SY33-8571-7
S 370/S
4300-40

Systems

DOS/VSE Handbook

IBM

Preface

This manual references to information contained in this manual is provided as a DOS/VSE serviceability aid and is, therefore, a summary of other DOS/VSE documentation. These manuals are intended for use by persons involved in program support.

This manual does not contain information on DOS/VSE Advanced Functions. This information represent licensed material and must be ordered separatly.

It will be distributed as System Library Supplement which logically fit into this manual.

The volume contains following information:

Volume 1, SY33-8571:
Chapter I : General Information
II : DOS/VSE General Information
III: DOS/VSE IOCS (General, SAM, DAM, ISAM)
IV: DOS/VSE Supervisor Control Blocks and Areas
V : DOS/VSE Service Aids

If there is any discrepancy between the information contained in this manual and the DOS/VSE optional programming material (e.g., PLMs and listings), the latter is assumed to be correct.

Eighth Edition (February, 1979)

This is a major revision of and obsoletes, SY33-8571-6. This edition applies to the Disk Operating System/Virtual Storage Extended (DOS/VSE) and to all subsequent versions and releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *System /370 Bibliography*, GC20-0001, for the editions that are applicable and current.

Request for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A handbook-sized binder, FE Part Number 453559, may be purchased from IBM. Customers may order it through their IBM marketing representative. IBM personnel should order it as an FE part from Mechanicsburg.

This manual has been prepared by WT-DP/CE Technical Operations, 7030 Boeblingen, Germany.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be sent to the above address. Comments become the property of IBM.

TABLE OF CONTENTS (. . . . Cont'd)

Chapter IV

Chapter V DOS/VSE Service Aids

TABLE OF CONTENTS

TABLE OF CONTENTS (. . . . Cont'd)

Chapter III

Chapter IV DOS/VSE Supervisor Control and Areas

CHAPTER I
GENERAL INFORMATION

MACHINE INSTRUCTIONS

Name	Mnemonic	Op Code	Format	Operands
Add (c)	AR	1A	RR	R1, R2
Add (c)	A	5A	RX	R1, D2(X2,B2)
Add Decimal (c)	AP	FA	SS	D1(L1,B1),D2(L2,B2)
Add Halfword (c)	AH	4A	RX	R1,D2(X2,B2)
Add Logical (c)	ALR	1E	RR	R1,R2
Add Logical (c)	AL	5E	RX	R1,D2(X2,B2)
AND (c)	NR	14	RR	R1,R2
AND (c)	N	54	RX	R1,D2(X2,B2)
AND (c)	NI	94	SI	D1(B1),12
AND (c)	NC	D4	SS	D1(L,B1),D2(B2)
Branch and Link	BALR	05	RR	R1,R2
Branch and Link	BAL	45	RX	R1,D2(X2,B2)
Branch on Condition	BCR	07	RR	M1,R2
Branch on Condition	BC	47	RX	M1,D2(X2,B2)
Branch on Count	BCTR	06	RR	R1,R2
Branch on Count	BCT	46	RX	R1,D2(X2,B2)
Branch on Index High	BXH	86	RS	R1,R3,D2(B2)
Branch on Index Low or Equal	BXLE	87	RS	R1,R3,D2(B2)
Clear I/O	CIRIO	9D01	S	D2(B2)
Clear Page	CLRP	B215	S	
Compare (c)	CR	19	RR	R1,R2
Compare (c)	C	59	RX	R1,D2(X2,B2)
Compare and Swap	CS	BA	RS	R1,R3,D2,(B2)
Compare Decimal (c)	CP	F9	SS	D1(L1,B1),D2(L2,B2)
Compare Double and Swap	CDS	BB	RS	R1,R3,D2,(B2)
Compare (c)	CR	19	RR	R1,R2
Compare (c)	C	59	RX	R1,D2(X2,B2)
Compare Decimal (c)	CP	F9	SS	D1(L1,B1),D2(L2,B2)
Compare Halfword (c)	CH	49	RX	R1,D2(X2,B2)
Compare Logical (c)	CLR	15	RR	R1,R2
Compare Logical (c)	CL	55	RX	R1,D2(X2,B2)
Compare Logical (c)	CLC	D5	SS	D1,(L,B1),D2(B2)
Compare Logical (c)	CLI	95	SI	D1(B1),I2
Compare Logical Characters under Mask (c)	CLM	BD	RS	R1,M3,D2(B2)
Compare Logical Long (c)	CLCL	0F	RR	R1,R2
Connect Page	CTP	B0	RS	R1,M3,D2(B2)
Convert to Binary	CVB	4F	RX	R1,D2(X2,B2)
Convert to Decimal	CVD	4E	RX	R1,D2(X2,B2)
Deconfigure Page	DEP	B21B	S	
Diagnose (p)		83	SI	
Disconnect Page	DCTP	B21C	S	
Divide	DR	ID	RR	R1,R2
Divide	D	5D	RX	R1,D2(X2,B2)
Divide Decimal	DP	FD	SS	D1(L1,B1),D2(L2,B2)
Edit (c)	ED	DE	SS	D1(L,B1),D2(B2)
Edit and Mark (c)	EDMK	DF	SS	D1(L,B1),D2(B2)
Exclusive OR (c)	XR	17	RR	R1,R2
Exclusive OR (c)	X	57	RX	R1,D2(X2,B2)
Exclusive OR (c)	XI	97	SI	D1(B1),I2
Exclusive OR (c)	XC	D7	SS	D1(L,B1),D2(B2)
Execute	EX	44	RX	R1,D2(X2,B2)
Halt I/O (c,p)	HIO	9E00	S	D1(B1)
Halt Device (c,p)	HDV	9E01	S	D1(B1)
Insert Character	IC	43	RX	R1,D2(X2,B2)
Insert Characters under Mask (c)	ICM	BF	RS	R1,M3,D2(B2)
Insert Page Bits	IPB	B4	S	
Insert Storage Key (p)	ISK	09	RR	R1,R2
Insert PSW Key	IPK	B20B	S	
Load	LR	18	RR	R1,R2
Load	L	58	RX	R1,D2(X2,B2)
Load Address	LA	41	RX	R1,D2(X2,B2)
Load and Test (c)	LTR	12	RR	R1,R2
Load Complement (c)	LCR	13	RR	R1,R2
Load Control (p)	LCTL	B7	RS	R1,R3,D2(B2)
Load Frame Index	LFI	B8	RS	R1,M3,D2(B2)
Load Halfword	LH	48	RX	R1,D2(X2,B2)
Load Multiple	LM	98	RS	R1,R3,D2(B2)
Load Negative (c)	LNR	11	RR	R1,R2
Load Positive (c)	LPR	10	RR	R1,R2

Name	Mnemonic	Op Code	Format	Operands
Load PSW (n,p)	LPSW	82	SI	D1(B1)
Load Real Address (c,p)	LRA	B1	RX	R1,D2(X2,B2)
Monitor Call (m)	MC	AF	SI	D1(B1),I2
Make Addressable	MAD	B21D	S	
Make Unaddressable	MUN	B21E	S	
Move	MVI	92	SI	D1(B1),I2
Move	MVC	D2	SS	D1(L,B1),D2(B2)
Move Long (c)	MVCL	0E	RR	R1,R2
Move Numerics	MVN	D1	SS	D1(L,B1),D2(B2)
Move with Offset	MVO	F1	SS	D1(L1,B1),D2(L2,B2)
Move Zones	MVZ	D3	SS	D1(L,B1),D2(B2)
Multiply	MR	1C	RR	R1,R2
Multiply	M	5C	RX	R1,D2(X2,B2)
Multiply Decimal	MP	FC	SS	D1(L1,B1),D2(L2,B2)
Multiply Halfword	MH	4C	RX	R1,D2(X2,B2)
OR (c)	OR	16	RR	R1,R2
OR (c)	O	56	RX	R1,D2(X2,B2)
OR (c)	OI	96	SI	D1(B1),I2
OR (c)	OC	D6	SS	D1(L, B1),D2(B2)
Pack	PACK	F2	SS	D1(L1,B1),D2(L2,B2)
Purge Table (p)	PTLB	B20D	S	
Reset Reference Bit (c,p)	RRB	B213	S	D1(B1)
Retrieve Status and Page	RSP	D8	SS	D1(L1,B1),D2(L2,B2)
Set Clock (c,p)	SCK	B204	S	D1(B1)
Set Clock Comparator (p)	SCKC	B206	S	D1(B1)
Set CPU Timer (p)	SPT	B208	S	D1(B1)
Set Page Bits	SPB	B5	RS	R1,D2(B2)
Set Program Mask (n)	SPM	04	RR	R1
Set Storage Key (p)	SSK	08	RR	R1,R2
Set PSW Key from Address	SPKA	B20A	S	
Set System Mask (p)	SSM	80	SI	D1(B1)
Shift and Round Decimal (c)	SRP	F0	SS	D1(L1,B1),D2(B2),I3
Shift Left Double (c)	SLDA	8F	RS	R1,D2(B2)
Shift Left Double Logical	SLDL	8D	RS	R1,D2(B2)
Shift Left Single (c)	SLA	8B	RS	R1,D2(B2)
Shift Left Single Logical	SLL	89	RS	R1,D2(B2)
Shift Right Double (c)	SRDA	8E	RS	R1,D2(B2)
Shift Right Double Logical	SRDL	8C	RS	R1,D2(B2)
Shift Right Single (c)	SRA	8A	RS	R1,D2(B2)
Shift Right Single Logical	SRL	88	RS	R1,D2(B2)
Start I/O (c,p)	SIO	9C00	S	D1(B1)
Start I/O Fast Release (c,p)	SIOF	9C01	S	D1(B1)
Store	ST	50	RX	R1,D2(X2,B2)
Store Capacity Counts	STCAP	B21F	S	
Store Channel ID (c,p)	STIDC	B203	S	D1(B1)
Store Character	STC	42	RX	R1,D2(X2,B2)
Store Characters under Mask	STCM	BE	RS	R1,M3,D2(B2)
Store Clock (c)	STCK	B205	S	D1(B1)
Store Clock Comparator (p)	STCKC	B207	S	D1(B1)
Store Control (p)	STCTL	B6	RS	R1,R3,D2(B2)
Store CPU ID (p)	STIDP	B202	S	D1(B1)
Store CPU Timer (p)	STPT	B209	S	D1(B1)
Store Halfword	STH	40	RX	R1,D2(X2,B2)
Store Multiple	STM	90	RS	R1,R3,D2(B2)
Store then AND System Mask (p)	STNSM	AC	SI	D1(B1)I2
Store then OR System Mask (p)	STOSM	AD	SI	D1(B1),I2
Subtract (c)	SR	1B	RR	R1,R2
Subtract (c)	S	5B	RX	R1,D2(X2,B2)
Subtract Decimal (c)	SP	FB	SS	D1(L1,B1),D2(L2,B2)
Subtract Halfword (c)	SH	4B	RX	R1,D2(X2,B2)
Subtract Logical (c)	SLR	1F	RR	R1,R2
Subtract Logical (c)	SL	5F	RX	R1,D2(X2,B2)
Supervisor Call	SVC	0A	RR	I
Test and Set (c)	TS	93	SI	D1(B1)
Test Channel (c,p)	TCH	9F	SI	D1(B1)
Test I/O (c,p)	TIO	9D	SI	D1(B1)
Test under Mask (c)	TM	91	SI	D1(B1),I2
Translate	TR	DC	SS	D1(L,B1),D2(B2)
Translate and Test (c)	TRT	DD	SS	D1(L,B1),D2(B2)
Unpack	UNPK	F3	SS	D1(L1,B1),D2(L2,B2)
Zero and Add Decimal (c)	ZAP	F8	SS	D1(L1,B1),D2(L2,B2)

