		

* - Remote only.

\+ - For details about S&F, see the 3270 Operator's Manual

** - Must be an unbuffered character printer

II.2 TRC327

TRC327 is used to display and/or print a formatted output of the debug log of a checkpoint file.

```
// TRC327
/I=xxxxxx@n.                   Checkpoint file name and drive.
/H=hh.                         Screen height.
/W=ww.                         Screen width.
/T=t                           Trace type.  (S for 3270 simulator)
//
```

Parameters not entered by JCL can be entered from keyboard 0:

```
+----------------------------------------------------+
+ ENTER OPTIONS: INPUT (FILE NAME/SECTOR ADDR)       +
+ HEIGHT (6/12/24), WIDTH (48/81). /I. WILL EXIT.    +
+ TYPE (S/V/T/M/L/R) S=3270/8437,V=VISION/8436       +
+ T=VISION/8437,M=MLAM/8437,L=LAM/8437,R=3770/8437   +
+ W=REMOTE TERMINAL, C=NTP/250                       +
+                                                    +
+----------------------------------------------------+
```

After parameters are entered and a valid log information table (LIT) has been found in the checkpoint the trace display can be initiated from the following screen:

```
+----------------------------------------------------+
+ TO ADVANCE 1=1 LINE, H=1/2 SCREEN, F=FULL SCREEN   +
+    P = START OR STOP PRINTING                      +
+    R = RESTART PROGRAM, E = EXIT PROGRAM           +
+    S = TURN FULL SYSTEM TRACEON OR OFF             +
+    ATTN = ENTER OPTION PARAMETERS                  +
+    L = GO TO END OF FILE (LAST 6 SECTORS)          +
+    B = BACKUP 1 SECTOR                             +
+ L AND B NOT IMPLEMENTED FOR CHAINED BLOCKS         +
+----------------------------------------------------+
```

III.5 TRC327 OUTPUT

```
     Trace Log Entry Types:

HOST   xx xx ... xx  `aa ... a' dd ... d
              xx ... xx      Up to 16 bytes of data from host in hex.
              aa ... a       ASCII print interpretation of data.
              dd ... d       Desciption of any line control information.

TERM          (Same as for HOST)

TIME  dd ... d
              dd .... d      One of the following messages:
                             3 MINUTE TIMER HAS BEEN SET
                             1 SEC TIMER HAS BEEN SET
                             WACK TIMER HAS BEEN SET
                             ENQ TIMER HAS BEEN SET
                             TEXT TIMER HAS BEEN SET
                             3 MINUTE TIMER HAS GONE OFF
                             1 SEC TIMER HAS GONE OFF
                             WACK TIMER HAS GONE OFF
                             ENQ TIMER HAS GONE OFF
                             TEXT TIMER HS GONE OFF

STAT  R9001STATUS oooooooo dd ... d
              oooooooo       Octal value of status
              dd ... d       Interpretation of status bits

              .1......................  Not SDLC.
              ..1.....................  Not flag detect.
              ....1...................  Not abort detect.
              .....1..................  Not ring detect.

STAT  R9002STATUS oooooooo dd ... d
              oooooooo       Octal value of status.
              dd ... d       Interpretation of status

              ..1.....................  Character lost.
              ...1....................  Not BSC.
              .....1..................  Transmit underrun.
              ......1.................  Signal quality low.
              .......1................  Not carrier.

STAT  R9003STATUS oooooooo dd ... d
              oooooooo       Octal value of status.
              dd ... d       Interpretation of status.

              1.......................  IOID loaded.
              ..1.....................  No clock interrupt.
              ...1....................  Transmit buffer empty.
              ....1...................  DMA interrupt.
              .....1..................  Receive buffer full.
              ......1.................  Data set not ready.
              .......1................  Not clear to send.


C8437  xx xx xx . . . xx
              xx ... xx      Request table in hex.


              **** REQUEST TABLE ON FOLLOWING PAGE ****
```

REQUEST TABLE FORMAT

Word:	Byte:	Bits:	Descpiption:
0	0	1...	Block complete
0	0	..1.	IO for block is complete.
0	1		Control data: Link addr high. Status data: Error type.
0	2		Control data: Link addr low. Status data: Error value.
1	0		Line identifier.
1	1		Device identifier.
1	2		Control unit identifier.
2	0		Command type:
		0000 0000	Control.
		0000 0001	Read.
		0000 0010	Write.
		0000 0011	Status.
2	1		Command modifier:
		0000 0000	Open.
CTL CMD		0000 0100	Set busy.
		0000 0101	Set not busy.
		0000 0110	Set ready.
		0000 0111	Set not ready.
		0000 1000	Cause transmit.
		0000 1001	Cancel transmit.
		0000 1010	Select.
		0000 1011	Deselect.
		0000 1100	Copy.
		0000 1101	WCC.
		0000 1110	Time.
		0001 0001	Copy select.
		0001 0010	Copy deselect.
		0001 0011	LOC.
READ CMD		0000 0000	Read buffer.
		0000 1010	Read select.
WRT CMD		0000 0000	Write buffer.
		0000 0001	Write select.
2	2		Control byte or WCC.
3	0		AID or SLCO.
3	1		Buffer address page.
3	2		Page offset.
4	0		Not used.
4	1		Page link number.
4	2		Page link offset.
5	0		"from" device for copy.
5	1		Post processor page number.
5	2		Post processor page offset.
6	0		Not used.
6	1		Buffer page number.
6	2		Buffer page offset.

S8437	xx xx ... xx	(Same as C8437)
CLOCK mm:ss:tt		
	mm	Minutes.
	ss	Seconds.
	tt	Tenths of seconds.
ADDR aaaaaaaa		
	aaaaaaaa	Octal address of NTP/150 call.
WSTAT ssssssss		
	ssssssss	7074 (WIDGET) status word.
WCNTL cccccccc		
	cccccccc	7074 control word

(TRACE LOG ENTRIES CONT)

```
TASK  aaaaaaaa hh:mm:ss.tt
              aaaaaaaa       Octal address of calling routine.
              hh:mm:ss.tt    Time of call in hours, min, sec, tenths.

COBOL eeeeeeee dddddddd ssssssss hh:mm:ss.tt
              eeeeeeee       Entry code.
              dddddddd       Detail code.
              ssssssss       User subscript.
              hh:mm:ss.tt    Time in hours, min, sec, tenths.

C7073 cccccccc dd ... d
              cccccccc       Octal 7073 control word.
              dd ... d       Interpretation of control bits.

S7073 ssssssss dd ... d
              ssssssss       Octal 7073 status word.
              dd ... d       Interpretation of status bits.

A7073 aaaaaaaa    AID CURSOR
              aaaaaaaa       Octal word containing AID and cursor addr
                             sent to channel by 7073.

X7073 aaaaaaaa    BEGXFER      mmmmmmmm     ENDXFER
              aaaaaaaa       Octal starting addr of 7073 data transfer.
              mmmmmmmm       Octal ending addr of 7073 data transfer.
```

SECTION III SDLC and SNA PROTOCOL

III.1 SNA TERMINOLOGY

VTAM - Virtual Telecommunications Access Method.
An IBM program which controls all communication flow.

SSCP - System Services Control Point.
A part of VTAM through which every message, request, or response flows.

NCP - Network Control Program.
A line control program in a 3705 controller that communicates with terminal using SDLC.

PU - Physical Unit.
A terminal control unit such as a 3271-12.

LU - Logical Unit.
A terminal such as a 3277, OR an application program in the host which uses VTAM such as CICS.

PIU - Path Information Unit.
The TH, RH, and RU that make up an I-frame.

TH - Transmission Header.
The first few bytes of text in an I-frame.

RH - Request header.
A header on the first block of a multi-frame message.

RU - Request or Response Unit.
A command, response, or text in an I-frame.

III.2 SDLC LINE TRANSMISSIONS

Each SDLC transmission consists of one or more frames seperated by a special flag byte X"7F". Each frame is individually addressed so that a transmission can contain frames destined for different stations (PU's or LU's in SNA terminology).

An SDLC Transmission:

```
F            F            F                                      F
L  (FRAME-1) L  (FRAME-2) L  . . . . . . . . . . . . . (FRAME-n) L
A            A            A                                      A
G            G            G                                      G
```

III.3 SDLC FRAMES

Each frame is bounded by two flag bytes. The first two bytes of a frame contain the destination address of the frame and the frame control byte. The last two bytes form the frame check sequence (FCS) that is similar to the BCC in Bisync.

An SDLC Frame:

```
F  ______________________________________________________________  F
L (ADDRESS) (CONTROL) (data . . .  data) (FRAME CHECK SEQUENCE)  L
A  — 1 ——————— 1 ———————————— n ———————————————— 2 ——————————    A
G                                                                G
```

Frame Control Byte:

BITS	
.... ..x0	I-FRAME, This frame contains an information unit.
.... ..01	This frame contains a supervisory command.
.... ..11	This frame contains a non-sequenced command.

For I-FRAMEs:

rrr. ...0	rrr = Number of frames received modulo 7.	
...p ...0	p = 1 means the PU is being polled from host.	(IN)
...f ...0	f = 1 means the last frame until polled by host.	(OUT)
.... sss0	sss = Number of frames sent modulo 7.	

Note: rrr and sss act like ACK's for up to 7 messages at a time.

For a Supervisory Frame:

rrr. ..01	rrr = Number of frames received modulo 7.
...x ..01	x = poll/final bit as in an I-FRAME.
.... xx01	xx = Receive status:
00	Receive Ready
01	Receive Not Ready Needs ack for frames sent.

For a Non-sequenced Frame:

cccc cc11	cccccc = Response or Command:
1001 00	Set Normal Response Mode Command
0101 00	Disconnect Command
0111 00	Non-sequenced Acknowledgement Response
0001 11	Request Online Response
1001 11	Command Reject Response
1111 00	Link test

III.4 FORMAT ID 3

I-FRAMES are formatted in different ways as they pass through an SNA network depending on the type of data link and type of station. Between the NCP and a 3271 PU they use Format ID 3 (FID3).

FID3

```
F A C  ---- TH ----------------- RH ------------------ RU -----   F F F
L D T (TRANSMISSION HDR) - (REQUEST HDR) - (REQUEST or RESPONSE)  C C L
A D L  ---- 2 ------------------ 3 ------------------- n ------   S S A
G R                                                                   G
```

Byte	Bits	Description
0	0011	Indicates that the TH uses FID3.
0	.... cc..	Chaining flags:
	00	Middle in chain. (NO RH IN FRAME !!!)
	01	Last in chain. (NO RH IN FRAME !!!)
	10	First in chain.
	11	Only in chain.
0	.... ..x.	Primary to secondary indicater.
0	x	Expedited flow indicator.
1	x...	LU/SSCP flag. Routes response to APPL or VTAM.
1	.xxx xxxx	LU device address.
2	x...	Request/Response Flag:
	0	Request
	1	Response
2	.x..	Request or Response Type:
	0	NCP
	1	Data
2	..x.	Subsystem Control Indicator.
2	...0	Always zero.
2	.... x...	Format indicator.
2	.... .1..	Sense data included.
2	11	Always set to one.
3	1...	Definite response required or sent.
3	.00.	Always zero.
3	...1	Exception response required or sent.
3	.... 000.	Always zero.
3	1	Pacing response required or sent.
4	1...	Begin Bracket.
4	.1..	End Bracket.
4	..00	Always zero.
4	.... c...	Code selection.
	0	EBCDIC
	1	ASCII
4	.... .000	Always zero.
5	cccc cccc	Command. (optional)
	1010 0001	Clear.
	1111 1000	Pseudo Bid.

III.5 STATUS/SENSE BYTE for SDLC 3271

BYTE	BIT	DESCRIPTION
0	1...	PATH ERROR - The device address received (TH byte 2 bits 1 - 7) was invalid.
0	.xx. .xxx	RESERVED
0	...1	REQUEST ERROR - The first byte of the request unit (RU) was not a valid command.
0	.... 1...	REQUEST REJECT - A pseudo bid command or begin bracket bit (in RU) was sent to a device that has attention pending.
1	xxx. xx..	RESERVED
1	...1 ..11	Bits 3, 6, and 7 are set whenever REQUEST REJECT is set.
2	xxxx ...x	RESERVED
2	.... 1...	DEVICE BUSY - Receipt of a command that requests a print operation for a printer which currently executing another print operation. - Receipt of a copy command whose "from" device is a busy printer. OPERATION CHECK is also set.
2	.... .1..	UNIT SPECIFY - A copy command was received for a device that was locked for copy. OPERATION CHECK is also set.
2	.... ..1.	DEVICE END - During system initialization DEVICE END status is generated for each video and printer. - A printer has become ready to use after being previously inoperative.
3	xx..	RESERVED
3	..1.	COMMAND REJECT - An invalid or illegal 3270 command has been received.
3	...1	INTERVENTION REQUIRED - A copy command specified an inoperative printer as the "from" device. OPERATION CHECK is also set. - A write or erase/write command was received for an in operative printer. - A pseudo bid was received for an inoperative printer.

(STATUS/SENSE BYTES CONT)

3	.... 1...	EQUIPMENT CHECK - This bit is not used by the BK01 simulator.
3	.... .1..	DATA CHECK - This bit is not used by the BK01 simulator.
3	.... ..1.	CONTROL CHECK - This bit is not used by the BK01 simulator.
3	1	OPERATION CHECK - Receipt of an invalid buffer address or an incomplete order sequence on a write or erase/write command. - Receipt of a copy command without a copy control character or without a "from" device specification character. - Receipt of a copy command with an invalid or inoperative "from" device. INTERVENTION REQUIRED is also set. - Receipt of a copy command whose "from" device is a busy printer. DEVICE BUSY is also set. - Receipt of a copy command whose "from" device is locked for copy. UNIT SPECIFY is also set.

IV. DTCOMM UTILITY

```
// DTCOMM
/INPUT =filename @ drive
/DISPLAY = byte address
/UPDATE = value
/OUTPUT = "PRINT" @ starting address
 (or)   = filename @ drive
/PATTERN = pattern value @ starting address
```

Note: Specify addresses, values and patterns in octal as "0nnn" or in hex as "%xx".

Parameter	Explanation
/INPUT	The input file at the selected drive will be loaded into the 8437 starting at location zero.
/DISPLAY	Displays on the video 256 bytes of 8437 memory starting at the selected byte address. The 8437 registers are also displayed.
/UPDATE	Replaces the selected byte with the specified value. DTCOMM enters display mode using the address of the modified byte as its starting point. Subsequent changes automatically increment this address.
/OUTPUT	The contents of the 8437 are dumped to the printer or a disk file starting at the selected address. The file dump is formatted as one 8437 byte per IV/70 word so that the file can be reloaded into the 8437.
/PATTERN	The 8437 is filled from the starting address to the end of memory with the selected 16 bit value.

commands to be entered.

```
+----------------------------------------------------+
+ DEBUGGING TOOL FOR COMMUNICATIONS CONTROLLER B02   +
+ /ENTER COMMANDS HERE !!!                           +
+ REG. P = pppp  /INPUT   = FILENAME @ ADDRESS       +
+ REG. A = aaaa  /DISPLAY = ADDRESS                  +
+ REG. X = xxxx  /UPDATE  = VALUE @ ADDRESS          +
+ REG. Y = yyyy  /OUTPUT  = "PRINT @ ADDRESS         +
+ STACK  = llll           = FILENAME @ ADDRESS       +
+ STATUS = ssss  /PATTERN = VALUE @ADDRESS           +
+ 0000  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF     +
+ 0010  FFFF           . . .                FFFF     +
+  .    FFFF           . . .                FFFF     +
+  .    FFFF           . . .                FFFF     +
+  .    FFFF           . . .                FFFF     +
+ 00F0  FFFF FFFF FFFF FFFF  FFFF FFFF FFFF FFFF     +
+----------------------------------------------------+
```

SECTION V RELATED INFORMATION

The following information related to the BK01 simulator can be found in section AG of this handbook:

SECTION	TOPIC
I	SYSTEM CONFIGURATION
I.5	MINIGEN
I.6	Store and Forward
I.7	TBLEDT
II	NTP/150
II.1	SB3270 Interface
II.2	Entry Codes
II.3	Detail Codes
II.4	NTP/150 Subroutines
II.5	SBRSET
II.6	Software Action Codes
III	DEBUGGING
III.2	Display of Attribute Characters
III.6	Taking Memory Dumps
III.7	DMP327 Execution
III.8	DMP327 Output
IV	BISYNC and 3270 PROTOCOL
IV.1	3270 Control Characters
IV.2	Bisync Data Link Control
IV.3	3270 Message Formats
IV.4	Local and Remote Command Codes
IV.5	Write and Copy Control Characters
IV.6	Buffer Control Orders
IV.7	Attribute Bytes
IV.8	Attention ID Byte
IV.9	Remote Status/Sense Bytes (BSC)
IV.9	Local Status and Sense Bytes
V	COMMUNICATION CONTROLLERS
V.2	7073 Status, Control, and IOID's
V.3	7074 Status and Control (WIDGET)

FIXED LOCATIONS WITHIN M.F.E.

MEM LOC	DESCRIPTION
************** GOLDEN RAM	****************
0 - 077	IOID 0
0100 - 0177	COMM IOID TABLE
0200 - 0277	812x PRINTER IOID TABLE
0300 - 0677	RETURNED AS "FREE GOLDEN RAM"
0700	POINTER TO ACTIVE PARTITION CONTROL BLOCKS QUEUE (PQ).
0701	POINTER TO TRANSITIONAL MONITOR QUEUE (TMQ).
0702 - 0704	OTHER FIXED ENTRY POINTS IN APPEARING IN IDS-CB.
01000 - APROX	FIXED ENTRY POINTS IN RDS-SY
APROX - 03000	CODE EXECUTING IN GOLDEN RAM
03000 - MFETOP	DYNAMIC GOLDEN RAM
*************** MFESYS	******************
MFETOP - #OVART	MFE SYSTEM EXECUTIVE OVERLAY AREA
#OVART - RESTOP	MFESYS CODE
RESTOP - MAXMFE	MFESYS DYNAMIC RAM, ALLOCATED AS NEEDED FOR MSG BLOCKS, CONTEXT BLOCKS, TCBS, ERROR LOG BLOCKS
MAXMFE - 077777	JOB SCHEDULERS AND TRANSITIONAL MONITORS EXECUTE HERE

WHERE:

```
    MFETOP EQU 04000
    #OVART EQU 07000
    RESTOP EQU 031000
    MAXMFE EQU 074000
```

All windows allocated by MFESYS map the same physical pages in RAM between 0 and MFETOP.

THE PROGRAM LOAD TABLE (PLT)

The PLT describes to MFE the memory, interrrupt levels, etc. that a new job requires to be executed. It is a preamble to any job that wishes to execute under MFE.

The following example is taken from MFEDYN as a sample PLT.

```
PLTAO    BRA     APLO      pointer to application open routine.
PLTLEV   PZE     MFELEV    resolved as the current level of MFE.
         BSS     6         reserved for MFE.
PLTUSA   DCN     0         pointer to 16 word inter-program comm region.
PLTLPO   BSS     1         LPOUT assignment for this partiton.
PLTWIN   BSS     1         primary window assignment.
PLTSCN   BSS     1         screen type.
PLT:TO   PZE     :TOPTL    timer stack.
PLTLV7   DCN     0         pointer to level 4 interrupt routine.
PLTAC    PZE     APLC      pointer to application close routine.
PLTTO    PZE     APLTO     pointer to terminal open routine.
PLTMIN   PZE     MFETOP    load above MFE.
PLTMAX   PZE     $TOP      top of application.
PLTTIM   BSS     2         time slice address vectors.
         BSS     2         reserved for MFE.
PLTTRP   DCN     0         hardware trap flag.
PLTCSA   DCN     0         address of console attention routine.
```

PARTITION CONTROL BLOCK - PCB

The Partition Control Block is the primary control block for any tas executing, or awaiting execution, under the Multi-function Executive.

+0	PLINKF PCB CHAIN POINTER ('NEXT PARTITION CONTROL BLOCK)	
+1	PTCB POINTER TO PRIMARY TASK CONTROL BLOCK (bits 0-8 must=0!)	
+2	PWNDOW PRIMARY WINDOW NUMBER IN BITS 1-8 (rest = 0)	
+3	P:TOPT POINTER TO TIMER STACK (IDOS EQUIVALENT OF :TOPTL)	
+4	PAPL7 POINTER TO INTERRUPT LEVEL 7 ROUTINE (FROM PLT)	
+5	PAPLTO POINTER TO TERMINAL LOGON ROUTINE	
+6	PAPLC POINTER TO APPLICATION PROGRAM CLOSE (STOP COMMAND)	*1*
+7	PKYVAL bits 0-15 = JOB SEQ #, bits 16-23 = KEYVALUE assigned to job	
+010	PDBASE SECTOR ADDRESS OF LOAD MODULE (LM) ON DISC	
+011	PAPLO SECTOR COUNT + LOAD POINT (used by Dispatcher, zeroed by MFERDY	*2*
+012	PCSA POINTER TO CONSOLE INTERRUPT ROUTINE (APLCSA)	
+013	PSTATE PCB DISPATCH STATE CODE (If=0, PCB in PQ, else in TMQ)	*=
+014- 015	PONAME TWO WORD NAME OF CURRENT OBJECT PROGRAM	
+016- 020	PSNAME STARTED PROGRAM NAME (2 WORDS) OR NAME ASSIGNED BY EXECUTING // ASSIGN WITH /NAME = XXXXXXXXX(UP TO 9 CHARACTERS)	
+021	PPZONE POINTER TO 3-WORD PREEMPTION ZONE TABLE (TIME SLICE OPTION)	|
+022	PUSARA POINTER TO 16-WORD INTER-PROGRAM COMMUNICATIONS BLOCK	
+023	PPGS NO.PAGES ALLOCATED - BIT 0=1 THIS VALUE SET BY /RAM ± ¦	
+024- 025	PNEXT LAST // NAME CONTROL CARD PASSED OPTION OR PASSED TO MFE $EXIT	

PCB (Continued)

+026	PSIB	"SYSTCK" BOTTOM	
+027	PSIC	"SYSTCK" CURRENT	
+030	PSIT	"SYSTCK" TOP	
+031	PPRI	JOB PRIORITY - BIT 0 = 1 THEN SET BY /PRI =	
+032	PFLAGS	FLAG WORD - BIT 8 =1 INDICATES STOP REQUEST EXECUTED	*4*
+033	PFMARK	MARK USED BY MFEDRL/$PEXIT TO RELEASE ACCESS TO FILES	
+034	PSTCTX	CURRENT PROGRAM CONTEXT SAVED HERE WHILE APLC EXECUTES (STOP)	
+035	PSTRP	SAVED RP FOR THE TASK "PREEMPTED" FOR APLC EXECUTION (STOP)	
+036	PDULIN	DUAL INTENSITY FLAG	*5*
+037	PAPLMS	POINTER TO APPLICTION APLMSG ROUTINE	
+040	PLPOUT	LPOUT VALUE FOR THIS PARTITION	
+041	PCON	CONNECTION CONTROL BLOCK LIST	
+042	PAPLTIM	POINTER TO APLTIME ROUTINE	
+043	PUID	POINTER TO USERID INFO (DEFAULTS TO USERID WHO STARTED JOB)	
+044	PSADB	PARTITION'S CONNECT/DISCONNECT SERVICE ADDRESS	
+045	PCPOST	POINTER TO COMPLETION CODE POST ROUTINE	
+046	PSECRT	SECRET WORD - PSSSSSSTT! SHHHHHHH!	
+047	PTIME	# OF LEVEL 4 TICKS BEFORE JOB SHOULD BECOME ACTIVE	
+050-052	PDPNAM	DESTINATION PRINTER NAME	

PCB (continued)

+053- 055	PFRNAM FORMS NAME
+056- 057	PSYNAM SYSOUT NAME
+060	PSYDRV SYSOUT DRIVE
+061	PCLSNM CLASS NAME ASSIGNED TO PCB + ADDR OF CLASS BYTE IN CMCLAS
+062	P2UID SECOND USERID TO SEND MSG IN CASE OF ABORT OR STOP

PCBLEN EQU 064 = 50

PCB NOTES

1 PAPLC (+6)
WHEN A STOP COMMAND IS FOUND FOR A PCB IN THE TMQ (TRANSITIONAL MONITOR QUEUE) THEN THIS ROUTINE POINTER IS VALID ONLY IF PFLAGS (PCB + 032) BIT 9 IS 1. THIS INDICATES THAT A LM IS IN THE PARTITION BUT HAS NOT EXECUTED "MFERDY".