Floating Point Instructions

Name	Mnemonic	Op Code	Format	Operands
Add Normalized, Extended (c,x)	AXR	36	RR	R1,R2
Add Normalized, Long (c)	ADR	2A	RR	R1,R2
Add Normalized, Long (c)	AD	6A	RX	R1,D2(X2,B2)
Add Normalized, Short (c)	AER	3A	RR	R1,R2
Add Normalized, Short (c)	AE	7A	RX	R1,D2(X2,B2)
Add Unnormalized, Long (c)	AWR	2E	RR	R1,R2
Add Unnormalized, Long (c)	AW	6E	RX	R1,D2(X2,B2)
Add Unnormalized, Short (c)	AUR	3E	RR	R1,R2
Add Unnormalized, Short (c)	AU	7E	RX	R1,D2(X2,B2)
Compare, Long (c)	CDR	29	RR	R1,R2
Compare, Long (c)	CD	69	RX	R1,D2(X2,B2)
Compare, Short (c)	CER	39	RR	R1,R2
Compare, Short (c)	CE	79	RX	R1,D2(X2,B2)
Divide, Long	DDR	2D	RR	R1,R2
Divide, Long	DD	6D	RX	R1,D2(X2,B2)
Divide, Short	DER	3D	RR	R1,R2
Divide, Short	DE	7D	RX	R1,D2(X2,B2)
Halve, Long	HDR	24	RR	R1,R2
Halve, Short	HER	34	RR	R1,R2
Load and Test, Long (c)	LTDR	22	RR	R1,R2
Load and Test, Short (c)	LTER	32	RR	R1,R2
Load Complement, Long (c)	LCDR	23	RR	R1,R2
Load Complement, Short (c)	LCER	33	RR	R1,R2
Load, Long	LDR	28	RR	R1,R2
Load, Long	LD	68	RX	R1,D2(X2,B2)
Load Negative, Long (c)	LNDR	21	RR	R1,R2
Load Negative, Short (c)	LNER	31	RR	R1,R2
Load Positive, Long (c)	LPDR	20	RR	R1,R2
Load Positive, Short (c)	LPER	30	RR	R1,R2
Load Rounded, Extended t Long (x)	LRDR	25	RR	R1,R2
Load Rounded, Long to Short (x)	LRER	35	RR	R1,R2
Load, Short	LER	38	RR	R1,R2
Load, Short	LE	78	RX	R1,D2(X2,B2)
Multiply, Extended (x)	MXR	26	RR	R1,R2
Multiply, Long	MDR	2C	RR	R1,R2
Multiply, Long	MD	6C	RX	R1,D2(X2,B2)
Multiply, Long/Extended (x)	MXDR	27	RR	R1,R2
Multiply, Long/Extended (x)	MXD	67	RX	R1,D2(X2,B2)
Multiply, Short	MER	3C	RR	R1,R2
Multiply, Short	ME	7C	RX	R1,D2(X2,B2)
Store, Long	STD	60	RX	R1,D2(X2,B2)
Store, Short	STE	70	RX	R1,D2(X2,B2)
Subtract Normalized, Ext'd (c,x)	SXR	37	RR	R1,R2
Subtract Normalized, Long (c)	SDR	2B	RR	R1,R2
Subtract Normalized, Long (c)	SD	6B	RX	R1,D2(X2,B2)
Subtract Normalized, Short (c)	SER	3B	RR	R1,R2
Subtract Normalized, Short (c)	SE	7B	RX	R1,D2(X2,B2)
Subtract Unnormalized, Long (c)	SWR	2F	RR	R1,R2
Subtract Unnormalized, Long (c)	SW	6F	RX	R1,D2(X2,B2)
Subtract Unnormalized, Short (c)	SUR	3F	RR	R1,R2
Subtract Unnormalized, Short (c)	SU	7F	RX	R1,D2(X2,B2)

(a) Direct Control Feature
(c) Condition Code is set
(m) Monitoring Feature
(n) New Condition Code is loaded
(p) Privileged Instruction
(x) Extended precision floating point feature

EXTENDED MNEMONIC INSTRUCTION CODES

GENERAL

Extended Code		Machine Instruction		Meaning
B	D2(X2,B2)	BC 15,	D2(X2,B2)	Branch Unconditionally
BR	R2	BCR 15,	R2	Branch Unconditionally
NOP	D2(X2,B2)	BC 0,	D2(X2,B2)	No Operation
NOPR	R2	BCR 0,	R2	No Operation (RR)

AFTER COMPARE INSTRUCTIONS (A:B)

BH	D2(X2,B2)	BC 2,	D2(X2,B2)	Branch on A High
BL	D2(X2,B2)	BC 4,	D2(X2,B2)	Branch on A Low
BE	D2(X2,B2)	BC 8,	D2(X2,B2)	Branch on A equal B
BNH	D2(X2,B2)	BC 13,	D2(X2,B2)	Branch on A not High
BNL	D2(X2,B2)	BC 11,	D2(X2,B2)	Branch on A not Low
BNE	D2(X2,B2)	BC 7,	D2(X2,B2)	Branch on A not Equal B

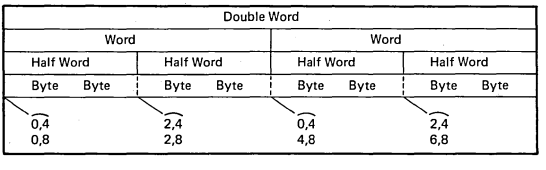
AFTER ARITHMATIC INSTRUCTIONS

BO	D2(X2,B2)	BC 1,	D2(X2,B2)	Branch on Overflow
BP	D2(X2,B2)	BC 2,	D2(X2,B2)	Branch on Plus
BM	D2(X2,B2)	BC 4,	D2(X2,B2)	Branch on Minus
BZ	D2(X2,B2)	BC 8,	D2(X2,B2)	Branch on Zero
BNP	D2(X2,B2)	BC 13,	D2(X2,B2)	Branch on not Plus
BNM	D2(X2,B2)	BC 11,	D2(X2,B2)	Branch on not Minus
BNZ	D2(X2,B2)	BC 7,	D2(X2,B2)	Branch on not Zero

AFTER TEST UNDER MASK INSTRUCTIONS

BO	D2(X2,B2)	BC 1,	D2(X2,B2)	Branch if Ones
BM	D2(X2,B2)	BC 4,	D2(X2,B2)	Branch if Mixed
BZ	D2(X2,B2)	BC 8,	D2(X2,B2)	Branch if Zero's
BNO	D2(X2,B2)	BC 14,	D2(X2,B2)	Branch if not Ones

CNOP ALIGNMENT



EDIT AND EDMK PATTERN CHARACTERS (In Hex)

20 – Digit selector
21 – Start of significance
22 – Field separator
40 – Blank
4B – Period
5B – Dollar sign
5C – Aterisk
6B – Comma
C3D9 – CR

ASSEMBLER INSTRUCTIONS

Function	Mnemonic	Meaning
Data Definition	DC	Define Constant
	DS	Define Storage
	CCW	Define Channel Command Word
Program Sectioning and Linking	START	Start Assembly
	CSECT	Identify Control Section
	DSECT	Identify Dummy Section
	COM	Identify blank common Control Section
	ENTRY	Identify Entry Point Symbol
	EXTRN	Identify External Symbol
	WXTRN	Identify weak External Symbol
Base Register Assignment	USING	Use Base Address Register
	DROP	Drop Base Address Register
Control of listings	TITLE	Identify Assembly Output
	EJECT	Start new Page
	SPACE	Space Listing
	PRINT	Print Optional Data
Program Control	ICTL	Input Format Control
	ISEQ	Input Sequence Checking
	PUNCH	Punch a Card
	REPRO	Reproduce following Card
	ORG	Set Location Counter
	EQU	Equate Symbol
	LTORG	Begin Literal Pool
	CNOP	Conditional No-Operation
	COPY	Copy predefined Source Coding
	END	End Assembly
Macro Definition	MACRO	Macro Definition Header
	MNOTE	Request for Error Message
	MEXIT	Macro Definition Exit
	MEND	Macro Definition Trailer
Conditional Assembly	ACTR	Conditional Assembly Loop Counter
	AGO	Unconditional Branch
	AIF	Conditional Branch
	ANOP	Assembly No-Operation
	GBLA	Define global SETA Symbol
	GBLB	Define global SETB Symbol
	GBLC	Define global SETC Symbol
	LCLA	Define local SETA Symbol
	LCLB	Define local SETB Symbol
	LCLC	Define local SETC Symbol
	SETA	Set Arithmetic Variable Symbol
	SETB	Set Arithmetic
	SETC	Set Character Variable Symbol

SUMMARY OF CONSTANTS

Type	Implied Length Bytes	Alignment	Format	Truncation Padding
C	—	Byte	Characters	Right
X	—	Byte	Hexadecimal Digits	Left
B	—	Byte	Binary Digits	Left
F	4	Word	Fixed-point Binary	Left
H	2	Halfword	Fixed-point Binary	Left
E	4	Word	Short Floating-point	Right
D	8	Doubleword	Long Floating-point	Right
L	16	Doubleword	Extended Floating-point	Right
P	—	Byte	Packed Decimal	Left
Z	—	Byte	Zoned Decimal	Left
A	4	Word	Value of Address	Left
Y	2	Halfword	Value of Address	Left
S	2	Halfword	Address in Base-Displacement form	—
V	4	Word	Externally defined Address Value	Left

CONDITION CODES

Condition Code Setting	0	1	2	3
Mask Bit Position	8	4	2	1
Floating Point Arithmetic				
Add Normalized S/L/E	zero	<zero	>zero	–
Add Unnormalized S/L	zero	<zero	>zero	–
Compare S/L (A:B)	equal	A low	A high	–
Load and Test S/L	zero	<zero	>zero	–
Load Complement S/L	zero	<zero	>zero	–
Load Negative S/L	zero	<zero	–	–
Load Positive S/L	zero	–	>zero	–
Subtract Normalized S/L/E	zero	<zero	>zero	–
Subtract Unnormalized S/L	zero	<zero	>zero	–
Fixed Point and Decimal Arithmetic				
Add H/F/Dec.	zero	<zero	>zero	overflow
Add Logical	zero, no carry	not zero, no carry	zero, carry	not zero, carry
Compare H/F/Dec. (A:B)	equal	A low	A high	–
Compare and Swap/Double	equal	not equal	–	–
Load and Test	zero	<zero	>zero	–
Load Complement	zero	<zero	>zero	overflow
Load Negative	zero	<zero	–	–
Load Positive	zero	–	>zero	overflow
Shift and Round Decimal	zero	<zero	>zero	overflow
Shift Left Single/Double	zero	<zero	>zero	overflow
Shift Right Single/Double	zero	<zero	>zero	–
Subtract H/F/Dec.	zero	<zero	>zero	overflow
Subtract Logical	–	not zero, no carry	zero , carry	not zero, carry
Zero and Add	zero	<zero	>zero	overflow
Logical Operations				
AND	zero	not zero	–	–
Compare Logical (A:B)	equal	A low	A high	–
Edit	zero	<zero	>zero	–
Edit and Mark	zero	<zero	>zero	–
Exclusive OR	zero	not zero	–	–
Insert Characters under Mask	all zero	1st bit one	1st bit zero	–
Move Long (A:B)	equal	A low	A high	overlap
OR	zero	not zero	–	–
Test under Mask	zero	mixed	–	one
Translate and Test	zero	incomplete	complete	–
Input/Output Operations				
Clear I/O	no oper in	CSW stored	chan busy	not oper
Halt I/O, Halt Device	interruption pending	CSW stored	See Prin Op	not oper
Start I/O, SIOF	started	CSW stored	busy	not oper
Store Channel ID	ID stored	CSW stored	ID not stored	not oper
Test I/O	available	CSW stored	busy	not oper
Test Channel	available	interruption pending	burst mode	not oper

CONDITION CODES (. . . Cont'd)

Condition Code Setting Mask Bit Position	0 8	1 4	2 2	3 1
Miscellaneous Operations				
Set Clock	set	secure	—	not oper
Store Clock	set	not set	error	not oper
Test and set	zero	one	—	—
Load Real Address	Translation available	Segment table entry invalid	Page table entry invalid	Segment- or Page table length violation
Reset Reference Bit	Ref bit zero, change bit zero	Ref bit zero, change bit one	Ref bit one, change bit zero	Ref bit one, change bit one
Make Addressable	Page was connected	Page was already addressable	—	—
Make Unaddressable	Page was addressable	Page was already connected	—	—
Retrieve Status and Page	Save information is valid	—	—	Save information is invalid
Set Page Bits	Ref bit 0, C bit 0	R bit 0, C bit 1	R bit 1, C bit 0	R bit 1, C bit 1
Load Frame Index	Index returned, page is addressable	Index returned, page is connected	Index not returned, page is disconnected	Index not returned, address is invalid
Connect Page	Successful, index returned	Page was already connected, index returned	Not successful, index not returned	—
Disconnect Page	Page was connected	Page was already disconnected	—	—