2 PAPLO (+011)
SECTOR COUNT + LOAD POINT WHEN PASSED TO THE JOB DISPATCHER. WHEN "MFERDY" IS EXECUTED THIS WORD BECOMES ZERO. IF NON-ZERO WHEN $EXIT IS EXECUTED THE JOB STREAM IS ABORTED.

3 PSTATE (+013)
PCB DISPATCH STATE CODE:

- 0 = ACTIVE, PCB IS IN PQ (POINTER TO ACTIVE PCB'S). IF PCB IS IN TMQ (TRANSITIONAL MONITOR QUEUE) WITH THIS CODE THEN JOB HAS CALLED $EXIT, BUT NO OTHER PROCESSING HAS OCCURRED. $EXIT WILL LOAD ANOTHER JOB INTO PARTITION IF A CORRESPONDING ENTRY IS IN SYSTCK.

(FOR ALL OTHER STATES PCB IS IN TMQ)

- 1 = WAITING FOR JOB SCHEDULER;
- 2 = SCHEDULED , BUT WAITING ON RESOURCES;
- 3 = DISPATCHED BUT LM NOT LOADED
- 4 = APLO DISPATCHED BUT MFERDY NOT EXECUTED (MFERDY CHANGES STATE TO 0)
- 5 = WAITING FOR DISPATCH & MAYBE COMPETING FOR RESOURCES IN A TRANSITIONAL MONITOR

4 PFLAGS (+032)

BIT	FUNCTION
0	IF TRUE, JOB HAS BEEN ABORTED WITH TIME LIMIT EXCEPTION OR CALLED MFEPC WITH DUMP REQUEST
8	STOP REQUEST WAS EXECUTED
9	IF TRUE, WAITING FOR SYSOUT; RESET WHEN BIT 10 GOES TRUE
10	IF TRUE, PROGRAM HAS SYSOUT ALLOCATED
11	IF TRUE, ABORT WAS EXECUTED
12	IF TRUE, PROGRAM WAITING FOR $ICARD; RESET WHEN BIT 13 GOES TRUE
13	IF TRUE, PROGRAM HAS $ICARD ALLOCATED

5 PDULIN

BIT 0 = 0 -> PARTITION WANTS DUAL INTENSITY OFF; = 1 -> WANTS IT ON
1 = 0 -> PARTITION DOES NOT NEED DUAL INTENSITY DISABLED;
= 1 -> DOES NEED DUAL INTENSITY DISABLED.
9-23 = APL1SW POINTER IF NONZERO

MFE DUMP CONSIDERATIONS

IF you have configured for a dump you must create a dump file.

MFE UTILITY MAKD47

```
// MAKD47
/O=DISC@0         "DISC" AT DRIVE.  REQUIRED.
/BLNKS= 2.        FOR 192K SYSTEMS. ONE OF THESE IS REQUIRED.
        3.        FOR 288K SYSTEMS.
        4.        FOR 384K SYSTEMS.
```

If you have optioned for a dump MFE will dump when:

1. A MANUAL DUMP IS FORCED:
    - A. PLACE THE CPU IN MANUAL MODE.
    - B. ENTER 071100001 (BRM# 01) INTO THE CONSOLE KEYS.
    - C. PRESS RESET, STEP, AND LOAD KEYS IN THIS ORDER.
    - D. CLEAR HALT (MANUAL MODE TO AUTO MODE).
    - E. CLEAR HALT AGAIN WHEN MACHINE HALTS.

The dump will be written to disc (NOTE: If X3 is not = ZERO the dump may have failed.)

2. A system console operator ABORTS a job or task. (enters ABORT JOBXXX on the system console)

3. The software (either MFE or the application) causes a TIME LIMIT EXCEPTION upon an error condition.

PRINTING AN MFE DUMP:

The processor MFEDUMP prints a formatted dump of an MFE dump file.

MFEDMP is executed as follows.

1. Enter SYSTEM CONSOLE mode.
2. Type START,MFEDMP or // MFEDMP.
   (YOU MAY WISH TO CREATE A CONTROL FILE THAT HAS THE MFEDUMP OPTION JCL THAT CAN BE CALLED AT RUN TIME)
3. Specify MFEDMP options:

```
// MFEDMP

/RAM OR /ANALYSIS.              IF SPECIFICED THIS WILL CAUSE
                                EITHER THE FORMATTED SECTION
                                OR THE OCTAL RAM DUMP TO PRINT

                                IF OMITTED BOTH SECTIONS WILL PRINT.

/INPUT = FILENAME @ DRIVE.      IF OMITTED DEFAULTS TO "DUMP47 @ 0"


/WINDOW = WWW.                  OPTIONAL. DEFAULTS TO RUN TIME WINDOW.

/PHYSICAL.                      OPTIONAL INDICATES RAM IS TO BE PRINTED
                                IN PHYSICAL PAGE ORDER NOT LOGICAL
                                ADDRESSING BY WINDOW ORDER.

/LOW = LL.  (PAGE NO.)          OPTIONAL. LOW BOUNDRY.

/HIGH = HH. (PAGE NO.)          OPTIONAL. HIGH BOUNDRY.

/NAME = JOBNAME.                OPTIONAL. SETS WINDOW TO "JOBNAME"
                                PRIMARY WINDOW.
//
```

TASK CONTROL BLOCK

```
                              1 1 1 1 1 1 1 1 1 1 2 2 2 2
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
         ---------------------------------------------------
TLINK    |     ? ? ?       |            LINK WORD          |  + 0
         ---------------------------------------------------
TRBRG    |     ? ? ?       |    REGISTER SAVE BLOCK PTR    |  + 1
         ---------------------------------------------------
TWNDOW   |0| WINDOW NBR    |               0               |  + 2
         ---------------------------------------------------
TRP      |CONDITION CODES|        RESTART ADDRESS          |  + 3
         ---------------------------------------------------
TCLOCK   |           TEMPORARY PRIORITY                    |  + 4
         ---------------------------------------------------
         |S|                                               |
TPRI     |T|         PERMANENT PRIORITY                    |  + 5
         |P|                                               |
         ---------------------------------------------------
TPCB     |     ? ? ?       |        PCB POINTER            |  + 6
         ---------------------------------------------------
TTRAP    |     ? ? ?       | TRAP ROUTINE POINTER          |  + 7
         ---------------------------------------------------
TCBCHK   |             OCTAL  23456701                     |  + 8
         ---------------------------------------------------
TDAD     |       0         |     ANCESTOR'S TCB POINTER    |  + 9
         ---------------------------------------------------
TKID     |     ? ? ?       | POINTER TO DESCENDANTS LIST   |  +10
         ---------------------------------------------------
TSIB     |     ? ? ?       |   LIST OF PARALLEL TASKS      |  +11
         ---------------------------------------------------
TCTXP    |   LONG POINTER TO ADDITIONAL CONTEXT WORD       |  +12
         ---------------------------------------------------
TCTX     |              ADDITIONAL CONTEXT                 |  +13
         ---------------------------------------------------
TMNE     |               TASK'S MNEMONIC                   |  +14
         ---------------------------------------------------
TWKTIM   |                 WAKEUP TIME                     |  +15
         ---------------------------------------------------
TWAITL   |          WAITER'S QUEUE LINK WORD               |  +16
         ---------------------------------------------------
TWAITT   |                  WAIT TYPE                      |  +17
         ---------------------------------------------------
TCBTP    | LONG POINTER TO TASK TERMINATION PROCEDURE      |  +18
         ---------------------------------------------------
TTASOC   |A|          ZERO                     | TERM #    |  +19
         ---------------------------------------------------
TPWNDO   |              PERMANENT WINDOW                   |  +20
         ---------------------------------------------------
TFLAGS   |       |V| |X|Y|Z|     TERMINATION REQUESTOR     |  +21
         ---------------------------------------------------
TITCM    |         ICTM WAIT STATE INDICATOR               |  +22
         ---------------------------------------------------
TCIWT    |      MFEMEM OR MFEOIO WAIT INDICATOR            |  +23
         ---------------------------------------------------
```

SYSTEM MAINTENANCE SERVICES - FUNCTIONS

SMS FUNCTION ID	SMS PROCESS
0	Enter system terminal mode (LOGOFF)
1	System initialization (part 1)
2	$PEXIT (start transitional monitor)
3	Time limit exception message formatter
4	Terminal logon
5	Console ATTN processor
6	Time and date initialization
7	Error message processor
8	STOP command processor
9	Exit to IDOS
10	Program dispatcher
11	Request job scheduler
12	ABORT command
13	VARY STNOTE command processor
14	Reset time/date, execute APLTIME
15	System initialization (part 2)
16	MFEMEM, RAMWTQ processor

INTERRUPT CONTROL BLOCK

+0,+1	LISV23	X2, X3 Save area
+2,+3	LISVA1	RA, X1 save area
+4	LISVB	RB save area
+5	LISVTX	$CTX for this level
+6	LISMAP	Contents of window reg at interrupt
+7	LISCTX	Contents of $CTX at interrupt

CU3270 / 8437

Release BP03

A32ERR Error Codes

The following error codes are passed to the application error routine, A32ERR. The positive errors are information messages. The negative errors are catastrophic internal errors which cause CU3270 to call $XCLOSE to save the application. (The negative errors appear in VISION as "CU3270 INTERNAL ERROR ###" with the value filled in.)

Error	Meaning
0	Previous error conditions are cleared.
1	Data set not ready.
3	Host computer has not addressed CU3270 in more than 3 minutes.
-1	Illegal command received from the 8437.
-2	Illegal command modifier received from the 8437.
-3	8437 attempted to select a previously selected device.
-4	8437 issued a print request for a device that is not selected.
-5	$XFER error. $XFER's error code is passed in register RB.
-6	8437 received an invalid request from the IV/xx.
-7	8437 attempted to pass a completion when there are no outstanding requests.

C32xxx Routine Error Codes

The following error codes are passed in register RA when the application gets the error return from the C32xxx routines (C32SB, C32SNB, C32SR, C32SNR, C32QTX, C32CTX, C32DBG). Note that C32STA is usually included in this group but it has no error return and no error codes.

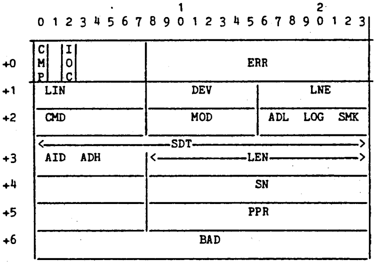
Error	Routines	Meaning
0	All	The device is currently selected.
1	All but SR	The device has an AID transmission pending.
2	SB, QTX	The device is currently BUSY.
3	All	CU3270 is not open.
4	All	Invalid device ID number.
5	SNB, SNR, QTX	The device is currently Not READY.
6	SNB	The device is already Not BUSY.
7	SR	The device is already READY.
8	CTX	There is no transmission pending for this device.
9	DBG	A prior C32DBG call has not yet completed.

CU3270 8437 Dump Analyzer

There are two versions of the dump analyzer. DMP32 is the IDOS version and MDMP32 is the MFE version. The JCL is the same for both versions. DMP32/MDMP32 can interpret either the contents of the 8437 RAM or an IDOS file created by DTCOMM. The default values are logical controller 0 and the load module type (as identified by location $100).

```
// DMP32 or MDMP32          To start execution of the dump analyzer.
/ INPUT = file @ drive.     Analyze 8437 dump file on the disc (overrides /L).
  or
/ LOGICAL = controller.     Analyze 8437 RAM in the Logical 8437 specified.
/ TYPE = BA, BE, SA, SE.    Force the interpretation type.
//
```

Requests from the IV/xx and the Completions from the 8437



Word	Byte	Definition
0	0	CMP/IOC. $20 = Request tables from the IV/xx. $80 = Completion tables from the 8437.
	1-2	ERR. Ignore in requests to the 8437. The error code is contained in these bytes on completions from the 8437.
1	0	LIN. Not used.
	1	DEV. Logical device number (0-31).
	2	LNE. Logical 8437 to which this request is addressed.
2	0	CMD. Command type: 0 = Control 1 = Read (from the 8437 to the IV/xx) 2 = Write (from the IV/xx to the 8437)
	1	MOD. For Control commands (CMD=0): 0 = No command (occurs in completions sent by the 8437). 1 = Open (associated buffer = open parameter table) 2 = Hard Close 3 = Soft Close 4 = Set Busy* 5 = Set Not Busy* 6 = Set Ready* 7 = Set Not Ready* 8 = Queue Transmission* 9 = Cancel Transmission* 10 = Change Logging* Note: Requests with modifiers marked with an asterisk (*) do not cause the 8437 to send a completion.
	1	MOD. For Read or Write commands (CMD = 1 or 2): 1 = R/W Select Table (associated buffer = address of the Select Table). 2 = R/W Copy "from" device Select Table (associated buffer = address of the Copy Select Table). 3 = R/W Screen Buffer (associated buffer = First word address of the screen). 4 = R/W Copy "from" device Screen Buffer (associated buffer = First word address of the screen). 5 = For Read Command: request 8437 RAM 'snapshot' dump (associated buffer = address to read dump into). 6 = For Read Command: request 8437 Comm status counters (associated buffer = address to read counters into).

Word	Byte	Definition
2	2	LOG. For change logging (CMD=0, MOD=10): 0 = Turn logging off. 1 = Log request tables only. 2 = Log line data and request tables. 3 = Log line data, request tables and 8437 status timers. 4 = Log line, request tables, 8437 timers and DMA events.
	2	SMK. For Read Screen Buffer completion: 4 = Byte 0 of last word has screen data (40 column screens) 6 = Bytes 0 and 1 have screen data (80 column screens)
	2	ADL. Low address byte for 8437 dump request (CMD=1, MOD=5).
3	0	AID. For Queue Xmit (CMD=0, MOD=8), Logical AID as follows: 2 = PA3 5 = PA2 20 = ENTER 3 = PA1 7 = TEST REQUEST 1 & 6 = Reserved 4 = CLEAR 8-19 = PF1-PF12
	0	ADH. High address byte for 8437 dump request (CMD=1, MOD=5).
	0-2	SDT. Last word of the screen for Read Screen (CMD=1, MOD=3).
	1-2	LEN. Contains the length of the associated buffer.
4	0-2	SN. Used by $XFER
5	0-2	PPR. Address of Level 1 routine to receive completion table.
6	0-2	BAD. Address of the associated buffer.

Requests from the 8437 and the Completions from the IV/xx

		1	2
	0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3
+0	CMP IOC	ERR	
+1	LIN	DEV	LNE
+2	CMD	MOD	WCC
+3	CCM	CPL	

Word	Byte	Definition
0	0	CMP/IOC. $00 = Request tables from the 8437. $A0 = Completion tables from the IV/xx.
	1-2	ERR. Contains error code in status requests from the 8437 only.
1	0	LIN. Not used.
	1	DEV. Logical device number (0-31).
	2	LNE. Logical 8437 from which this request originated.
2	0	CMD. Command type: 0 = Control 3 = Status (there is no modifier for status commands).

Word	Byte	Definition
2	1	MOD. For Control commands (CMD=0): 1 = Select device (causes A32SL call). Note that a Write Select Table request (CMD=2, MOD=1) from the IV/xx is a positive response to this request. 2 = Deselect device (causes A32DSL call).* 3 = WCC (print) request (causes A32PR call). 4 = Select Copy "from" device (causes A32CCS call). 5 = Deselect Copy "from" device (causes A32CCD call).* Note: Requests with modifiers marked with an asterisk (*) do not cause the IV/xx to send a completion.
	2	WCC. Used for Deselect and WCC requests (CMD=0, MOD= 2 or 3). Contains bits 16-23 of the Communications Operation Word.
3	0	CCM. Used for Deselect and WCC requests (CMD=0, MOD= 2 or 3). Contains bits 8-15 of the Communications Operation Word.
	1	CPL. Application's response to Select or WCC (print) request (CMD=0, MOD= 1, 3 or 4): 1 = Accepted, 2 = Rejected.
	2	Not used.

CU3270 Timers

Number	Meaning
0	Host Down timer. (3 minutes) Set every time the host sends something to CU3270. This is the only timer used by the SDLC versions of CU3270.
1	Control Mode timer (1 second) Set every time we receive an EOT from the Host. Cleared when the background monitor is returned to the 8437. If the timer expires, it implies an unknown problem has occured.
2	WACK Response timer (0.1 second) Set when CU3270 receives a WACK in response to text. When it expires, we send an ENQ.
3	ENQ timer (3.5 seconds) Set when CU3270 finishes sending text. Cleared when a response is received. If it expires, we send an ENQ.
4	Receive Text timer (3.0 seconds) Set when CU3270 starts receiving a block and every time it encounters an imbedded Sync. Cleared on receive of ETX or ETB. If it expires, we assume loss of ETX or ETB and send a NAK.
20*	Received a request table from the IV/xx.
21*	Processed a request table from the IV/xx.
22*	Queued a request/response table to be sent to the IV/xx.
23*	Sent a previously queued request/response table to the IV/xx.
24*	Sent a block of Log data to the IV/xx.
25*	Received up to 256 bytes of data from the IV/xx. The IV/xx performed a Data-Out I/O instruction and the 8437 has started to DMA the data into 8437 RAM.
26*	Sent up to 256 bytes of data to the IV/xx. The IV/xx performed a Data-In I/O instruction and the 8327 has started to DMA the data out of 8437 RAM.
27*	Received an unknown interrupt from the IV/xx.
30*	IV/xx is trying to send a request table to the 8437 and the 8437 has no room for it.
31*	8437 received data from the line while transmitting. The received data is disregarded while the transmission is inprocess.

Note: Those timers that are marked with an astersik (*) are used by CU3270 to mark internal events and are not really timers.

Communications Operation Word

Bits	Meaning
0 - 8	Not used
9	Erase Write Command
10	Erase All Unprotected Command
11	Copy Command
12	Write Command
13	Read Modified Command
14	Read Buffer Command
15	Escape-in-Data Indicator
16 - 17	Reserved
18 - 19	Print Format Bits: 00 - Unformatted 10 - 64 Char/line 01 - 40 Char/line 11 - 80 Char/line
20	Start Print Command
21	Not used
22	Restore Keyboard
23	Not used

3271 SDLC

SDLC Frame

<——— Link Header ———> <— Link Trailer —>

Flag 01111110	Address	Control	P.I.U. I-frame only	FCS 16 bits	Flag 01111110

Nonsequenced Commands and Responses

Control	C/R	Usage	P/F	¬(P/F)
SNRM	C	Set Normal Response Mode	'93'	'83'
DISC	C	Disconnect	'53'	'43'
SIM*	C	Set Initialization Mode	'17'	'07'
RIM*	R	Request Initialization Mode	'17'	'07'
UI (NSI)*	C/R	Unnumbered Information	'13'	'03'
UP (NSP)*	C	Unnumbered Poll	'33'	'23'
XID*	C/R	Exchange IDs	'BF'	--
TEST	C/R	Link Test	'F3'	'E3'
UA (NSA)	R	Unsequenced Acknowledgement	'73'	'63'
DM (ROL)	R	Disconnect Mode	'1F'	'0F'
FRMR(CMDR)	R	Frame Reject (command reject)	'97'	'87'
RD*	R	Request Disconnect	'53'	'43'

Commands marked with and asterisk (*) are not supported by a 3271 and will result in a frame rejected (FRMR).

Supervisory Commands and Responses

Control	Usage	(See Below)
RR	Receive Ready	'a1'
RNR	Receive Not Ready	'a5'
REJ*	Reject	'a9'

Commands marked with and asterisk (*) are not supported by a 3271 and will result in a frame rejected (FRMR).

Information Commands and Responses

Control	Usage	(See Below)
I	Information Frame	'ab'

Hexadecimal digit for "a"

Nr=	P/F	¬(P/F)
0	1	0
1	3	2
2	5	4
3	7	6
4	9	8
5	B	A
6	D	C
7	F	E

Hexidecimal digit for "b"

Ns=	Hex
0	0
1	2
2	4
3	6
4	8
5	A
6	C
7	E

Path Information Unit (PIU -- I-frames only)

TH	RH	RU (request unit)
2 Bytes	3 bytes	Up to 256 Bytes

TH - Transmission Header (on all I-frames)

Byte	Bits	Description
0	0011	FID. Format Identifier. Always 0011.
	.... cc..	MPF. Mapping Field. Placement of segment in the PIU.
	00	Middle segment of PIU. (Frame contains no RH).
	01	Last segment of PIU. (Frame contains no RH).
	10	First segment of PIU.
	11	Only segment of PIU.
	.... ..x.	Primary to secondary indicator.
	x	EFI. Expedited Flow Indicator. 1 = Expedited flow.
1	x...	LU/SSCP bit. Determines where response is routed.
	0	To/From the SSCP.
	1	To/From the LU (the host application).
	.1..	LU/PU bid. Always to/from the LU.
	..xx xxxx	LU Device Address (0-31 for 3271).

RH - Request/Response Header (Only in first I-frame of the segment)

Byte	Bits	Description
0	x...	Request/Response indicator.
	0	Request.
	1	Response.
	.xx.	RU category. (returned as received by CU3270. Always zero on Clear and Psuedo-Bid responses).
	00	Function Management Data.
	01	Network Control.
	10	Data Flow Control.
	11	Session Control.
	...0	Unused. Always 0.
	.... x...	Format Indicator. Sending a request to the Host, set to zero. Sending a response to the Host, as received from the Host.
	.... .1..	Sense Data Included. (no status/sense data if zero).
	11	Unused. Always 11.
1	x...	Definite Response required or sent (see table below).
	.00.	Unused. Always 00.
	...x	Exception Response required or sent (see table below).
	.... 000.	Unused. Always 000.
	x	Pacing Response required or sent (see table below).
2	1...	Begin Bracket (on I-frame following a Psuedo Bid).
	.1..	End Bracket.
	..00	Unused. Always 00.
	.... c...	Code selection.
	0	EBCDIC.
	1	Alternate transmission code (usually ASCII).
	.... .000	Unused. Always 000.

The following table shows what response is appropriate, if any, in the Definate Response (DR), Exception Response (EX) and Pacing (P) bits or the RH.

Request			Response			
DR	EX	P	DR	EX	P	Explanation
1	0	1	1	0	1	Completion of a Read, Write, Copy or print operation by the device. Printers operate in a Definite response with pacing mode only.
			1	1	1	Unsuccessful operation.
1	1	1	0	0	1	Successful Read, Write or Copy.
			1	1	1	Unsuccessful operation.
0	0	1	0	0	1	Sent upon completion regardless of success or failure.
1	0	0	1	0	0	Successful Read, Write or Copy.
			1	1	0	Unsuccessful operation.
1	1	0	No Response			Successful Read, Write or Copy.
			1	1	0	Unsuccessful operation.
0	0	0	No Response			Regardless of success or failure of the operation.

Command (The first byte of the RU, only in the first I-frame of a segment)

Command	EBCDIC	ASCII	Notes
Clear	'A1'	'A1'	Cancel Definite and Pacing responses.
Copy	'F7'	'37'	Followed by 1 byte CCC.
EAU	'6F'	'3F'	Erase All Unprotected.
Erase/Write	'F5'	'35'	Followed by 1 byte WCC.
Psuedo Bid	'F8'	'F8'	Reserve device for host access.
Read Buffer	'F2'	'32'	Read entire buffer.
Read Mod.	'F6'	'36'	Read modified fields from buffer.
Write	'F1'	'31'	Followed by 1 byte WCC.