CODE TRANSLATION TABLE

Dec	Hex	Instruction (RR)	Graphics and Controls BCDIC	EBCDIC	(1) ASCII	7-Track Tape BCDIC (2)	Card Code	Binary
0	00			NUL	NUL		12-0-1-8-9	0000 0000
1	01			SOH	SOH		12-1-9	0000 0001
2	02			STX	STX		12-2-9	0000 0010
3	03			ETX	ETX		12-3-9	0000 0011
4	04	SPM		PF	EOT		12-4-9	0000 0100
5	05	BALR		HT	ENO		12-5-9	0000 0101
6	06	BCTR		LC	ACK		12-6-9	0000 0110
7	07	BCR		DEL	BEL		12-7-9	0000 0111
8	08	SSK			BS		12-8-9	0000 1000
9	09	ISK			HT		12-1-8-9	0000 1001
10	0A	SVC		SMM	LF		12-2-8-9	0000 1010
11	0B			VT	VT		12-3-8-9	0000 1011
12	0C			FF	FF		12-4-8-9	0000 1100
13	0D			CR	CR		12-5-8-9	0000 1101
14	0E	MVCL		SO	SO		12-6-8-9	0000 1110
15	0F	CLCL		SI	SI		12-7-8-9	0000 1111
16	10	LPR		DLE	DLE		12-11-1-8-9	0001 0000
17	11	LNR		DC1	DC1		11-1-9	0001 0001
18	12	LTR		DC2	DC2		11-2-9	0001 0010
19	13	LCR		TM	DC3		11-3-9	0001 0011
20	14	NR		RES	DC4		11-4-9	0001 0100
21	15	CLR		NL	NAK		11-5-9	0001 0101
22	16	OR		BS	SYN		11-6-9	0001 0110
23	17	XR		IL	ETB		11-7-9	0001 0111
24	18	LR		CAN	CAN		11-8-9	0001 1000
25	19	CR		EM	EM		11-1-8-9	0001 1001
26	1A	AR		CC	SUB		11-2-8-9	0001 1010
27	1B	SR		CU1	ESC		11-3-8-9	0001 1011
28	1C	MR		IFS	FS		11-4-8-9	0001 1100
29	1D	DR		IGS	GS		11-5-8-9	0001 1101
30	1E	ALR		IRS	RS		11-6-8-9	0001 1110
31	1F	SLR		IUS	US		11-7-8-9	0001 1111
32	20	LPDR		DS	SP		11-0-1-8-9	0010 0000
33	21	LNDR		SOS	!		0-1-9	0010 0001
34	22	LTDR		FS	"		0-2-9	0010 0010
35	23	LCDR			#		0-3-9	0010 0011
36	24	HDR		BYP	$		0-4-9	0010 0100
37	25	LRDR		LF	%		0-5-9	0010 0101
38	26	MXR		ETB	&		0-6-9	0010 0110
39	27	MXDR		ESC	'		0-7-9	0010 0111
40	28	LDR			(		0-8-9	0010 1000
41	29	CDR			)		0-1-8-9	0010 1001
42	2A	ADR		SM	*		0-2-8-9	0010 1010
43	2B	SDR		CU2	+		0-3-8-9	0010 1011
44	2C	MDR			,		0-4-8-9	0010 1100
45	2D	DDR		ENQ	-		0-5-8-9	0010 1101
46	2E	AWR		ACK	.		0-6-8-9	0010 1110
47	2F	SWR		BEL	/		0-7-8-9	0010 1111
48	30	LPER			0		12-11-0-1-8-9	0011 0000
49	31	LNER			1		1-9	0011 0001
50	32	LTER		SYN	2		2-9	0011 0010
51	33	LCER			3		3-9	0011 0011
52	34	HER		PN	4		4-9	0011 0100
53	35	LRER		RS	5		5-9	0011 0101
54	36	AXR		UC	6		6-9	0011 0110
55	37	SXR		EOT	7		7-9	0011 0111
56	38	LER			8		8-9	0011 1000
57	39	CER			9		1-8-9	0011 1001
58	3A	AER			:		2-8-9	0011 1010
59	3B	SER		CU3	;		3-8-9	0011 1011
60	3C	MER		DC4	<		4-8-9	0011 1100
61	3D	DER		NAK	=		5-8-9	0011 1101
62	3E	AUR			>		6-8-9	0011 1110
63	3F	SUR		SUB	?		7-8-9	0011 1111

1 EBCDIC graphics shown are standard bit pattern assignment. For specific print train/chain: See printer manual.
2 Add C (Check bit) for odd or even parity as needed, exept as noted.
3 For even parity use CA

Dec	Hex	Instruction (RR)	Graphics and Controls BCDIC	EBCDIC (1)	ASCII	7-Track Tape BCDIC (2)	Card Code	Binary
64	40	STH		SP	@	(3)	no punches	0100 0000
65	41	LA			A		12-0-1-9	0100 0001
66	42	STC			B		12-0-2-9	0100 0010
67	43	IC			C		12-0-3-9	0100 0011
68	44	EX			D		12-0-4-9	0100 0100
69	45	BAL			E		12-0-5-9	0100 0101
70	46	BCT			F		12-0-6-9	0100 0110
71	47	BC			G		12-0-7-9	0100 0111
72	48	LH			H		12-0-8-9	0100 1000
73	49	CH			I		12-1-8	0100 1001
74	4A	AH		¢	J		12-2-8	0100 1010
75	4B	SH		.	K	B A 8 2 1	12-3-8	0100 1011
76	4C	MH	¤	<	L	B A 8 4	12-4-8	0100 1100
77	4D		(	(	M	B A 8 4 1	12-5-8	0100 1101
78	4E	CVD	<	+	N	B A 8 4 2	12-6-8	0100 1110
79	4F	CVB	+	|	O	B A 8 4 2 1	12-7-8	0100 1111
80	50	ST	& +	&	P	B A	12	0101 0000
81	51				Q		12-11-1-9	0101 0001
82	52				R		12-11-2-9	0101 0010
83	53				S		12-11-3-9	0101 0011
84	54	N			T		12-11-4-9	0101 0100
85	55	CL			U		12-11-5-9	0101 0101
86	56	O			V		12-11-6-9	0101 0110
87	57	X			W		12-11-7-9	0101 0111
88	58	L			X		12-11-8-9	0101 1000
89	59	C			Y		11-1-8	0101 1001
90	5A	A		!	Z		11-2-8	0101 1010
91	5B	S	$	$	[	B 8 2 1	11-3-8	0101 1011
92	5C	M	*	*	\	B 8 4	11-4-8	0101 1100
93	5D	D	]	)	]	B 8 4 1	11-5-8	0101 1101
94	5E	AL	;	;	¬ ^	B 8 4 2	11-6-8	0101 1110
95	5F	SL	△	¬	-	B 8 4 2 1	11-7-8	0101 1111
96	60	STD	-	-	'	B	11	0110 0000
97	61		/	/	a	A 1	0-1	0110 0001
98	62				b		11-0-2-9	0110 0010
99	63				c		11-0-3-9	0110 0011
100	64				d		11-0-4-9	0110 0100
101	65				e		11-0-5-9	0110 0101
102	66				f		11-0-6-9	0110 0110
103	67	MXD			g		11-0-7-9	0110 0111
104	68	LD			h		11-0-8-9	0110 1000
105	69	CD			i		0-1-8	0110 1001
106	6A	AD		¦	j		12-11	0110 1010
107	6B	SD	.	.	k	A 8 2 1	0-3-8	0110 1011
108	6C	MD	%	%	l	A 8 4	0-4-8	0110 1100
109	6D	DD	Y	—	m	A 8 4 1	0-5-8	0110 1101
110	6E	AW	\	>	n	A 8 4 2	0-6-8	0110 1110
111	6F	SW	≠	?	o	A 8 4 2 1	0-7-8	0110 1111
112	70	STE			p		12-11-0	0111 0000
113	71				q		12-11-0-1-9	0111 0001
114	72				r		12-11-0-2-9	0111 0010
115	73				s		12-11-0-3-9	0111 0011
116	74				t		12-11-0-4-9	0111 0100
117	75				u		12-11-0-5-9	0111 0101
118	76				v		12-11-0-6-9	0111 0110
119	77				w		12-11-0-7-9	0111 0111
120	78	LE			x		12-11-0-8-9	0111 1000
121	79	CE			y		1-8	0111 1001
122	7A	AE		:	z	A	2-8	0111 1010
123	7B	SE	# =	#	{	8 2 1	3-8	0111 1011
124	7C	ME	@'	@	|	8 4	4-8	0111 1100
125	7D	DE	:	'	}	8 4 1	5-8	0111 1101
126	7E	AU	>	=	~	8 4 2	6-8	0111 1110
127	7F	SU	V	"	DEL	8 4 2 1	7-8	0111 1111

1 EBCDIC graphics shown are standard bit pattern assignments. For specific print train/chain: See printer manual.

2 Add C (check bit) for odd or even parity as needed, exept as noted.

CODE TRANSLATION TABLE (. . . . Cont'd)

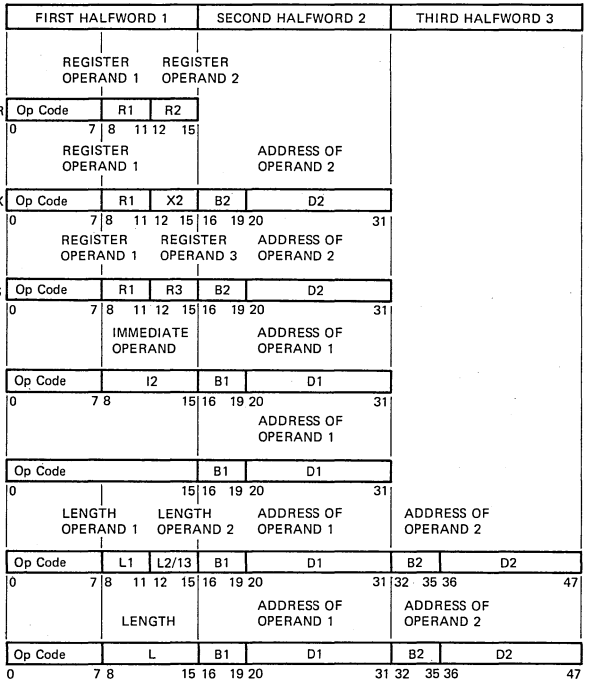
Dec	Hex	Instruction (RS, SI, S)	Graphics and Controls BCDIC	EBCDIC (1)	ASCII	7-Track Tape BCDIC (2)	Card Code	Binary
128	80	SSM					12-0-1-8	1000 0000
129	81			a			12-0-1	1000 0001
130	82	LPSW		b			12-0-2	1000 0010
131	83	Diagnose		c			12-0-3	1000 0011
132	84			d			12-0-4	1000 0100
133	85			e			12-0-5	1000 0101
134	86	BXH		f			12-0-6	1000 0110
135	87	BXLE		g			12-0-7	1000 0111
136	88	SRL		h			12-0-8	1000 1000
137	89	SLL		i			12-0-9	1000 1001
138	8A	SRA					12-0-2-8	1000 1010
139	8B	SLA					12-0-3-8	1000 1011
140	8C	SRDL					12-0-4-8	1000 1100
141	8D	SLDL					12-0-5-8	1000 1101
142	8E	SRDA					12-0-6-8	1000 1110
143	8F	SLDA					12-0-7-8	1000 1111
144	90	STM					12-11-1-8	1001 0000
145	91	TM		j			12-11-1	1001 0001
146	92	MVI		k			12-11-2	1001 0010
147	93	TS		l			12-11-3	1001 0011
148	94	NI		m			12-11-4	1001 0100
149	95	CLI		n			12-11-5	1001 0101
150	96	OI		o			12-11-6	1001 0110
151	97	XI		p			12-11-7	1001 0111
152	98	LM		q			12-11-8	1001 1000
153	99			r			12-11-9	1001 1001
154	9A						12-11-2-8	1001 1010
155	9B						12-11-3-8	1001 1011
156	9C	SIO, SIOF					12-11-4-8	1001 1100
157	9D	TIO, CLRIO					12-11-5-8	1001 1101
158	9E	HIO, HDV					12-11-6-8	1001 1110
159	9F	TCH					12-11-7-8	1001 1111
160	A0						11-0-1-8	1010 0000
161	A1			~			11-0-1	1010 0001
162	A2			s			11-0-2	1010 0010
163	A3			t			11-0-3	1010 0011
164	A4			u			11-0-4	1010 0100
165	A5			v			11-0-5	1010 0101
166	A6			w			11-0-6	1010 0110
167	A7			x			11-0-7	1010 0111
168	A8			y			11-0-8	1010 1000
169	A9			z			11-0-9	1010 1001
170	AA						11-0-2-8	1010 1010
171	AB						11-0-3-8	1010 1011
172	AC	STNSM					11-0-4-8	1010 1100
173	AD	STOSM					11-0-5-8	1010 1101
174	AE						11-0-6-8	1010 1110
175	AF	MC					11-0-7-8	1010 1111
176	B0	CTP					12-11-0-1-8	1011 0000
177	B1						12-11-0-1	1011 0001
178	B2	See below					12-11-0-2	1011 0010
179	B3						12-11-0-3	1011 0011
180	B4	IPB					12-11-0-4	1011 0100
181	B5	SPB					12-11-0-5	1011 0101
182	B6	STCTL					12-11-0-6	1011 0110
183	B7	LCTL					12-11-0-7	1011 0111
184	B8	LFI					12-11-0-8	1011 1000
185	B9						12-11-0-9	1011 1001
186	BA	CS					12-11-0-2-8	1011 1010
187	BB	CDS					12-11-0-3-8	1011 1011
188	BC						12-11-0-4-8	1011 1100
189	BD	CLM					12-11-0-5-8	1011 1101
190	BE	STCM					12-11-0-6-8	1011 1110
191	BF	ICM					12-11-0-7-8	1011 1111

OP Code:

B202 - STIDP
B203 - STIDC
B204 - SCK
B205 - STCK
B206 - SCKC
B207 - STCKC
B208 - SPT
B209 - STPT
B213 - RRB
B20A - SPKA
B20B - IPK
B215 - CLRP
B21B - DECP
B21C - DCTP
B21D - MAD
B21E - MUN
B21F - STCAP

CODE TRANSLATION TABLE (. . . . Cont'd)

Dec	Hex	Instruction (SS)	Graphics and Controls			7-Track Tape								Card Code	Binary
			BCDIC	EBCDIC (1)	ASCII	BCDIC (2)									
192	C0		?	{		B	A	8				2		12-0	1100 0000
193	C1		A	A		B	A						1	12-1	1100 0001
194	C2		B	B		B	A					2		12-2	1100 0010
195	C3		C	C		B	A					2	1	12-3	1100 0011
196	C4		D	D		B	A		4					12-4	1100 0100
197	C5		E	E		B	A		4				1	12-5	1100 0101
198	C6		F	F		B	A		4	2				12-6	1100 0110
199	C7		G	G		B	A		4	2	1			12-7	1100 0111
200	C8		H	H		B	A	8						12-8	1100 1000
201	C9		I	I		B	A	8					1	12-9	1100 1001
202	CA													12-0-2-8-9	1100 1010
203	CB													12-0-3-8-9	1100 1011
204	CC			⌐										12-0-4-8-9	1100 1100
205	CD													12-0-5-8-9	1100 1101
206	CE			⊬										12-0-6-8-9	1100 1110
207	CF													12-0-7-8-9	1100 1111
208	D0		!	}		B		8		2				11-0	1101 0000
209	D1	MVN	J	J		B							1	11-1	1101 0001
210	D2	MVC	K	K		B				2				11-2	1101 0010
211	D3	MVZ	L	L		B				2	1			11-3	1101 0011
212	D4	NC	M	M		B			4					11-4	1101 0100
213	D5	CLC	N	N		B			4				1	11-5	1101 0101
214	D6	OC	O	O		B			4	2				11-6	1101 0110
215	D7	XC	P	P		B			4	2	1			11-7	1101 0111
216	D8	RSP	Q	Q		B		8						11-8	1101 1000
217	D9		R	R		B		8					1	11-9	1101 1001
218	DA													12-11-2-8-9	1101 1010
219	DB													12-11-3-8-9	1101 1011
220	DC	TR												12-11-4-8-9	1101 1100
221	DD	TRT												12-11-5-8-9	1101 1101
222	DE	ED												12-11-6-8-9	1101 1110
223	DF	EDMK												12-11-7-8-9	1101 1111
224	E0		‡	\			A	8			2			0-2-8	1110 0000
225	E1													11-0-1-9	1110 0001
226	E2		S	S			A				2			0-2	1110 0010
227	E3		T	T			A				2	1		0-3	1110 0011
228	E4		U	U			A		4					0-4	1110 0100
229	E5		V	V			A		4					0-5	1110 0101
230	E6		W	W			A		4	2				0-6	1110 0110
231	E7		X	X			A		4	2	1			0-7	1110 0111
232	E8		Y	Y			A	8						0-8	1110 1000
233	E9		Z	Z			A	8					1	0-9	1110 1001
234	EA													11-0-2-8-9	1110 1010
235	EB													11-0-3-8-9	1110 1011
236	EC			⊣										11-0-4-8-9	1110 1100
237	ED													11-0-5-8-9	1110 1101
238	EE													11-0-6-8-9	1110 1110
239	EF													11-0-7-8-9	1110 1111
240	F0	SRP	0	0				8		2				0	1111 0000
241	F1	MVO	1	1									1	1	1111 0001
242	F2	PACK	2	2							2			2	1111 0010
243	F3	UNPK	3	3							2	1		3	1111 0011
244	F4		4	4						4				4	1111 0100
245	F5		5	5						4			1	5	1111 0101
246	F6		6	6						4	2			6	1111 0110
247	F7		7	7						4	2	1		7	1111 0111
248	F8	ZAP	8	8					8					8	1111 1000
249	F9	CP	9	9					8				1	9	1111 1001
250	FA	AP		I										12-11-0-2-8-9	1111 1010
251	FB	SP												12-11-0-3-8-9	1111 1011
252	FC	MP												12-11-0-4-8-9	1111 1100
253	FD	DP												12-11-0-5-8-9	1111 1101
254	FE													12-11-0-6-8-9	1111 1110
255	FF													12-11-0-7-8-9	1111 1111


FIRST HALFWORD 1
SECOND HALFWORD 2
THIRD HALFWORD 3
REGISTER OPERAND 1
REGISTER OPERAND 2
RR
Op Code
R1
R2
0 7 8 11 12 15
REGISTER OPERAND 1
ADDRESS OF OPERAND 2
RX
Op Code
R1
X2
B2
D2
0 7 8 11 12 15 16 19 20 31
REGISTER OPERAND 1
REGISTER OPERAND 3
ADDRESS OF OPERAND 2
RS
Op Code
R1
R3
B2
D2
0 7 8 11 12 15 16 19 20 31
IMMEDIATE OPERAND
ADDRESS OF OPERAND 1
SI
Op Code
I2
B1
D1
0 7 8 15 16 19 20 31
ADDRESS OF OPERAND 1
S
Op Code
B1
D1
0 15 16 19 20 31
LENGTH OPERAND 1
LENGTH OPERAND 2
ADDRESS OF OPERAND 1
ADDRESS OF OPERAND 2
SS
Op Code
L1
L2/I3
B1
D1
B2
D2
0 7 8 11 12 15 16 19 20 31 32 35 36 47
LENGTH
ADDRESS OF OPERAND 1
ADDRESS OF OPERAND 2
SS
Op Code
L
B1
D1
B2
D2
0 7 8 15 16 19 20 31 32 35 36 47

CONTROL REGISTER ALLOCATION

	0	1	2	3
0	SYSTEM CONTR	TRANSL CONTR	EXTERNAL INTERRUPTION MASKS	
1	SEGM TBL LENGTH	SEGMENT TABLE ORIGIN ADDRESS		
2	CHANNEL MASKS			
3				
4				
5				
6				
7				
8			MONITOR MASKS	
9	PER EVENT MASKS		PER GR ALTERATION MASKS	
10		PER STARTING ADDRESS		
11		PER ENDING ADDRESS		
12				
13				
14	ERROR RECOVERY CONTR & MASKS			
15				

ASSIGNMENT OF CONTROL REGISTER FIELDS

CR	Bits	Name of Field	Associated with	Initial Value
0	0	Block-Multiplexing Control	Block-Multiplexing Channels	0
0	1	SSM-Suppression Control	SET System Mask	0
0	8-9	Page Size**	Dynamic Addr. Translation	0
0	10	Reserved**	Dynamic Addr. Translation	0
0	11-12	Segment Size**	Dynamic Addr. Translation	0
0	20	Clock-Comparator Mask	Clock Comparator	0
0	21	CPU-Timer Mask	CPU Timer	0
0	24	Interval-Timer Mask	Interval Timer	1
0	25	Interrupt-Key Mask	Interrupt Key	1
0	26	External-Signal Mask	External Signal	1
2	0-31	Channel Masks	Channels	1
8	16-31	Monitor Masks	Monitor Call	0
9	0	Successful-Branching-Event Mask	Program-Event Recording	0
9	1	Instruction-Fetching-Event Mask	Program-Event Recording	0
9	2	Storage-Alternation-Event Mask	Program-Event Recording	0
9	3	GR-Alternation-Event Mask	Program-Event-Recording	0
9	16-31	PER* General-Register Masks	Program-Event-Recording	0
10	8-31	PER Starting Address	Program-Event Recording	0
11	8-31	PER Ending Address	Program-Event Recording	0
14	0	Check Stop Control**	Machine Check Handling	1
14	1	Synchronous MCEL Control**	Machine Check Handling	1
14	2	I/O Extended Logout Control**	Machine Check Handling	0
14	4	Recovery-Report Mask	Machine-Check Handling	0
14	5	Degradation-Report Mask	Machine-Check Handling	0
14	6	External-Damage-Report Mask	Machine-Check Handling	1
14	7	Warning Mask	Machine-Check Handling	0
14	8	Asynchronous MCEL Control**	Machine-Check-Handling	0
14	9	Asynchronous Fixed Log Contr.**	Machine-Check-Handling	0
15	8-28	MCEL Address**	Machine-Check-Handling	512

Explanation:

* PER Means Program-Event Recording
** Only Used in /370 Mode

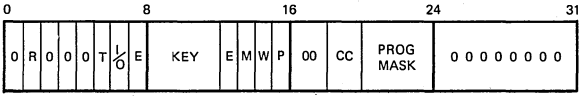
PERMANENT MAIN STORAGE ASSIGNMENT

Storage Loc Hex	Storage Loc Dec	Byte	Byte	Byte	Byte
0	0	Restart New PSW (IPL PSW)			
4	4				
8	8	Restart Old PSW (IPL CCW1)			
C	12				
10	16	(IPL CCW2)			
14	20				
18	24	External Old PSW			
1C	28				
20	32	Supervisor Call Old PSW			
24	36				
28	40	Program Old PSW			
2C	44				
30	48	Machine Check Old PSW			
34	52				
38	56	Input/Output Old PSW			
3C	60				
40	64	Channel Status Word			
44	68				
48	72	Channel Address Word			
4C	76				
50	80	Interval Timer			
54	84				
58	88	External New PSW			
5C	92				
60	96	Supervisor Call New PSW			
64	100				
68	104	Program New PSW			
6C	108				
70	112	Machine Check New PSW			
74	116				
78	120	Input/Output New PSW			
7C	124				
80	128				
84	132	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	External Interruption Code		
88	136	0 0 0 0 0 0 0 0 0 0 0 0 0 0 ILC	0	SVC Interruption Code	
8C	140	0 0 0 0 0 0 0 0 0 0 0 0 0 0 ILC	0	Program Interruption Code	
90	144	0 0 0 0 0 0 0 0	ACCESS Exception Address		
94	148	0 0 0 0 0 0 0 0	Monitor Class #	PER Code	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
98	152	0 0 0 0 0 0 0 0	PER Address		
9C	156	0 0 0 0 0 0 0 0	Monitor Code		
A0	160				
A4	164				
A8	168	Channel ID			
AC	172	Reserved			