Sense Information

This information is sent from CU3270 to the host in four bytes immediately following the RH if byte 0, bit 5 is true. The information is as follows:

Byte	Bits	Description
0	1...	Path Error. The device address (TH byte 2, bits 2-7) was invalid.
	.xx. .xxx	Reserved.
	...1	Request Error. The first byte of the RU was not a valid command.
	.... 1...	Request Reject. A Psuedo Bid command or Begin Bracket bit (RH byte 2, bit 0) was sent to a device that has attention pending.
1	xxx. xx..	Reserved.
	...1 ..11	Set whenever Request Reject is set.
2	xxxx ...x	Reserved.
	.... 1...	Device Busy. Device is executing an operation (either it is printing or the application has called C32SB).
	.... .1..	Unit Specify. Not used by CU3270.
	.... ..1.	Device End. A device previously reported unavailable (IR or Busy) is now available. Printers do not report completions with device end.
3	xx..	Reserved.
	..1.	Command Reject. Invalid command received.
	...1	Intervention Required. The application called C32SNR or a previous print operation completed unsuccessfully and the printer is not yet ready.
	.... 1...	Equipment Check. Not used by CU3270.
	.... .1..	Data Check. Not used by CU3270.
	.... ..1.	Control Check. Not used by CU3270.
	1	Operation Check. Received an invalid buffer address or an incomplete or invalid order sequence. Also occurs if the Copy "from" device is unavailable to CU3270.

NP/80 DISK DRIVE CHARACTERISTICS

	67 MB	13.5 MB	26.5 MB	138MB
	8260/8280 F	8280 R	8280 F	8290
# heads	5	1	2	10
# cylinders	823	823	823	823
sectors/track	22	22	22	22
redirect cyls	4	4	4	4
diagnostic cyls	2	2	2	2
redirect secs	440	88	176	880
diagnostic secs	220	44	88	440
MPE sectors	1210 ($4BA)	1210 ($4BA)	1210 ($4BA)	1210 ($4BA)
User sectors	88,660 0255124 $015A54	16,764 040574 $0417C	34,738 0103662	178,530 0534542 $02B962
User bytes (768 bytes/sec)	68,090,088	12,874,752	26,678,784	137,111,040

NP/80 Disk Sector Usage

Region	Phys.Sec.	Use
NPOS	0	NP/80 BOOTSTRAP SECTOR. The ROM code reads this sector; NP(L)GI installs ABOOT here.
NPOS	1-20	NP/80 POSTBOOT PROGRAM. The boot pgm reads it in; NP(L)GI installs APOSTB here. After "glance" diagnostics, boots DE or MPE based on switch 3.
NPOS	21	VOLUME HEADER SECTOR. Built by NPFMTX, fixed by NPVHDR.
NPOS	23	AREA DEFINITION TABLE. Built by ALMCLM. Referenced by MPE upon drive becoming ready.
NPOS	220-439	MULTI-PROGRAMMING EXECUTIVE ("MPE")
NPOS	550-1209	DIAGNOSTIC EXECUTIVE ("DE")
USER	1210.....	USER SECTORS. Size depends on drive type (see chart); virtual areas carved here.
REDIR	after USER	REDIRECT SECTORS. Bad sectors in NPOS & USER regions are redirected here. Because headers are different here, these sectors cannot be read by "ordinary" means.
DIAG	after REDIR	DIAGNOSTIC CYLINDERS. These sectors have regular headers. Used by Diagnostics. First sector also used by BACK80 on backup packs only.

Volume Header Sector - Physical Sector 21

bytes	set by	meaning
0-3	NPFMTX	4-bytes sector header
4-5	NPFMTX NPVHDR	CHECKWORD: If not = $A7D7 then pack was formatted prior to NU01. IF = $A7D7 then pack formatted with explicit volume header.
6-7	NPFMTX NPVHDR	NPFMTX release ID = 0 for NU01/BQ03 1 for NU02/BQ04 or later
8-11	NPFMTX NPVHDR	TOTAL # OF SECTORS on physical pack
12-15	NPFMTX NPVHDR	TOTAL # OF SECTORS MINUS REDIRECTS & DIAGNOSTICS = user area sectors + NPOS region(1210);not used by NU02 utilities
16-17	NPFMTX NPVHDR	# SECTORS/TRACK (always = 22)
18	NPFMTX NPVHDR	Bit 0: 1 = fixed 8280, 0 = removable 8280 or other pack.
19	NPFMTX NPVHDR	# HEADS ON DRIVE: $01=8280 R; $02=40MB 8280 F; $05 = 8260 or 80MB 8280 F; $0A = 8290
20-23	NPFMTX	# SECTORS IN USER AREA (should be 1210 less than figure in bytes 12-15)
24	NPVOL	CHECKBYTE. If = $AE, pack has been initialized by NPVOL.
25	NPVOL	VOLUME SEQUENCE # for backup packs (1st is 1)
26-27	NPCOBK	CHANGE-ONLY BACKUP CHECKWORD. If $CBCB, pack has been initialized for COB and the MPE (BQ04) will flag all sectors it writes. If $EEEE, NPCOBK aborted initialization of pack.
28-29	NPCOBK	C-O-B CURRENT LEVEL -1 if pack initialized for COB.
30-31	NPCOBK	C-O-B CURRENT LEVEL
32-37	NPCOID	C-O-B PACK ID
38-39	NPCOBK	C-O-B BACKUP RESTORE LOCKOUT FLAG. Normally 0; set to $FFFF by NPCOBK when a RESTORE is in process and reset upon successful restore completion. If set, pack contents probably invalid; cleared by NPCOBK in initialization mode.
40-47	-------	<reserved for future use>
48-53	NPVOL	PACK ID for backup packs

NP/80 Console Switch Settings

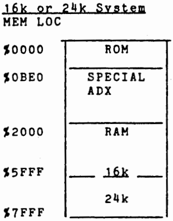
switch	significance	interpreted by
0	"up" -> halt $14/100. Execution may resume after clear halt. Rarely used, intended for patch entry.	MPE
1	"up" -> ignore "booted bit" and stay in ROM on IV/xx reset. (NP/80 reset or power on resets the booted bit so this switch is irrelevant then).	ROM
3	"up" -> load program beginning at 550 ie DE "down" -> load program beginning at 220 ie MPE	Post Boot (APOSTB)
B-F	BOOT DEVICE SELECTION: The position of switch B determines how switches C-F are interpreted. "B" may be considered an operation flag, "C-F" the device unit specification.	ROM

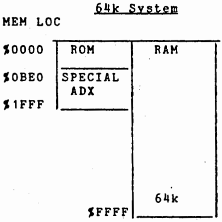
switch B "DOWN" -> BOOT CONTROLLED BY IV/xx AFTER NP/80 RESET
  [C-F] = CPU # (pre-Phoenix Rom code = 2 x CPU#)
  Rom code will loop awaiting cut word from designated IV/xx. Boot and Utility IO cutwords only will be recognized. Upon receipt of boot cut, NP/80 AND IDOS boots will be performed.

switch B "UP" -> UNCONDITIONAL IMMEDIATE NP/80 BOOT UPON RESET
  [C-F] = drive # (eg. [C-F]=$1 -> 8260, 8280R or 8290 @1;
  [C-F]=$8 -> fixed portion 8280@0)
  Rom code will <u>immediately</u> load NP/80 boot program from physical sector 0 of specified drive which in turn loads post-boot and finally MPE (or DE). MPE will then interpret IV/xx cutwords, including boot which now invokes IDOS-only boot sequence.

switches B-F ALL "UP" -> BRANCH TO LOCATION $2070.
  Executes at level 0; rarely used.

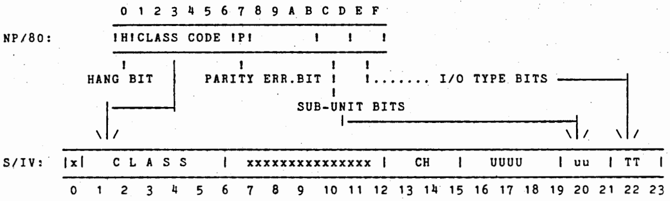
NP/80 MEMORY LAYOUT





$0BE0 $0BE2	ERROR LATCH ADDRESS#1 ——— (failing address) ERROR LATCH ADDRESS#2 (status bits)
$0C00	I/O INTERFACE DEVICE ADDRESS (HANG REGISTER,DISC CONTROL WORDS, IV/70 INTERRUPT REG, DMA COUNT REGISTERS)
$0E00	RESERVED FOR FUTURE USE (non-existant)
$2000	RAM BEGINS (workspace for ROM code)
$2070	Level 0 Interrupt Branch
$2072	Level 1 Interrupt Branch
$2074	Level 2 Interrupt Branch
$2076	Level 3 Interrupt Branch
$2078	Level 4 Interrupt Branch
$207A	Level 5 Interrupt Branch
$207C	Level 6 Interrupt Branch
$207E	register set 1 saved here by a level 0 interrupt
$208A	register set 0 saved here by a level 0 interrupt
$2096	STK saved here by a level 0 interrupt
$2098	MPE BEGINS HERE

IV/xx CUTWORD and NP/80 HANG REGISTER CONTENTS



notes: xxxxx = don't care
CH = channel (standard 2 ie 010)
UUUU = unit (standard 034 ie 0111)
uu = LSB of standard unit address
TT = usual I/O type (only IN and OUT valid)
Classes detailed below.

NPOS CLASS CODES

CLASS	TYPE OF REQUEST	PRECONFIG'D MPE(S)
0	Sector I/O (logical)	ALL
1	Sector I/O (physical)	ALL
2	Resource Allocator	ALMMP2
3	Sector I/O (NPMAM multiple mode)	ALMMP5
4	reserved	———
5	reserved	———
6	MKAM	ALMMP6
7	NPMAM	ALMMP5
8	IV/IV Transfer	ALMMP3
9	Bank manager (future)	none yet
10	Change-only backup	ALL (BQ04)
11-62	reserved	———

FORMATTED DUMP (ALMFMD) SECTION NAMES

SECTION A	MOST RECENT IOB ERRORS LOG
SECTION A.VN	REQUESTS RECEIVED
SECTION B	8260 ERROR OCCURANCES LOG
SECTION C	IV/70 INTERFACE SUBSYSTEM ERROR LOG
SECTION D	8260 MOST RECENT IOB ERRORS LOG
SECTION D.CEL	MOST RECENT COLT REQUESTS ERROR LOG
SECTION D.CTT	VIRTUAL DISK AREA DEFINITION TABLES
SECTION D.UAD	IDOS-UAD TRANSLATION TABLE SIGN ON MATRIX
SECTION E.RA	RESOURCE ALLOCATOR ALLOCATION LIST
SECTION G.BM	BANK MANAGER ERROR LOG
SECTION B.ME	MEMORY ERROR LOG FOR SELF CORRECTING MEMORY
SECTION E.IA	MIDOS ACCESS METHOD FILE ASSIGNMENT LIST
SECTION F	IV70-IV70 TRANSFER RESPONDER TABLE
SECTION G.IA	MIDOS ACCESS METHOD ERROR LOG
SECTION G.IS	MULTI-KEY ACCESS METHOD ERROR LOG
SECTION G.IS.OZ	MULTI-KEY ACCESS METHOD MEMORY STRUCTURES
SECTION G.I7	IV70-IV70 TRANSFER SUBSYSTEM ERROR LOG
SECTION H	OVERLAY FACILITY ERROR LOG
SECTION J	COLT SUBSYSTEM BUFFER PALS
SECTION J.SB	IN-USE SECTOR BUFFERS
SECTION J.SIP	COLT SEQUENCERS IN-PROCESS
SECTION K	RESOURCE ALLOCATOR VN IOB'S
SECTION L	TASK CONTROL BLOCKS (ONE FOR EACH TASK)
SECTION M	IOB QUEUES
SECTION M.BCB	BANK CONTROL BLOCKS
SECTION M.MM	HARDWARE MEMORY MAP
SECTION M.PT	BANK MANAGER PHYSICAL PAGE TABLE
SECTION N.CL	COLT-DISK TASK WORK AREAS (ONE PER TASK)
SECTION N.DV	DV-DISK TASK WORK AREAS (ONE PER TASK)
SECTION N.IA	MIDOS ACCESS METHOD TASK WORK AREAS (ONE PER TASK)

SECTION N.IS	MULTI-KEY ACCESS METHOD TASK WORK AREAS (ONE PER TASK)
SECTION O	DV ICB'S (ONE FOR EACH DRIVE)
SECTION P	NV ICB'S (ONE FOR EACH IV/70)
SECTION Q	8260 ICB'S (ONE FOR EACH DRIVE)
SECTION R	IV70-IV70 TRANSFER TASK WORK AREAS (ONE FOR EACH TASK)
SECTION S	OPEN OVERLAY FAMILY/OVERLAY DESCRIPTORS
SECTION Y	NP/80 REGISTER DUMP
SECTION Z	NP/80 MEMORY DUMP

NP/80 CODED HALTS

If the NP/80 halt light is on, the following steps will retrieve the major and minor coded halts.