PERMANENT MAIN STORAGE ASSIGNMENT (. . . Cont'd)

Storage Loc Hex	Storage Loc Dec	Byte	Byte	Byte	Byte
B0	176	Limited Channel Logout			
B4	180				
B8	184		0 0 0 0 0 0 0 0	I/O Address	
BC	188	Zero if an I/O Address is stored here			
⋮	⋮				
D4	212				
D8	216	CPU-Timer Save Area			
DC	220				
E0	224	Clock-Comparator Save Area			
E4	228				
E8	232	Machine Check Interruption Code			
EC	236				
F0	240				
F4	244				
F8	248	0 0 0 0 0 0 0 0	Failing Storage Address		
FC	252	Reserved			
⋮	⋮				
15C	348				
160	352	Floating Point Register Save Area			
⋮	⋮				
17C	380				
180	384	General Register Save Area			
⋮	⋮				
1BC	444				
1C0	448	Control Register Save Area			
⋮	⋮				
1FC	508				

PROGRAM STATUS WORD (EC-mode)

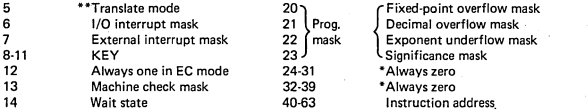


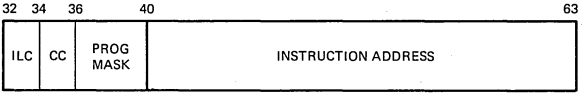
Bit	Description	Bit	Description
0	*Always zero	15	Problem state
1	PER mask	16-17	*Always zero
2-4	*Always zero	18-19	Condition code
5	**Translate mode	20	Fixed-point overflow mask
6	I/O interrupt mask	21	Decimal overflow mask
7	External interrupt mask	22	Exponent underflow mask
8-11	KEY	23	Significance mask
12	Always one in EC mode	24-31	*Always zero
13	Machine check mask	32-39	*Always zero
14	Wait state	40-63	Instruction address

(Bits 20-23: Prog. mask)

*If not zero a Specification Exeption is recognized
**Only used in /370 mode

PROGRAM STATUS WORD (BC-mode)

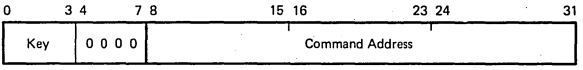




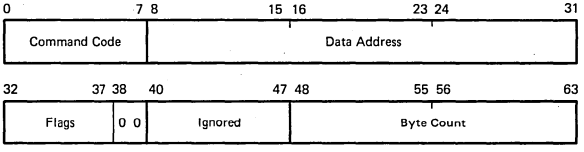
Bit	Description	Bit	Description
0	Channel 0 mask	14	Wait state
1	Channel 1 mask	15	Problem state
2	Channel 2 mask	16-31	Interruption code
3	Channel 3 mask	32-33	Instruction length code
4	Channel 4 mask	34-35	Condition code
5	Channel 5 mask	36	Fixed-point overflow mask
6	Mask for channel 6 and up	37	Decimal overflow mask
7	External interrupt mask	38	Exponent underflow mask
8-11	Protection key	39	Significance mask
12	Always 0 in BC mode	40-63	Instruction address
13	Machine check mask		

(Bits 36-39: Prog. mask)

CHANNEL ADDRESS WORD (CAW) (X'48')



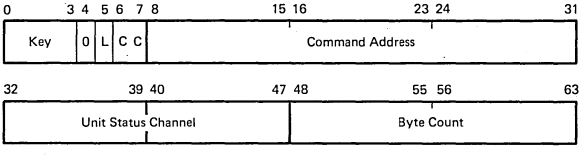
CHANNEL COMMAND WORD (CCW)



Flags

Bit	Description	
32	CD-bit (80)	: causes use of address portion of next CCW
33	CC-bit (40)	: cuases use of command code and data address of next CCW
34	SLI-bit (20)	: causes suppression of possible incorrect length indication
35	Skip bit (10)	: suppresses transfer of information to main storage
36	PCI-bit (08)	: causes a channel Program Controlled Interruption
37	IDA-bit (04)	: specifies indirect data addressing (only /370 mode)

CHANNEL STATUS WORD (CSW) (X'40')

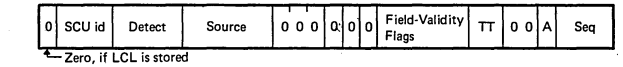


Status

Bit	Description		Bit	Description	
5		Logout pending	40	(0080)	Program controlled interruption
6, 7		Deferred cond. code	41	(0040)	Incorrect length
32	(8000)	Attention	42	(0020)	Program check
33	(4000)	Status Modifier	43	(0010)	Protection check
34	(2000)	Control unit end	44	(0008)	Channel data check
35	(1000)	Busy	45	(0004)	Channel control check
36	(0800)	Channel end	46	(0002)	Interface control check
37	(0400)	Device end	47	(0001)	Chaining check
38	(0200)	Unit check			
39	(0100)	Unit exception			

Byte count: Bits 48-63 form the residual count for the last CCW used.

LIMITED CHANNEL LOGOUT (X'B0')



Detect

Bit	Description
4	CPU
5	Channel
6	Storage control
7	Storage

Source

Bit	Description
8	CPU
9	Channel
10	Storage control
11	Storage
12	Control unit

Field Validity Flags

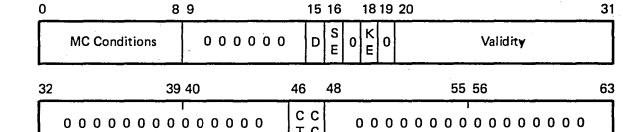
Bit	Description
19	Sequence code
20	Unit status
21	CCW address and key
22	Channel address
23	Device address

TT (Type of termination)

Bit 24-25	Description
0 0	Interface disconnect
0 1	Stop, stack or normal
1 0	Selective reset
1 1	Reserved

A (bit 28): I/O error alert
Seq (bits 29-31): Sequence code

MACHINE CHECK INTERRUPTION CODE (X'E8')



MC Conditions

Bit	Description
0	System damage
1	Instr. proc'g damage
2	System recovery
3	Timer damage
4	Timing Facility
5	External damage
6	–
7	Degradation
8	Warning

Time

Bit	Description
15	Delayed

Stg error

Bit	Description
16	Storage error uncorrected
18	Storage-key error uncorrected

Validity

Bit	Description
20	PSW bits 12-15
21	PSW masks and key
22	Program mask and cond. code
23	Instruction address
24	Failing storage address
25	Region code

Bit	Description
27	Floating-point registers
28	General registers
29	Control registers
30	CPU extended logout
31	Storage logical
46	CPU Timer
47	Clock comparator

CODES FOR PROGRAM INTERRUPTION (X'8E')

Interruption Code		Program Interruption Cause	Interruption Code		Program Interruption Cause
Dec	Hex		Dec	Hex	
1	0001	Operation	12	000C	Exponent overflow
2	0002	Privileged operation	13	000D	Exponent underflow
3	0003	Execute	14	000E	Significance
4	0004	Protection	15	000F	Floating point divide
5	0005	Addressing	16	0010	Segment translation*
6	0006	Specification	17	0011	Page translation*
7	0007	Data	18	0012	Translation specification*
8	0008	Fixed-point overflow	19	0013	Special operation
9	0009	Fixed-point divide	24	0018	Page access
10	000A	Decimal overflow	26	001A	Page state
11	000B	Decimal divide	27	001B	Page translation
			64	0040	Monitor event
			128	0080	Program event

*Used only in /370 mode

CODES FOR EXTERNAL INTERRUPTIONS (X'86')

Interruption Code (Hex)	External Interruption Cause	Indication of Concurrent Condition by Bit Positions
0080	Interval timer	9-15
0040	Interrupt key	8 and 10-15
0020	External signal 2	8, 9, 11-15
0010	External signal 3	8-10, 12-15
0008	External signal 4	8-11, 13-15
0004	External signal 5	8-12, 14, 15
0002	External signal 6	8-13, 15
0001	External signal 7	8-14
1004	Clock comparator	
1005	CPU timer	

CODES FOR INPUT/OUTPUT INTERRUPTIONS (X'BA')

Interruption Code (Hex)	Input/Output Interruption Cause
00 dd	Channel 0
01 dd	Channel 1
02 dd	Channel 2
03 dd	Channel 3
04 dd	Channel 4
05 dd	Channel 5
06 dd	Channel 6

Note: d = device address

CODES FOR SUPERVISOR CALL INTERRUPTIONS (X'8A')

Interruption Code (Hex)	Supervisor Call Interruption Cause
00 rr	Instruction (0A)

Note: r = R1 and R2 field of SUPERVISOR CALL

CHANNEL COMMANDS

Standard Command Code Assignments (CCW bits 0-7) for I/O Operations

xxxx	0000	Invalid	††††††01	Write
††††	0100	Sense	††††††10	Read
xxxx	1000	Transfer in Channel	††††††11	Control
††††	1100	Read backward	0000 0011	Control No Operation

x = Bit ignored † = Modifier bit for specific type of I/O device

3210, 3215 CONSOLES — Source: GA 24-3557

Write, No Carrier Return	01	Sense	04
Write, Auto Carrier Return	09	Audible Alarm	0B
Read Inquiry	0A		

3505 CARD READER / 3525 CARD PUNCH — Source: GA 21-9124

Command	Binary	Hex
Sense	0 0 0 0 0 1 0 0	0 4
Feed, Select Stacker	S S 1 0 F 0 1 1	
Read Only*	1 1 D 0 F 0 1 0	
Diagnostic Read	1 1 0 1 0 0 1 0	D 2
Read, Feed, Select Stacker*	S S D 0 F 0 1 0	
Write RCE Format* †	0 0 0 1 0 0 0 1	1 1
3505 only		
Write OMR Format	0 0 1 1 0 0 0 1	3 1
3525 only		
Write, Feed, Select Stacker	S S D 0 0 0 0 1	
Print Line*	L L L L L 1 0 1	

Bit Meanings

SS	Stacker
00	1
01	2
10	2

Format Mode

0	Unformatted
1	Formatted

D	Data Mode
0	1-EBCDIC
1	2-Card Image

L — Line Position

5 Bit Binary Value

*Special feature on 3525 † Special feature on 3505

3211 PRINTER/3811 CONTROL UNIT — Source: GA24-3543

	After Write	Immed.
Space 1 Line	09	0B
Space 2 Lines	11	13
Space 3 Lines	19	1B
Skip to Channel 0	—	83
Skip to Channel 1	89	8B
Skip to Channel 2	91	93
Skip to Channel 3	99	9B
Skip to Channel 4	A1	A3
Skip to Channel 5	A9	AB
Skip to Channel 6	B1	B3
Skip to Channel 7	B9	BB
Skip to Channel 8	C1	C3
Skip to Channel 9	C9	CB
Skip to Channel 10	D1	D3
Skip to Channel 11	D9	DB
Skip to Channel 12	E1	E3

Write without spacing	01
Sense	04
Load UCSB	FB
Fold	43
Unfold	23
Load FCB	63
Block Data Check	73
Allow Data Check	7B
Read PLB	02
Read UCSB	0A
Read FCB	12
Check Read	06
Diagnostic Write	05
Raise Cover	6B
Diagnostic Gate	07

3803/3420 MAGNETIC TAPE — Source: GA32-0020

Write	01
Read Forward	02
Read Backward	0C
Sense	04
Sense Reserve*	F4
Sense Release*	D4
Request Track in Error	1B
Loop Write to Read	8B
Set Diagnose	4B
Rewind	07
Rewind Unload	0F
Erase Gap	17
Write Tape Mark	1F
Backspace Block	27
Backspace File	2F
Forward Space Block	37
Forward Space File	3F

Data Security Erase	97
Diagnostic Write Mode Set	0B
Set Mode 1 (7-track) †	

Density	Parity	DC	Trans	Cmd
556	odd	on	off	53
		off	off	73
			on	7B
	even		off	63
		off	on	6B
800	odd	on	off	93
		off	off	B3
			on	BB
	even		off	A3
		off	on	AB

Set Mode 2 (9-track)

1600 bpi	C3
800 bpi †	CB

*Two-channel switch required † Special feature for NRZI operation

8809 MAGNETIC TAPE

Command	Command Code
Write	01
Read	02
Rewind	07
Rewind-Unload	0F
Erase Gap	17
Write Tape Mark	1F
Backspace Block	27
Backspace File	2F
Forwardspace Block	37
Forwardspace File	3F
Data Security Erase	97
Set Low Speed	83
Set Long Gap	13
Set Normal Gap	23
Set High Speed and Long Gap	93
Set High Speed and Normal Gap	33
Set Low Speed and Long Gap	53
Set Low Speed and Normal Gap	63
Set High Speed	E3
ERP Loop Write-to-Read	8B
Control-No Op	03
Sense	04
Sense I/O	E4
Read and Reset Buffered Log	A4

DIRECT ACCESS DEVICES

Source: GA26-1592 for 3830/3330
GA26-3599, GA26-1606 for 2314,2319

	Command	MT Off	MT on*	Count
Control	Orient (c)	2B	B9	Nonzero
	Recalibrate	13		Nonzero
	Seek	07		6
	Seek cylinder	0B		6
	Seek Head	1B		6
	Space Count	0F		3 (a); nonzero (d)
	Set File Mask	1F		1
	Set Sector (a)	23		1
	Restore (a)	17		Nonzero
	Vary Sensing (c)	27		1
	Diagnostic Load (a)	53		1
	Diagnostic Write (a)	73		512
Search	Home Address Equal	39	B9	4
	Identifier Equal	31	B1	5
	Identifier High	51	D1	5
	Identifier Equal or High	71	F1	5
	Key Equal	29	A9	KL
	Key High	49	C9	KL
	Key Equal or High	69	E9	KL
	Key and Data Equal (d)	2D	AD	Number of bytes (incl. mask bytes) in search argument
	Key and Data High (d)	4D	CD	
	Key and Data Equal or High (d)	6D	ED	
Continue Scan	Search Equal (d)	25	A5	
	Search High (d)	45	C5	
	Search High or Equal (d)	65	E5	
	Set Status Modifier (d)	35	B5	
	Set Status Modifier (d)	75	F5	
	No Status Modifier (d)	55	D5	

* Code same as MT Off except as listed
a 3830/3330 and 2835/2305 only
b 3830/3330 only
c 2835/2305 only
d 2314, 2319 only
e Channel attachment and 2-channel switch feature required; standard on 2314 with 2844

DIRECT ACCESS DEVICES (. . . Cont'd)

	Command	MT Off	MT On*	Count
Read	Home Address	1A	9A	5
	Count	12	92	8
	Record 0	16	96	Number of bytes to be transferred
	Data	06	86	
	Key and Data	0E	8E	
	Count, Key and Data	1E	9E	
	IPL	02		
	Sector (a)	22		1
Sense	Sense I/O	04		24 (a); 6 (d)
	Read, Reset Buffered Log (b)	A4		24
	Read Buffered Log (c)	24		128
	Device Release (e)	94		24 (a); 6 (d)
	Device Reserve (e)	B4		24 (a); 6 (d)
	Read Diagnostic Status 1 (a)	44		16 or 512
Write	Home Address	19		5
	Record 0	15		8+KL+DL or R0
	Erase	11		8+KL+DL
	Count, Key and Data	1D		8+KL+DL
	Special Count, Key and Data	01		8+KL=DL
	Data	05		DL
	Key and Data	0D		KL+DL

* Code same as MT Off except as listed
a 3830/3330 and 2835/2305 only
b 3830/3330 only
c 2835/2305 only
d 2314, 2319 only
e Channel attachment and 2-channel switch feature required; standard on 2314 with 2844

DIRECT ACCESS DEVICES (3310,3370) FBA MODE

	Command	MT Off	MT On*	Count
Control	No-Operation	03		
	Define Extent	63		
	Locate	43		
Read	Read	42		
	Read IPL	02		
Write	Write	41		
Sense	Test I/O	00		
	Sense I/O	E4		
	Sense	04		
	Read and Reset Buffered Log	A4		
	Read Device Characteristics	64		
	*Device Reserve	B4		
	*Unconditional Reserve	14		
	*Device Release	94		
Diagnostic	Diagnostic Control	F3		
	Diagnostic Sense	C4		

*These commands are executed as sense

HEXADECIMAL AND DECIMAL CONVERSION

From Hex: Locate each hex digit in its corresponding column position and note the decimal equivalents. Add these to obtain the decimal value.

From Dec: Locate the largest decimal value in the table that will fit into the decimal number to be converted.
Note its hex equivalent and hex column position.
Find the decimal remainder.
Repeat the process on this and subsequent remainders.

Hexadecimal Columns											
6		5		4		3		2		1	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0	0	0	0	0
1	1 048 576	1	65 536	1	4 096	1	256	1	16	1	1
2	2 097 152	2	131 072	2	8 192	2	512	2	32	2	2
3	3 145 728	3	196 608	3	12 288	3	768	3	48	3	3
4	4 194 304	4	262 144	4	16 384	4	1 024	4	64	4	4
5	5 242 880	5	327 680	5	20 480	5	1 280	5	80	5	5
6	6 291 456	6	393 216	6	24 576	6	1 536	6	96	6	6
7	7 340 032	7	458 752	7	28 672	7	1 792	7	112	7	7
8	8 388 608	8	524 288	8	32 768	8	2 048	8	128	8	8
9	9 437 184	9	589 824	9	36 864	9	2 304	9	144	9	9
A	10 485 760	A	655 360	A	40 960	A	2 560	A	160	A	10
B	11 534 336	B	720 896	B	45 056	B	2 816	B	176	B	11
C	12 582 912	C	786 432	C	49 152	C	3 072	C	192	C	12
D	13 631 488	D	851 968	D	53 248	D	3 328	D	208	D	13
E	14 680 064	E	917 504	E	57 344	E	3 584	E	224	E	14
F	15 728 640	F	983 040	F	61 440	F	3 840	F	240	F	15
0123		4567		0123		4567		0123		4567	
Byte				Byte				Byte			

POWERS OF 2

2^n	n
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

$2^n = 16^{n:4}$

POWERS OF 16

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 976 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15

SENSE INFORMATION SUMMARY

1017 - Paper Tape Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention reject 2 : bus-out check 3 : – 4 : data check 5 : – 6 : – 7 : broken tape

1018 - Paper Tape Punch

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : – 6 : – 7 : -

1287 - Optical Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : non-recovery 7 : keyboard correction (tape only)
1	Bit 0 : tape mode 1 : late stacker select 2 : no document found 3 : – 4 : invalid operation 5 : – 6 : – 7 : –

1288 – Optical Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : non-recovery 7 : –
1	Bit 0 : – 1 : end-of-page 2 : no document found 3 : – 4 : invalid operation 5 : – 6 : – 7 : –

1403 – Printer

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : parity check (UCS storage) 6 : – 7 : channel 9
1	Not used
2	Bit 0 : chain interlock 1 : forms check 2 : coil protect check 3 : subscan ring check 4 : chain buffer address register check 5 : – 6 : any hammer on check 7 : –
3	Not used
4	Bit 0 : hammer reset failure check 1 : no fire check 2 : misfire check 3 : print data buffer parity check 4 : check bit buffer parity check 5 : chain buffer parity check 6 : buffer address register parity check 7 : clock check
5	Bit 0 : open hammer coil check 1-7 : –

1419 – PCU – MICR

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : – 4 : data check 5 : overrun 6 : autoselect 7 : –
1	Bit 0 : – 1 : – 2 : document under read head 3 : amount field valid 4 : process-control field valid 5 : account-number field valid 6 : transit field valid 7 : serial-number field valid

1419 – SCU – MICR

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : – 4 : – 5 : late stacker select 6 : autoselect 7 : operator attention

SENSE INFORMATION SUMMARY (. . . . Cont'd)

1442 – Card Read-Punch / Card Punch

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : – 7 : –

1443 – Printer

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : type bar 5 : type bar 6 : – 7 : –

2260 – Display Station

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4-7 : –

2311 – Disk Storage

Sense Byte	Deisgnation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : track condition check 7 : seek check
1	Bit 0 : data check in count area 1 : track overrun 2 : end of cylinder 3 : invalid sequence 4 : no record found 5 : file protect 6 : missing address marker 7 : overflow incomplete

2311 – Disk Storage (Cont'd)

Sense Byte	Designation
2	Bit 0 : unsafe 1 : – 2 : serializer check 3 : – 4 : ALU check 5 : unselected file status 6 : – 7 : –
3	Bit 0 : ready 1 : on line 2 : unsafe 3 : – 4 : on line 5 : end of cylinder 6 : – 7 : seek incomplete
4	Bit 0-7 : –
5	Bit 0-7 : command in progress when overflow incomplete occurs

2314/2319 – Direct Access Storage

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : track condition check 7 : seek check
1	Bit 0 : data check in count area 1 : track overrun 2 : end of cylinder 3 : invalid sequence 4 : no record found 5 : file protect 6 : missing address marker 7 : overflow incomplete
2	Bit 0 : unsafe 1 : – 2 : SERDES check 3 : – 4 : ALU check 5 : unselected status 6 : – 7 : –
3	Bit 0 : busy 1 : on line 2 : unsafe 3 : wr current sense 4 : pack change 5 : end of cylinder 6 : multi-module select 7 : seek incomplete

2314/2319 – Disk Access Storage (Cont'd)

Sense Byte	Designation
4	Bit 0 : wrong length record (2314 with multiplex storage control feature only) 1 : pending status (2314 with multiplex storage control feature only) 2 : – 3 : – 4-7 : Module indentification

bits 4567	physical drive
0000	A
0001	B
0010	C
0011	D
0100	E
0101	F
0110	G
0111	H
1000	J
1111	module not defined

5	Bit 0-7 : command in progress when overflow incomplete occurs.

2400 – Magnetic Tape

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : word count zero 7 : data converter check
1	Bit 0 : noise 1-2 : B'00' = not existent B'01' = not ready B'10' = ready and not rewinding B'11' = ready and rewinding 3 : seven-track 4 : at loadpoint 5 : selected and write status 6 : file protect 7 : not capable
2	Bit 0-7 : contains the track-in-error indicator bits that are set at the end of a read, or read-backward command if a data check has been encountered. Bits 6 and 7 on together indicate either more than one error or no error found.
3	Bit 0 : R/W VRC 1 : LRCR 2 : skew 3 : CRC 4 : skew register VRC 5 : phase encoding 6 : backward 7 : C compare
4	Bit 0 : echo check 1 : reject TU 2 : read clock error 3 : write clock error 4 : delay counter 5 : sequence indicator C 6 : sequence indicator B 7 : sequence indicator A

2501 – Card Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : – 7 : –

2520 – Card Read-Punch/Card Punch

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : – 7 : –

2540 – Card Reader/Card Punch

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : – 6 : unusual command 7 : –

2560 – Multifunction Card Machine

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : – 3 : equipment check 4 : data check 5 : feed/Machine check 6 : no card available 7 : print operation in progress
1	Bit 0 : cover interlock/punch pusher check 1 : jam bar check 2 : corner station check 3 : call 8 to 9 feed check 4 : print station feed check 5 : punch station feed check 6 : read station feed check 7 : input station feed check
2	Location of individual card: Bit 0 : secondary select 1 : card in punch station 2 : preprint SC7 exposed 3 : prepunch SC5 exposed 4 : prepunch SC4 exposed 5 : preread SC3 exposed 6 : preread SC2 exposed 7 : input station SC1 exposed

2560 – Multifunction Card Machine (. . . . Cont'd)