1. Record the hex number represented by lights A-F (6 rightmost lights). This is the major code. (Lights 0-9 will be on when NP/80 halts).
2. Place the NP/80 in MANUAL (destroys major code display).
3. Select RPO (1000 in the display select switches). The minor code is now displayed in lights A-F.
4. While you're here you should record CUR STS: set display switches to 1010 In some cases, noted below, you'll want to record values for RPO, RAO, X10, and X20. Display switch settings are listed on the console.
5. To take a formatted NP/80 dump after a halt, see ALMFMD in section A3.

DESCRIPTION OF MAJOR AND MINOR HALT CODES

MAJOR HALT CODE	DESCRIPTION	GENERATED BY
1	GLANCE DIAGNOSTICS ERRORS Major code 1 halts, except the DBRK instruction error (minor $F), can occur when the Glance Diagnostics in the ROM code are run, ie upon NP/80 master reset or upon IV/xx reset with booted bit in NP/80 CUR STS off. The DBRK instruction error can occur when the machine check (level 1) and program check (level 2) interrupt levels are cleared by debreaking. The minor codes associated with major code 1 are detailed in Table 10-1 of NP/80 Installation and Maintenance Manual (NP/80-22-1B). Note: for minor codes $20-22, note contents of RAO & X20; for $23, note X1 in addition.	(ROM CODE)
2	I/O ERRORS in the ROM Code Major code 2 halts indicate a I/O errors which may be the result of a hardware error of a user error involving incorrect console key settings. Minor code meanings depend on vintage of Rom code; see Table 10-2 in NP/80 Installation and Maintenance Manual for details.	(ROM CODE)
5	ERROR DURING BOOT PROGRAM Major code $5 halts are detected by the boot program in sector 0 of the disignated NP/80 boot device disk. Minor code 1: disk error. X10 will contain the address of a seven word error information area. Enter contents of X10 into RP1 and manually display/record contents of 14 successive bytes. Analyse using Table 10-3 in NP/80 Installation and Maintenance Manual (NP/80-22-1B). Minor code 2: the post-boot program is not entirely within the first 22 sectors, ie first track, on disk.	(ABOOT)
6	ERROR DURING POST-BOOT PROGRAM minor meaning 1 Disk error. X1 has address of info area identical to that for major code $5. See above. 2 <not used> 3 Operating system being loaded (MPE) is too large for the existing memory.	(APOSTB)
7	UNDEFINED MEMORY The post-boot program initializes unused memory to $FFF7 which will cause a major code 7 halt upon execution. The implication of a $7 halt is therefore a branch to undefined memory. Note register contents.	

MAJOR HALT CODE		
8	QUICK LOOK DIAGNOSTICS - ERROR IN TYPE 1 INSTRUCTION	(APOSTB)
9	QUICK LOOK DIAGNOSTICS - ERROR IN TYPE 2 INSTRUCTION	(APOSTB)
A	QUICK LOOK DIAGNOSTICS - ERROR IN PRIVILEGED TYPE 2 INST.	(APOSTB)
B	QUICK LOOK DIAGNOSTICS - MACHINE CHECK ERROR	(APOSTB)
C	QUICK LOOK DIAGNOSTICS - PROGRAM CHECK ERROR	(APOSTB)
D	QUICK LOOK DIAGNOSTICS - EXTERNAL INTERRUPT ERROR	(APOSTB)
E	QUICK LOOK DIAGNOSTICS - ERROR IN DBRK, ACLIC, OR LEVEL 6	(APOSTB)
F	QUICK LOOK DIAGNOSTICS - UNEXPECTED INTERRUPT	(APOSTB)
10	MACHINE CHECK Display CUR STS register indicators in bits 2-4. bit meaning / cause 2 Read error - parity error on the bus on data coming into CPU; DEFINITE HARDWARE. 3 Bus parity - parity on bus on data going out of CPU; DEFINITE HARDWARE. 4 Bus timeout - CPU has put an address on the bus whhc is t acknowledged; can be caused by software addressing nonexistent RAM or a nonexistant I/O device OR by hardware.	(MPE)
11	PROGRAM CHECK A program check is generated if background attempts to access protected memory, to use an odd adress with a word instruction or to execute a privileged instruction. Note that the HLT instruction is privileged and that MPE tasks code, which executes in background, will HLT upon error conditions, causing an immediate level 2 program check. Under these circumstances, it is imperative to dump the NP/80 using ALMFMD or to use the STK register to manually find the original halt codes as stacked in memory. A program check is also generated if an interrupt is received at interrupt level 3. This is a hardware error as there are currently no devices assigned to level 3.	(MPE)
12	DISK SUBSYSTEM ERRORS 10 op code in internal table is not 1 or 2 11 TRPEND is already on 13 supervisor interrupted lev for new IOB when ICB not idle 14 supv interrupted interface to start IOB but IOB addr not zero 16 supv signalled interface to start IOB but IOB addr was zero 17 timeout timer started when it was already on 18 invalid op code in internal request tab 19 excessive xmission length 20 invalid opcode in internal request tab 21 bad keep-trying flag in SS sequence 22 bad command byte in SS sequence 23 bad CDCRN subsystem parameter 24 header-only operations not supported 25 ran out of carriers 26 bad checkword in carrier free-stack 27 problem getting carrier from free-queue	(MPE)

MAJOR HALT CODE		
12	28	problem putting carrier into free-stack
	29	bad chkwd adding carrier to free-stack
	2A	problem starting driver execution of carrier
	2B	problem moving carrier to comp-queue
	2C	problem strting driver execution of carrier
	2D	problem moving carrier to comp-queue
	2E	problem completing carrier
	2F	problem queuing a carrier
	30	aatempt to issue Release, Change, Verify operation with a stolen buffer
	31	bad buffer pal checkword
	32	buffer does not currently have a sector
	33	bad checkword in "WAITING FOR BUFFERS" list
	34	bad workarea checkword in waiting work area
	35	bad workarea checkword or state code
	36	bad 8260 ICB checkword
	37	attempt to issue Release, No Change operation with a stolen buffer
	38	attempt to issue Release, Change operation with a stolen buffer
	39	bad IOB or carrier checkword
	3A	bad checkword "OLDEST/NEWEST" buffer pal linked list
	3B	invalid buffer pool pal checkword
	3C	unexpected "KNOW-ITS-THERE" flag set
	3D	bad buffer pal oldest/newest linked list checkword
	3E	bad carrier checkword on 8260 IOB being returned
	3F	buffer pal with buffer being returned is already linked into some list
	40	bad sequencer workarea checkword
	41	bad buffer pool pal checkword
	42	buffer contains a sector it should not have
	43	bad checkword in "WAITING-FOR-BUFFER" queue
	44	bad workarea checkword in "WAITING-FOR-BUFFER" queue
	45	workarea is waiting for this sector and the KIT flag is alread set
	46	bad carrier or sequencer workarea checkword
	47	bad buffer pool pal checkword
	48	tried to assign a buffer that is already in use
	49	bad colt IOB checkword
	4A	bad sequencer workarea checkword
	4B	bad carrier or sequencer workarea checkword
	4C	bad buffer pool pal checkword
	4D	tried to deassign a buffer that is not in use
	4E	bad colt IOB checkword
	4D	tried to deassign a buffer that is assigned to another request
	50	tried to set "CLFREB", freed buffer indicator, & it was alread set
	51	bad buffer pool pal checkword, free buffer
	52	bad checkword in "WAITING FOR BUFFER" LL
	53	bad sequencer workarea checkword
	54	error in "WAITING FOR BUFFER" LL scan or in PREF-KIT flag logic
	55	bad checkword in sequencer workarea
	56	bad X2 input returned from CLWAIT
	58	bad sequencer workarea checkword
	59	impossible sector counter
	5A	unexpected sector type was detected
	5B	bad sequencer workarea checkword
	5C	bad Colt IOB checkword

MAJOR HALT CODE		
12	5D	bad buffer checkword(s)
	5E	bad buffer pal checkword
	5F	bank-to-bank sector move was requested
	60	"Release, No Change" failed
	61	bad sector counter
	62	"Release, No Change" in error portion failed
	63	impossible IOB.STT in Colt IOB
	64	impossible IOB.STT in NV IOB
	66	Being-Read flag is already set
	67	Being-Read and Being-Written flags both set
	68	Being-Read and Being-Written flags both off
	69	unexpected sector type detected
	6A	unexpected sector type detected
	6B	bad Colt IOB checkword
	6C	bad checkword buffer pal "OLDEST/NEWEST" linked list descriptor
	6D	bad buffer pal checkword
	6E	unsuccessful "RELEASE, NO CHANGE"
	6F	after-complete-subr called w/invalid carrier address
	70	invalid drive # input to CDPKRY
	71	slot area checkword bad
	72	slot area checkword bad
	73	slot area checkword bad
	74	loop counter (TWA.TDWSCT) > limit
	75	bad checkword on pal from free-queue
	76	bad pal checkword after A-completion
	77	bad pal checkword after B-completion
	78	bad pal checkword after C-completion
	79	bad checkword on incoming IOB
	7A	COLT IOB and it had a buffer
	7B	impossible IOB.CLIOOP on C-OPERATION
	7C	bad checkword in "IN-PROCESS" LL element descriptor
	7D	bad checkword in sequencer workarea
	7E	bad buf-pal ckwd. KIT flag is garbage in sequencer workarea
	7F	asynchronous read done but counter said no reads outstanding
	80	bad VN IOB checkword (NV IOB.SIIOAV points to VN)
	81	" "
	82	bad subroutine or type 1 trace checkword
	83	logger code is zero
	84	bad opcode
	85	bad IOB checkword in L8LIOB option
	86	IOB successful. only unsuccessful should be logged
	87	rightmost 3 bits of error cd not zero
	8C	bad ECB count in record descriptor contention block
	8D	bad ECB count in index set descriptor
	8E	invalid byte count in index sector
	8F	invalid byte count in selection descr or index s read by SECIO
	90	invalid SECIO subroutine checkword
	91	two consecutive read requests
	92	two consecutive write requests
	93	invalid $COLT op code
	94	invalid zone descriptor checkword
	95	invalid zoneset descriptor checkword
	96	bad checkbyte in zone descriptor
	97	bad checkword in zone descriptor
	98	a re-entrant call was attempted
	99	" "
	9A	the TWA stack addr wasn't zero on entry to D4KSEA
	9B	encountered non-root or last root sector w/no dummy entry

MAJOR HALT CODE		
12	9C	invalid #-of-bytes in the sector bytes-in-use field
	9D	bad checkword in key path stack
	9E	displacement within sector field > # sector bytes in use
	9F	Forward pointer of sector chain to be extended is not zero
	A0	invalid zoneset descriptor checkword
	A1	to-pointer reached end of index sector before from-pointer
	A2	invalid indset descriptor checkword
	A3	init call while input routine already in use
	A4	init sub D4RIWI not called first
	A5	init call while input routine already in use
	A6	init sub D4RIWI not called first
	A7	info sector no longer in sector buffer
	A8	attempting to deallocate a zoneset which contains index sets
	A9	invalid checkword in zone or zoneset descriptor
	AA	inval checkword in indset descriptor, field descr, or key descr
	AB	bad checkword in selection descriptor or VNIOB
	AC	bad checkword in selection descriptor
	AD	try to execute nonexistent SECIO subr
	AE	additional SECIO subr already allocated
	AF	additional SECIO subr doesn't exits
	B0	bad field descriptor checkword
	B1	bad record or selection descriptor checkword
	B2	bad test subroutine checkword
	B3	bad record descriptor checkword
	B4	an invalid key byte supplied by caller in TWA.D4TIOP
	B5	invalid stack entry checkword
	B6	invalid pointer entry checkbyte
	B7	last data record overflows sector
	B8	bad linked list ckwrd in rec or selection descr
	B9	invalid rec descr - bad data record secad displ
	BA	-not used
	BC	try to release addtl SECIO rtne while it still has a secbuf
	BD	# of bytes to be moved > 768
	BE	program error - tried to promote >1 sector's worth pointer ent
	BF	D4TNEP > 2 sectors worth of pointer entries
	C2	problem queueing a carrier
	C3	bad checkwords
	C4	TCB inconsistencies when activating
	C5	IOB.CDIODR indicates fixed CMD req but drive not CMD
	C6	returned slot area that is not in use
	D6	dummy file used for MPE's that must be less than 64K
	D7	COLT buffer add called when multi-bank not configured
	DE	addtl SECIO subr still configured at end of request
	E0	IAUPAT called to allocate a sector that's already allocated
	E1	IARDRL called out of appropriate sequence
	E2	Invalid assignment list entry checkword
	E3	file assigned by IADRFD not found in assignment list
	E4	invalid drive descriptor checkword
	E5	invalid assignment list entry checkword
	E6	invalid lockout list entry
	E7	invalid drive descriptor checkword
	E8	invalid op code in req tab or not enough free memory
	E9	lockout list entry not found
	EA-FF	reserved for MIDOS

MAJOR HALT CODE		
13	OTHER SUBSYSTEM ERRORS (BANK MANAGER, MKAM)	(MPE)
	0	this MPE config'ed for multi-bank, NP80 IOP doesn't support
	1	unable to initialize memory map
	2	memory missing or parity at $F000
	3	bad bank or logical page > 15
	4	bad physical page number >255
	5	physical page mapped but not even aloocated
	6	try to unmap page already unmapped
	7	bad bank or logical page number >15
	8	invalid COLT opcode
	9	two consecutive read requests
	A	add buffer req issued after read req
	B	bank mgr active flag not set and new req list not empty
	C	State code doesn't indicate that bank mgr waiting for VN IOB
	E	bad bank control block linked list
	F	physical page to be mapped was already mapped
	10	tried to map an unallocated page
	11	tried to map a page allocated exclusively to another user
	12	tried to map a phys page to two logical pages in same bank
	13	bank generation number not valid
	14	bad logger subroutine checkword
	15	bad logger subr or type 1 trace chkwrd
	16	logger code = 0
	17	bad logger op code
	18	bad IOB checkword
	19	IOB was successful
	1A	not enough zero-bits on right of code
	1B	completed IOB is not a COLT or a NV IOB
	1C	unable to find completed VN IOB in new request table
	1D	returned from wait req issued to permanently disable bank mgr
	1E	NP/80 doesn't support MVL instruction
	1F	see RA, word after minor code, and PMPJA9 for meaning
	20	machine check, program check in system not supporting >64K
	21	invalid call to machine check interrupt processor
	22	" "
	23	the task that had machine task was executing in bank 0
	24	no task was active when SVMCPC called
	25	currently active task is not in bank to be deallocated
	C0	bad logger subroutine checkword
	C1	IOB.SIIOAV doesn't point to VN IOB
	C3	key block size is too big
	C4	no selection descr's attached to record
	C5	bad level of access code in seldes
	C6	bad record descriptor checkword
	C7	after deleting index sector, find key path stack empty
	C8	bad key path stack checkword
	C9	bad key path stack checkword
	CA	invalid record descriptor checkword
	CB	the selection descriptor linked list being accessed is empty
	CC	bad sel desc checkword during access level contention check
	CD	record descriptor was not deleted
	CE	invalid checkbyte in zoneset or indset descriptor - X1 points
	CF	invalid checkword in seldes or recdes
	D0	no selection descriptor attached to record descriptor
	D1	invalid selection descriptor checkword
	D2	tried to get another block when previous one not returned
	D3	tried to return a block when none allocated
	D4	a SECIO error while trying to return a stolen COLT buffer
	D5	addr of block issued doesn't match addr of one being returned

MAJOR HALT CODE			
13	D6	return from 1-second $WAIT was not due to timeout	
	D7	invalid checkbyte in zoneset or indset descriptor	
	D8	bad record descr linked list checkword	
	D9	a record descriptor exists with no selection descr's attached	
	DA	sel descriptors still exist during indset deallocation	
	DB	active selections exist in indset being deleted (not in use)	
	DC	invalid byte count during try to browse forward	
	DD	invalid checkword in selection descr	
	DE-FF	reserved for MKAM	
14	TASK CONTROL SUBSYSTEM		(MPE)
	0	procedural halt	
	1	bad TCB checkword	
	2	task already active	
	3	bad TCB checkword	
	6	bad EBS checkword	
	7	dequeued TCB did not have its in-queue flag on	
	9	bad TCB or ECB checkword	
	11	bad ECB checkword	
	12	ECB queue is empty (ie nothing to signal)	
	14	bad EBC checkword	
	16	bad initialization subroutine descriptor checkword	
	1A	bad TCB area limits	
	1B	bad TCB checkword	
	20	bad checkword on IOB after-complete subroutine	
	21	bad IOB or ICB checkword	
	22	IOB is not $F1 (waiting to be processed by driver)	
	23	bad IOB or ICB checkword	
	24	IOB is not $F2 (being processed by driver)	
	25	not enough memory for a single sector buffer	
	26	ran out of free carriers	
	27	attempt to move carrier from an empty queue	
	28	bad carrier or destination queue checkbyte	
	29	bad carrier checkword	
	2A	bad carrier checkword	
	2B	bad after-complete subroutine checkword	
	2C	no room for a single carrier	
	2D	driver queue empty	
	2E	bad checkwords	
	30	timed ECB queue is empty	
	31	bad ECB checkword in ECB queue	
	32	no TCB's queued to ECB in ECB queue	
	33	bad TCB checkword in ECB queue	
	34	bad return from "TSDQUE", no TCB's queued to ECB	
	35	problem putting carrier in completed carriers queue	
	36	bad checkword in $REQ1 operation	
	37	ran out of cells in $REQ1 operation	
	38	bad $REQ1 IOB checkbyte. R1IOOP byte 1 does not equal $97.	
	39	after the DMA, advanced DMA address does not agree with what it should be	
	3A	Header-only operation attempted in MPE with those operations disabled as of BQ03-A.	

MAJOR HALT CODE			
15		DISPATCHER ERRORS	(MPE)
	0	unsupported SVC code	
	1	a high level interrupt signalled the supervisor but no ICB requires servicing	
	2	bad ICB service subrtn checkword	
	3	bad timer cell service subrtn checkword	
	4	bad SVC service subrtn checkword	
	21	bad IOB checkword	
	22	bad IOB checkword	
	23	bad IOB checkword	
	24	bad ICB checkword	
	25	bad ICB/ICB check	
	26	IOB not $F3 (completed by driver and waiting to be dequeued)	
	27	bad IOB after-complete subroutine checkword	
	28	bad ICB checkword	
	29	bad TCELL checkword (timed-VN)	
	2A	timer already running	
16		INTERFACE SUBSYSTEM ERRORS	(MPE)
	2	timer already running	
	3	squirt length given in squirt is greater than 48 bytes	
	6	supervisor firing error; bad NVSCI	
	8	zero or odd DMA length given	
	9	bad ICB address	
	A	no IOB to process when there should be	
	B	timed-VN, no IOB when there should be	
	C	timed-VN, bad TCELL checkword	
	D	NV IOB cannot be processed because the interface subsytem is not configured for that particular IOB.NVIORT (request type)	
	E	bad checkbyte in IOB.NVIO7A	
	F	bad checkbyte in IOB.LTIO7A	
17		UNDEFINED - This major code is used for halt conditions to which minor codes are not assigned. It is IMPERATIVE to note ALL register contents upon a $17 halt, especially RPO, and to correctly identify the MPE (configuration and thruno). The MPE should be dumped using ALMFMD.	(MPE)
20		GENERAL	(MPE)
	1	integrity check failed	
	2	integrity check failed	
	3	type-1 linked list insertions: bad chechword or other integrity failure	
	4	type-1 linked list deletions: bad chechword or other integrity failure	
	5	bad checkword in free list	
	6	bad checkword in returned memory block	
	7	bad checkword in free memory area	

MAJOR HALT CODE			
21		DV SUBSYSTEM	(MPE)
	1	bad DV ICB address	
	2	bad type argument	
	3	bad DV ICB checkword	
	4	bad PAL checkword	
	5	DVRAWP inconsistency (read-after-write)	
	6	R A W buffer being freed is not the one originally allocated by the DV subsystem	
	7	bad IOB checkword	
	8	DVLOGE called for successful sub-IOB	
	9	bad PAL chekword	
	A	bad PAL chekword	
	10	read complete on initial activation	
	11	write complete on initial activation	
	12	just freed PAL on initial activation	
	13	read-complete while waiting for new IOB	
	14	write-complete while waiting for new IOB	
	15	just-freed PAL while waiting for IOB	
	16	write-complete while waiting for seel	
	17	just freed PAL while waiting for seel	
	18	"READ" entered when no more needed	
	19	ICB (DVICYR) went negative	
	1A	ICB (DVICBR) went negative	
	1B	ICB (DVICBW) went negative	
	1C	ICB (DVICYW) went negative	
	1D	waiting for PAL inconsistency	
	1E	No IOB when there should be one	
22		BACKGROUND TASKS	(MPE)
	1	an unsupported mode value given	
	2	lost request table	
	3	inconsistency detected	
23		COMMUNICATIONS - does not exist. No minor codes.	(MPE)
24		IV/xx-IV/xx TRANSFER SUBSYSTEM	
	1	bad pal checkword before read	(MPE)
	2	bad pal checkword after read	
	3	bad pal checkword after write	
	4	responder table search problem	
	5	bad NV ICB checkword	
	6	bad linked-list checkword	
	7	bad linked-list checkword	
	8	bad busy-flag value	
	9	bad busy-flag value	
	A	some inconsistency	
25		OTHER	(MPE)
	1	MPE size exceeds NP/80 memory size	

Communications Services

$XFER - Release BR02

Initialization - $XINIT

$XINIT is called once, before $XOPEN is called for the first time.

```
BRM    $XINIT    Initialize $XFER
 DCN   3         ..Number of parameters
 PZE   IOID      ..Address of the IOID table
 PZE   CLUAT     ..Address of the CLUAT table
 BRA   Error     ..Error return
 ---             ..Normal return
```

Error Codes: 1 - $XFER is already initialized

Opening a controller - $XOPEN

```
BRM    $XOPEN    Open a logical controller
 DCN   5         ..Number of parameters
 PZE   LUN       ..Address of the logical unit number
 PZE   ACW       ..Address of the ACW
 PZE   EXIT.1    ..Address of the Level 1 exit routine
 PZE   MAPPING   ..Address of the mapping word
 BRA   Error     ..Error return
 ---             ..Normal return
```

Error Codes:
1 - $XINIT not called
2 - Invalid logical unit number was called
3 - The logical unit is already opened

Mapping word:
Bit 0 - 0 = Buffers are not mapped, 1 = Buffers mapped.
Bit 1 - 0 = Use primary window, 1 = Use current window (MFE only).

Closing a controller - $XCLOSE

```
BRM    $XCLOSE   Close a logical controller
 DCN   2         ..Number of parameters
 PZE   LUN       ..Address of the logical unit number
 BRA   Error     ..Error return
 ---             ..Normal return
```

Error Codes:
1 - $XINIT not called
2 - Invalid logical unit number was called

Changing logging on a controller - $XLOG

```
BRM     $XLOG      Change logging on a controller
 DCN    3          ..Number of parameters
 PZE    LUN        ..Address of the logical unit number
 BRA    LOGVAL     ..Address of the log value
 BRA    Error      ..Error return
---                ..Normal return
```

Error Codes:
- 2 - An invalid logical unit number was specified
- 3 - The logical unit referenced is not opened

Log Value:
- 0 = Turn logging off for this logical unit.
- 1 = Turn logging on for this logical unit.