Sense Byte	Designation
3	Stacker Select Information: Bit 0 : primary card 0) 1 : binary value 4) card at primary 2 : binary value 2) prepunch station 3 : binary value 1) 4 : secondary card 1) 5 : binary value 4) card at secondary 6 : binary value 2) prepunch station 7 : binary value 1)
4	Stacker Select Information: Bit 0 : primary (0)) secondary (1)) 1 : binary value 4) card at punch or 2 : binary value 2) preprint station 3 : binary value 1 4 : primary (0)) secondary (1)) 5 : binary value 4) card after print 6 : binary value 2) station 7 : binary value 1)
5	Stacker Select Information: Bit 0 : primary (0)) secondary (1)) 1 : 4) card at corner station 2 : 2) 3 : 1) 4 : primary (0)) secondary (1)) card in stacker pocket 5 : 4) (was just stacked) 6 : 2) 7 : 1)
6	Card column in which first (possibly only) error was detected: Bit 0 : multi data check 1 : binary value 64 2 : binary value 32 3 : binary value 16 4 : binary value 8 5 : binary value 4 6 : binary value 2 7 : binary value 1

2596 – Card Read-Punch

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : – 7 : –

2671 – Paper Tape Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : – 6 : – 7 : –

3203 – Printer

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : – 3 : equipment check 4 : data check 5 : chain buffer parity check 6 : no channel found 7 : channel 9
1	Bit 0-7 : –
2	Bit 0 : interlock (chain gate open) 1 : form check (jam) 2 : coil protect check 3 : subscan ring check 4 : chain buffer address register check 5 : hammer unit shift check (model 1 only) 6 : any hammer on check 7 : device ready check
3	Bit 0 : – 1 : – 2 : – 3 : carriage inhibit check 4 : – 5 : – 6 : step check 7 : move check
4	Bit 0 : hammer reset failure check 1 : no fire check 2 : misfire check 3 : print data buffer parity check 4 : check bit buffer parity check 5 : chain buffer parity check 6 : buffer address register check 7 : clock check
5	Bit 0 : open coil check 1-7 : –

3210/3215 – Console Printer Keyboard

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : – 3 : equipment check 4-7 : –

PRT1 (3203-4, 3203-5, 3211, 3289)

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : buffer parity check 6 : load check 7 : channel 9
1	Bit 0 : command parity 1 : print check 2 : print quality 3 : line position check 4 : forms check 5 : command suppress 6 : mechanical motion 7 : –

PRT1 (3203-4, 3203-5, 3211, 3289) (Cont'd)

Sense Byte	Designation
2	Bit 0 : carriage failed to move 1 : carriage sequence check 2 : carriage stop 3 : platen failed to advance 4 : platen failed to retract 5 : forms jam 6 : ribbon motion 7 : train overload
3	Bit 0 : UCSB parity 1 : PLB parity 2 : FCB parity 3 : coil protect 4 : hammer fire check 5 : service aid 6 : UCSAR sync check 7 : PSE sync check
4	Bit 0-7 : information used by service personnel
5	Bit 0-7 : –

3272 – (3270 Local)

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus out check 3 : equipment check 4 : data check 5 : unit specify 6 : control check 7 : operation check

FBM (3310 and 3370) Disk Storage

Sense Byte	Designation	Format
0	Bit 0 : command reject	0
	1 : intervention required	1
	2 : bus out parity (not used)	0
	3 : equipment check	1
	4 : data check	4
	5 : overrun	0
	6 : (unused)	–
	7 : (unused)	–
1	Bit 0 : permanent error	any
	1 : (unused)	–
	2 : (unused)	–
	3 : (unused)	–
	4 : (unused)	–
	5 : file protected	0
	6 : write inhibited	–
	7 : operation incomplete	0
2	Bit 0 : check data error	4
	1 : correctable	5
	2 : (unused)	–
	3 : environmental data present	6,4,1
	4 : (unused)	–
	5 : (unused)	–
	6 : (unused)	–
	7 : only logging required	–
3-6	physical address	
7	Bit 0-3 : hex: format number 4-7 : hex: message code	
8-23	only for diagnose information	

SENSE INFORMATION SUMMARY (. . . . Cont'd)

3330 – Disk Storage

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : – 7 : –
1	Bit 0 : permanent error 1 : invalid track format 2 : end of cylinder 3 : – 4 : no record found 5 : file protected 6 : write inhibited 7 : operation incomplete
2	Bit 0 : – 1 : correctable 2 : – 3 : environmental data present 4 : – 5 : – 6 : – 7 : –
3	Bit 0-7 : restart command
4	Bit 0-1 : storage control identification 2-7 : physical drive identification

bits 2 to 7	physical drive
111000	A
110001	B
101010	C
100011	D
011100	E
010101	F
001110	G
000111	H

5	Bit 0-7 : identify the eight low-order bits of the cylinder address in the most recent seek argument
6	Bit 0 : reserve 1 : cylinder number (high order bit of cylinder address) 2 : difference 3 : 16) 4 : 8) 5 : 4) head number 6 : 2) 7 : 1)
7	Bit 0-3 : format type of remaining sense bytes (8-23) 4-7 : encoded error message
8-23	Meaning depends on format type

3340 – Disk Storage

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : track condition check 7 : seek check

3340 – Disk Storage

Sense Byte	Designation
1	Bit 0 : permanent error 1 : invalid track format 2 : end of cylinder 3 : – 4 : no record found 5 : file protected 6 : write inhibited 7 : operation incomplete
2	Bit 0 : RPS feature present 1 : correctable 2 : – 3 : environmental data present 4 : – 5 : – 6 : data module size) 01 = 35MB 7 : data module size) 10 = 70MB
3	Bit 0-7 : restart command
4	Physical drive identification: Bit 0 : drive A 1 : drive B 2 : drive C 3 : drive D 4 : drive E 5 : drive F 6 : drive G 7 : drive H
5	Bit 0-7 : Identifies the eight low-order bits of the cylinder address in the most recent seek argument
6	Bit 0-2 : identifies the three high-order bits of the cylinder address 3 : – 4 : 8) 5 : 4) head number 6 : 2) 7 : 1)
7	Bit 0-3 : format type of remaining sense bytes (8-23) 4-7 : encoded error message
8-23	Meaning depends on format type

3344 – Direct Access Storage (3340 Mode)

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : track condition check 7 : seek check
1	Bit 0 : permanent error 1 : invalid track format 2 : end of cylinder 3 : – 4 : no record found 5 : file protected 6 : write inhibited 7 : operation incomplete
2	Bit 0 : RPS feature present 1 : correctable 2 : – 3 : environmental data present 4 : Compatibility Mode / 3344 5 : HDA Size 3 Bit 6 : HDA Size 2 Bit 7 : HDA Size 1 Bit

3344 — Direct Access Storage (Cont'd)

Sense Byte	Designation
3	Bit 0-7 : restart command
4	Controller device address Bit 0 : Controller Addr. Bit 2 1 : Controller Addr. Bit 1 2 : 3 : } Controller Device Address 4 : 5 : Device Addr. Bit 4 6 : Device Addr. Bit 2 7 : Device Addr. Bit 1
5	Bit 0-7 : Identifies the eight low-order bits of the cylinder address
6	Bit 0-3 : Identifies the four high-order bits of the cylinder address 0 : 1024 1 : 512 2 : 256 3 : 2048 4 : 8) 5 : 4) head number 6 : 2) 7 : 1)
7	Bit 0-3 : format type of remaining sense bytes (8-23) 4-7 : encoded error message
8-23	Meaning depends on format type

3350 - Direct Access Storage

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : channel bus out parity 3 : equipment check 4 : overrun 5 : not used 7 : not used
1	Bit 0 : permanent error 1 : invalid track format 2 : end of cylinder 3 : not used 4 : no record found 5 : file protected 6 : write inhibited 7 : operation incomplete
2	Bit 0 : not used 1 : correctable 2 : not used 3 : environmental data present 4 : computibility mode 5 : not used 6 : not used 7 : not used
3	Bit 0-7 : restart command
4	Bit 0-7 : physical drive identification Bit: physical drive 0 : A 1 : B 2 : C 3 : D 4 : E 5 : F 6 : G 7 : H

3350 – Direct Access Storage (Cont'd)

Sense Byte	Designation
5	Bit 0-7 : low order logical cylinder address Bit: Value 0 : 128 1 : 64 2 : 32 3 : 16 4 : 8 5 : 4 6 : 2 7 : 1
6	Bit 0-2 : Identifies the three high-order bits of the cylinder address Bit 0 : CE Cylinder 1 : 3330 - 11 = 512 3330 - 1 = 256 2 : 3330 - 11 = 256 3330 - 1 = 0 3 : 16) 4 : 8) 5 : 4) Head number 6 : 2) 7 : 1)
7	Bit 0-3 : FORMAT TYPE of remaining sense bytes (8-23)
8-23	Meaning depends on format type (see 3350 MLM)

3370 – Disk Storage

Sense Byte	Designation	Format
0	Bit 0 : command reject	0
	1 : intervention required	1
	2 : bus out parity (not used)	0
	3 : equipment check	1
	4 : data check	4
	5 : overrun	0
	6 : (unused)	–
	7 : (unused)	–
1	Bit 0 : permanent error	any
	1 : (unused)	–
	2 : (unused)	–
	3 : (unused)	–
	4 : (unused)	–
	5 : file protected	0
	6 : write inhibited	–
	7 : operation incomplete	0
2	Bit 0 : check data error	4
	1 : correctable	5
	2 : (unused)	–
	3 : environmental data present	6, 4, 1
	4 : (unused)	–
	5 : (unused)	–
	6 : (unused)	–
	7 : only logging required	–
3-6	physical address	
7	Bit 0-3 : hex: format number 4-7 : hex: message code	
8-23	Only for diagnose Information	

3410/3411 – Magnetic Tape

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : word count zero 7 : data converter check
1	Bit 0 : noise 1-2 : B'00' = non-existent B'01' = not ready B'10' = ready and not busy B'11' = ready and busy 3 : seven track 4 : at load point 5 : write status 6 : file protected 7 : not capable
2	Bit 0-7 : track in error bits
3	Bit 0 : VRC 1 : multiple track error (PE) or LRC (NRZI) 2 : skew 3 : end data check (PE) or CRC (NRZI) 4 : envelope check (PE only) 5 : phase encoding 6 : backword 7 : –
4	Bit 0 : tape unit positioning check 1 : tape unit reject 2 : end of tape 3 : – 4 : – 5 : diagnostic track check 6 : tape unit check 7 : illegal command
5	Bit 0 : new subsystem 2 : write tape mark check 3 : PE identification burst 4 : PE compare 5 : tachometer check 6 : false end mark 7 : RPQ
6	Bit 0 : seven track 1 : short gap mode 2 : dual density 4-7 : tape unit model
7	Bit 0 : lamp check 1 : left column check 2 : right column check 3 : ready reset 4 : data security erase 5-7 : –
8	Bit 0 : – 1 : feedthrough 2 : – 3 : end velocity check 4 : no read-back data 5 : start velocity check 6 : – 7 : –

SENSE INFORMATION SUMMARY (. . . . Cont'd)

3420/3803 - Magnetic Tape

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : overrun 6 : word count zero 7 : data converter check
1	Bit 0 : noise 1-2 : B'00' = non-existent B'01' = not ready B'10' = ready and not rewinding B'11' = ready and rewinding 3 : seven track 4 : at load point 5 : write status 6 : file protected 7 : not capable
2	Bit 0-7 : track in error bits
3	Bit 0 : VRC 1 : multiple track error (PE) or LPC (NRZI) 2 : skew 3 : end data check (PE) or CRC (NRZI) 4 : envelope check (PE only) 5 : phase encoding 6 : backward 7 : C-compare
4	Bit 0 : ALU hardware error 1 : reject tape unit 2 : tape indicate 3 : write trigger VRC 4 : microprogram detected error 5 : LWR 6 : tape unit check 7 : RPQ
5	Bit 0 : new subsystem 1 : new subsystem 2 : write tape mark check 3 : PE ID burst check 4 : start read check 5 : partial record 6 : excessive postable or tape mark 7 : RPQ
6	Bit 0 : seven track 1 : write current failure 2 : dual density 3 : NRZI density 4-7 : tape unit model
7	Bit 0 : lamp failure 1 : tape bottom left 2 : tape bottom right 3 : reset key 4 : data security erase 5 : erase head 6 : air bearing pressure 7 : load failure
8	Bit 0 : IBG drop while writing 1 : feed through check 2 : SDR counter 3 : early begin readback check 4 : early ending readback check 5 : slow begin readback check 6 : slow ending readback check 7 : velocity retry/restart