Transfer a table to the controller - $XFER

```
BRM     $XFER      Transfer table to the controller
 DCN    2          ..Number of parameters
 PZE    TABLE      ..Address of the request table
 BRA    Error      ..Error return
---                ..Normal return
```

Error Codes:
- 2 - An invalid logical unit number was specified
- 3 - The logical unit referenced is not opened
- 5 - IRQ timeout
- 6 - One word transfered, request table queued
- 7 - Previous buffer pending, request table queued

Request Table:
- Word 0, Bit 0, 0 = Request, 1 = Response
  - Bit 2, 1 = I/O of table to controller completed
- Word 1, Bits 16-23, Logical unit number
- Word 2, Bits 0-7, 0=Write, 1=Read, 2=Contol, 3=Status
- Word 4, Address of the request table.
- Word 5, Post-processing routine address
- Word 6, Address of the Associated Buffer, if any.

ELOG

Start logging - LOGON

```
BRM     LOGON      Initialize ELOG
 DCN    FIRST      ..Address of the first block of the log
 DCN    CTLWORD    ..Control word
---                ..Return
```

Return Code:
- RA = 0, LOGON was successful
- RA = -1, LOGON was not successful

First block:
- Bits 1-8, Log area window number
- Bits 9-23, Address of the first block of the log area.

Control Word:
- Bit 0, 0 = Wrap log, 1 = Do not wrap log.
- Bit 1, 0 = Start log, 1 = Do not start log.
- Bit 2, 0 = Log in window, 1 = Log cross window.
- Bits 9-23, 0 = log is chained, non-0 = length of contiguous log area.

Stop Logging - LOGOFF

```
            BRM     LOGOFF     Terminate ELOG
            ---                Return
```

Chained Log Block Format

The first word of each block of the chain is in the following format:

Bits 0-7, Blocksize of this block (in words) divided by 16
Bits 9-23, Pointer to the next block. (0 - last block in the chain).

Log Information Table Pointer, Location 7

Bit 0, 0 = Log has not wrapped, 1 = Log has wrapped.
Bits 1-8, Window number of LIT and log.
Bits 9-23, Address of the LIT.

Log Information Table Format

Word	Bits	Description
0	9-23	Pointer to the first log block
1	9-23	Pointer to the current log block
2	9-23	Pointer to current position in the log block
3	9-23	Length of contiguous area (in words). Zero means chained.
4	9-23	Negative number of words left in the current block.

Communications Utilities

DTCOMM, DTCOMF and COMDMP - Software Release BS04

These utilities are used to access the communications comtroller RAM. There are two versions, DTCOMM is the IDOS version, DTCOMF is the DKOS version and COMDMP is the MFE version.

A. An 8437/8460 load file may be changed by loading the file into the 8437/8460, changing the apropriate memory values and dumping the contents of the memory to the same file from which it was loaded. When using this method you must not exit DTCOMM before writing the updated memory contents back to a disc file.

B. DTCOMM aids the user in debugging programs which run on the 8437/8460 communications controller. With the processor, one may load, display to the screen, update and dump 8437/8460 memory. The product has been designed with the occasional user in mind and is very straightforward in its use.

C. Note: DTCOMM has a single threaded logic which handles only one parameter per input line. It uses OPTION to read parameters but it will only execute one operation at a time.

The following JCL has been implemented:

```
// DTCOMM
/ INPUT             = Filename @ Drive.
/ DISPLAY           = Byte Address.
/ UPDATE            = Value @ Byte Address.
/ OUTPUT            = PRINT @ Starting Byte Address.
                      -OR-
                      Filename @ Drive.
/ LOGICAL UNIT      = Logical Unit Number.
/ PATTERN           = Pattern Value @ Starting Byte Address.
//
```

// DTCOMM will load the processor and display a menu of commands on the screen. The release identifier (BS02) displays with the menu.

/ LOGICAL UNIT will cause DTCOMM to access up to four communications controllers. The default is controller zero. All operations are performed on the controller specified by this command until a subsequent LOGICAL UNIT command is entered. The logical unit to physical channel/unit address is as follows:

Logical Unit		Channel/Unit	
0	=	1	30
1	=	1	34
2	=	1	20
3	=	1	24

/ INPUT will load the file named from the drive specified and write the file into the controller RAM. The file is written into memory starting at location zero. DTCOMM will read compressed files (3 bytes/word as output by LOAD65) or uncompressed files (1 byte/word, right justified, as output by UASM65) and load them correctly into the controller RAM.

/ DISPLAY will display on the screen 256 bytes of the communication controller memory starting with the address specified. The controller registers will also be displayed (RP, A, X, Y, STACK and STATUS).

/ UPDATE will change the byte specified to the value specified. Then DTCOMM will display on the screen 256 bytes of controller memory starting with the location changed. After modifying a byte, updates to sequentially following bytes need not specify a byte address.

/ OUTPUT will output the contents of the communication controller memory to the printer (if PRINT is specified) or to the named file on the specified drive. If PRINT is specified, the output will start with the controller address specified and will be in hexadecimal format. Large portions of memory containing the same value will be compressed on the printout. If the specified output is a file name, the output will be a contiguous file with three bytes per word. The output file may be reloaded using DTCOMM.

/ PATTERN will fill the communication controller memory with the specified 16-bit pattern starting with the specified address through the end of memory.

DTCOMM obtains the screen size from the COMM region. The IDOS processor SCREEN can be used to set the screen size parameters. All parameters are optional. Entering a "// " will cause a return to IDOS. While DTCOMM is accessing the controller memory, the controller is executing in the ROM code, thus creating a basically stable state. The controller will be released to resume its interrupted activity when the user returns to DTCOMM. All values and addresses can be in octal, decimal or hexadecimal. Drive numbers may be in octal or decimal. Preceed octal numbers with a zero (0) and hexadecimal numbers with a percent sign (%).

TRACE and COMTRA - Communications Line Trace Analyzer - Release BS04-A

TRACE is the IDOS version and COMTRA is the MFE version of the trace analysis program. This utility reads the log area (created by ELOG) from a dump file. See the ELOG section in Communications Services (BR02) for the format of the log.

// TRACE Start execution of TRACE. A menu of functions displays.

/ INPUT = Filename @ Drive.
The input file is a IV/xx dump file. If no input file is specified, TRACE will attempt to open "DUMP47" or "CKPT" on drive 0. If neither or both of these files are found, an error message appears on the screen.

/ WINDOW = Window number.
If location 7 in the dump is invalid but the window that the trace is in is known, this parameter may be entered to force TRACE to find the correct window. The default is the window stored in location 7 of the dump file.

/ XLIT = Log Information Table Address.
If location 7 in the dump is invalid but the Log Information Table address is known, this parameter may be entered to force TRACE to find the LIT table. The default is the window stored in location 7 of the dump file.

/ LOGICAL UNIT = Logical controller number.
This specifies which logical controller TRACE is supposed to display/print data from. If this is specified, the type is not necessary.

/ TYPE = Access method type.
This specifies which access method TRACE is supposed to display/print data from. If this is specified, the Logical controller number is not necessary. Access method types follow;

/ C SIMULATOR TRACE FILE = Filename @ Drive.
The C file is a simulator trace file. If no name is entered, the default is TRCFIL @ 0.

Type	Access Method
C	NTP150
CBE	CU3270/8437 BSC EBCDIC
CBA	CU3270/8437 BSC ASCII
CSE	CU3270/8437 SDLC EBCDIC
CSA	CU3270/8437 SDLC ASCII
L	Local channel adaptor (7073)
LA	LAM/8437 ASCII
LE	LAM/8437 EBCDIC
M	MLAM/8437 EBCDIC
MLA	MLATMS/8460 (Multi-line Async)
RE	3770 EBCDIC
V	Vision/8436 (3270)
W	Remote Terminal

/ INPUT. To exit TRACE.
//

After enterint the above options, hit cursor return and a menu of function keys will appear. These are listed below.

Key	Function
1	Advance the Trace display 1 line.
H	Advance the Trace display a half screen.
F	Advance the Trace display a full screen.
P	Start or Stop printing the trace (toggle).
R	Restart the display.
S	Turn on or off the full trace (toggle, on at startup).
ATTN	Enter option parameters.
L	Go to the end of the file (contiguous log area only).
B	Back up one sector (contiguous log area only).
E	Exit to IDOS.

CONFIGURATOR

The configurator is executed by entering the following into SYSIN:

```
// C32OPT
```

The SIMED file (S32CFG) which is used by the configurator contains selectedmacros from the following list:

```
CONFIG
    DEVICES=n (screens plus printers)
    SCREENS=n
    PRINTERS=n (number of buffered printers)
    SIZE=80X24, 80X12, 40X24, 40X12
    ENVIRONMENT=IDOS, MFE, DKOS
    CONTROLLER=8436, 8437, NOCOMM
    LPBASE=030, 0
    CPBASE=0, 1, 2, ...
    WIDGET=YES, NO
DEVICE   (Should be listed once for each device)
    DEV.ADDR=n (n - number of devices)
    TYPE=TUBE, BUFPRT, CPYPRT, DUMMY
    TUBE.ADDR=n (address with respect to series IV cabling)
    TRANSLATE=TBXXXX
    PRINT.TYPE=CHARACTER, LINE
    PRINT.ADDR=n (n= increment to be added to printer base)
    INTERCEPT=NO, YES
COMM
    CONTROLLER.UNIT=033 & up for 8436, 0-7 for 8437
    CU.ADDR=40, 20, 01 & up
    DISCIPLINE=BISYNC, SDLC
    MODE=NRZ, NRZI
    MODEM.SPEED=LOW, HIGH
    LOG.TYPE=0-4
    CHARSET=EBCDIC, ASCIIA, ASCIIB
    BUFFERS=n (n defaults to 17 for 8436, 5 plus terminals for 8437)
    LOAD.8437=NO, YES
    TB2COL=NO,YES
SB3270
    ATFIELD=NO, YES
NOBEEP
CKPT
    DISK=8250, 8230, 8240, 8260, 8280
    C=YES, NO
LPRINT
    PRINT.ADDR=n (logical device number)
STFWD
    FILE=XXXXXX
ANAME
    FILE=S32DBX
DBGMEM
DBGTRD
DBGTRM
DBGMFE
ENDCFG
     PRINTER.LINES=nn  (number of lines per page)
```

MINGEN

A32MND or (A32MNF for 8250's) can be used to change the following items in the simulator load module:

```
                Control unit address
                Lines per page
                Device logical unit number
                Initial intercept flag
                Physical unit address of any printer
                Number of devices
           For 8437's only:
                Line discipline
                Character set
                Modem speed (dual speed modems only)
```

JCL FOR SIMULATOR UTILITIES

CKPTXX

```
// CKPTXX - allocate checkpoint or trace file and load simulator
/B=banks
/C=checkpoint file name
/T=size for trace file (TRCFIL)
/L=name of load module
//
```

A32FMD - print formatted dump

```
// A32FMD
/C=checkpoint file name @ drive
//
```

AF3270 - create local disk data file

```
// AF3270
/O=filename @ drive
/V=screensize
/T=number of terminals
/F=number of formats
/S=total images
//
```

ACPAFC - validate Store and Forward file

```
// ACPAFC
/I=filename @ drive
```

DEBUG ROUTINES

The following debug options are available under BV03:

A - Toggle attributes
B - Return to operating system
C - Take a checkpoint dump
D - Disable any dynamic display
E - Toggle the logging state
F - Move trace display by lines
G - Move trace display by disc sectors
H - Move to start or end of trace
I - Select memory logging
J - Select disc logging
K - Toggle the full system trace on or off
L - Set lines per page
M - Dynamic display of given memory locations
N - not implemented
O - Obtain printer under MFE
P - Password
Q - not implemented
R - Release printer under MFE
S - See others screens
T - not implemented
U - not implemented
V - Store a value into memory
W - not implemented
X - not implemented
Y - Change local screen print device
Z - Local screen print

10700 N. DeAnza Blvd., Cupertino, CA. 95014
(408) 255-0900