3420/3803 – Magnetic Tape (. . . Cont'd)

Sense Byte	Designation
9	Bit 0 : SDR counter 1 : velocity change during write 2-3 : SDR counter 4 : – 5 : – 6 : – 7 : tape control reserved
10	Bit 0 : command status reject 1 : – 2 : control status reject 3 : no block on record readback check 4 : WTM not detected block 5 : tachometer start fail 6 : – 7 : velocity check
11	Bit 0 : B bus parity error, ALU 1 1 : – 2 : low ROS parity/low IC/parity on branch instr. 3 : high IC/high ROS reg parity 4 : micro program detected hardware error 5 : D bus parity error, ALU1 6 : – 7 : branch condition error, ALU2
12	Bit 0 : B bus parity error, ALU 2 1 : – 2 : low ROS parity/low IC/parity on branch instr. 3 : high IC/BC/high ROS reg parity 4 : microprogram detected hardware error 5 : D bus parity error, ALU 2 6 : – 7 : branch condition error, ALU 2
13	Bit 0-1 : tape control density 2-7 : tape control unique ID high
14	Bit 0-7 : tape control unique ID low
15	Bit 0-7 : tape unit unique ID
16	Bit 0-7 : tape unit unique ID
17	Bit 0 : two-channel switch 1-3 : tape control device switch features 4-7 : EC level of tape control
18	Bit 0 : Power check/air flow 1-3 : – 4-7 : EC level of tape unit
19	Bit 0 : primed for device and tape unit 7 1 : primed for device and tape unit 6 2 : primed for device and tape unit 5 3 : primed for device and tape unit 4 4 : primed for device and tape unit 3 5 : primed for device and tape unit 2 6 : primed for device and tape unit 1 7 : primed for device and tape unit 0
20	Bit 0 : primed for device and tape unit F 1 : primed for device and tape unit E 2 : primed for device and tape unit D 3 : primed for device and tape unit C 4 : primed for device and tape unit B 5 : primed for device and tape unit A 6 : primed for device and tape unit 9 7 : primed for device and tape unit 8

3420/3803 – Magnetic Tape (. . . . Cont'd)

Sense Byte	Designation
21	Bit 0 : load button depressed 1 : left reel turning 2 : right reel turning 3 : tape present 4 : reels loaded 5 : load rewind 6 : load complete 7 : load check
22	Bit 0-7 : FRU identifiers for tape control
23	Bit 0-7 : FRU identifiers for tape control

3504/3505/3525 – Card I/O

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : – 6 : abnormal format reset 7 : permanent error key
1	Bit 0 : permanent error 1 : automatic retry 2 : motion malfunction 3 : retry after intervention complete 4-7 : –
2-3	Used for diagnostic purposes only

3540 – Diskette

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5-7 : –
1	Bit 0 : permanent error 1 : automatic retry 2 : motion malfunction 3 : retry after intervention complete 4 : special record transferred 5-7 : –
2	Used for diagnostic purposes only
3	Bit 0-7 : cylinder address in binary
4	Bit 0-7 : head address, must be binary zero
5	Bit 0-7 : record address in binary

3881 – Optical Mark Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : – 5 : – 6 : unusual command sequence 7 : –

3886 – Optical Character Reader

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : – 5 : – 6 : non-initialized 7 : RCP error
1	Bit 0 : – 1 : mark check 2 : invalid format 3 : – 4 : incomplete scan 5 : – 6 : non-recovery 7 : outboard

5203 – Printer

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : – 3 : equipment check 4 : data check 5 : chain buffer parity check 6 : no channel found 7 : channel 9
1	Bit 0-7 : –
2	Bit 0 : interlock (chain gate open) 1 : forms check (jam) 2 : coil protect check 3 : subscan ring check 4 : chain buffer address register check 5 : hammer unit shift check 6 : any-hammer-on check 7 : thermal overload
3	Bit 0-7 : –
4	Bit 0 : hammer reset failure check 1 : no fire check 2 : misfire check 3 : print data buffer parity check 4 : check bit buffer parity check 5 : chain buffer parity check 6 : buffer address register check 7 : clock check
5	Bit 0 : open coil check 1-7 : –

5424/5425 – Multifunction Card Unit

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : bus-out check 3 : equipment check 4 : data check 5 : – 6 : no card available 7 : –

5424/5425 – Multifunction Card Unit (. . . . Cont'd)

Sense Byte	Designation
1	Bit 0 : read check 1 : punch check 2 : – 3 : print data check 4 : print clutch check 5 : hopper check 6 : feed check 7 : –
2	Bit 0 : – 1 : – 2 : card in primary wait station 3 : card in secondary wait station 4 : NPRO allowed 5 : hopper cycle not complete 6 : card in transport counter bit 2 7 : card in transport counter bit 1
3	Contains a hexadecimal number whose value can represent feedchecks and emitter checks in the 5425
4	Defines the card column group and tier where the error was detected which caused the first read check or punch check of a card cycle: Bit 0 : multiple error 1-2 : B'00' Tier 1 B'01' Tier 2 B'10' Tier 3 3-7 : B'00000' column group 1 B'00001' column group 2 . . . . . . B'11111' column group 32
5	Bit 0 : D row miscompare 1 : C row miscompare 2 : B row miscompare 3 : A row miscompare 4 : 8 row miscompare 5 : 4 4ow miscompare 6 : 2 row miscompare 7 : 1 row miscompare
6-10	Forms a table of the five most recent command strings Bit 0 : Secondary 1 : print four lines 2 : stacker select M2 3 : stacker select M3 4 : punch 5 : feed command sample 6 : print 7 : read

8809 – Tape Unit

Sense Byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : – 3 : equipment check 4 : data check 5 : overrun 6 : – 7 : –
1	Bit 0 : noise 1 : tape unit status A 2 : tape unit status B 3 : – 4 : at load point 5 : write status 6 : file protected 7 : not capable

8809 – Tape Unit (Cont'd)

Sense Byte	Designation
2	Bit 0-7 : represent track in error pointers
3	Bit 0 : 128 ERP number 1 : 64 ERP number 2 : 32 ERP number 3 : 16 ERP number 4 : 8 ERP number 5 : 4 ERP number 6 : 2 ERP number 7 : 1 ERP number
4	Bit 0 : – 1 : – 2 : tape indicate 3 : permanent error 4 : host detected error 5 : loop write to read error 6 : not used 7 : not used
5	Bit 0 : – 1 : – 2 : – 3 : PE-ID burst check 4 : – 5 : – 6 : – 7 : –
6	Contains all zeros
7	Bit 0 : 8 format code 1 : 4 format code 2 : 2 format code 3 : 1 format code 4 : Data security erase 5 : – 6 : – 7 : –
8-31	Sense bytes 8-31 are only used for hardware diagnostics

DOC – Display Operator Console

Sense byte	Designation
0	Bit 0 : command reject 1 : intervention required 2 : – 3 : equipment check 4 : – 5 : – 6 : – 7 : operation check

CHAPTER II
DOS/VSE GENERAL INFORMATION

Operation	Operand	Remarks
ADD	cuu[(k)],device-type[,ss|,ssss|,ssssss]	The ADD command is used to define the physical devices attached to the system. Note: The X''notation is no longer required but is still supported for compatibility reasons. cuu — Indicates the channel and unit number of the device to be added. k — Can be specified as either S or a decimal number from 0 to 255: S indicates that the device can be switched (that is, physically attached to two adjacent channels). The designated channel is the lower of the two channels. 0 to 255 indicates the priority of a device that cannot be switched, with 0 indicating the highest priority. If k is not given, the assumed priority is 255. device-type — Specifies the actual device (2400T9, 1443, etc.). See device type codes in Figure 1. ss ssss ssssss — Device specifications (see ASSGN Statement). If absent, the following values are assigned: C0 for 9-track tapes (2400, 3410 series) D0 for 9-track tapes (3420 series) 60 for 8809 Magnetic Tape Unit 90 for 7-track tapes 00 for non-tapes 00, 01, 02, and 03 are invalid as ss for magnetic tape. ss is required for a 3284/3286/3287 printer used as a hard-copy device for a 3277 operator console. The required entry is 02. ss specifies SADxxx (Set Address) requirements for IBM 2702 lines: 00 for SAD0 01 for SAD1 02 for SAD2 03 for SAD3 This information is not accepted on the ASSGN statements. ss is required for 1270, 1275, 1419, 1419P, and 1419S device types. It specifies the external interrupt bit associated with magnetic ink or optical character readers. The settings 01 through 20 correspond to the external interrupt code in low real storage byte 87, bits 7 through 2 respectively. The corresponding external lines to which the control units are attached are as follows: 01 byte 87 bit 7 02 byte 87 bit 6 04 byte 87 bit 5 08 byte 87 bit 4 10 byte 87 bit 3 20 byte 87 bit 2 The ss parameter specifies whether or not the error correction feature is present on an IBM 1018 Paper Tape Punch with 2826 Control Unit Model 1.

Operation	Operand	Remarks
ADD (Cont'd)		ss can be: 00 no error correction feature 01 error correction feature For the 3704 and the 3705, ss is required and must be specified as one of the following: 01 Type 1/4 channel adapter 02 Type 2/3 channel adapter 10 for 4300 ICA For the 2703 of the Model 115 or 125, ss, ssss, or ssssss is used to specify the line mode setting for a Start/Stop line or a BSC line. The bit settings of the line mode specification for each line involved are explained in IBM System /370 Model 115 Functional Characteristics, and in IBM System /370 Model 125 Functional Characteristics, repectively. The line mode setting is not accepted on the ASSGN statement. If a one or two byte value is specified, the specified value is right-justified and the rest of the three bytes is filled with zeros.
DEF	SYSREC=cuu,SYSCAT= {cuu|UA} [,SYSDMP=cuu]	The DEF command is used to assign SYSREC, the logical device for the system recorder and hard-copy files SYSCAT, the logical device for the VSAM-E master catalog SYSDMP, the logical device for the high-speed system dump. SYSCAT=cuu Indicates the channel and unit number of the physical device to be assigned to SYSCAT. SYSREC=cuu Indicates the channel and unit number of the physical device to be assigned to SYSREC. SYSDMP=cuu Indicates the channel and unit number of the physical device assigned to SYSDMP. SYSDMP can be assigned to any CKD or FBA device. The assignment cannot be changed until the next IPL.
DEL	cuu	The DEL command is used to delete a device from the PUB table. cuu Indicates the channel and unit number of the device to be deleted.
DPD	[TYPE= N|F], UNIT=cuu, CYL=n| BLK=n [,VOLID=xxxxxx]	The DPD command, which is mandatory, is used to define the pape data set. The operands of the DPD command may be given in any order. TYPE=N TYPE=N is the default and indicates that the page data set need not be formatted. If TYPE=N is specified but the page data set does not exist or the extent limits have been changed, TYPE=N is ignored and the page data set is formatted during IPL. TYPE=F Indicates that the page data set is to be formatted during IPL. Formatting during IPL is required if part of the formatted extent has been destroyed. The TYPE operand is ignored for FBA devices.

Operation	Operand	Remarks
DPD Cont'd		UNIT=cuu Specifies the channel and unit number of the device that is to contain the page data set. CYL=n Specifies, for CKD devices, the sequential number of the cylinder, relative to zero, where the page data set is to begin (in decimal). BLK=n Specifies, for FBA devices, the sequential number of the block, relative to zero, where the page data set is to begin (in decimal). VOLID=xxxxxx Identifies the volume serial number (one to six alphabetic or numeric characters) of the disk pack that contains the page data set (for label checking). If this operand is omitted, the volume serial number is not checked.
SET	[DATE=value 1, CLOCK=value 2] [,ZONE={EAST / WEST}/hh/mm]	value 1 In one of the following formats: mm/dd/yy or mm: month (01-12) dd: day (01-31) yy: year (00-99) value 2 In the following format: hh/mm/ss, hh: hours (00-23) mm: minutes (00-59) ss: seconds (00-59) EAST Specifies a geographical position east of Greenwich. WEST Specifies a geographical position west of Greenwich. hh/mm A decimal value which indicates the difference in hours and minutes between local and Greenwich Mean Time. hh : 0-23 mm: 0-59 Note Required statement. If any ADD or DEL commands are required, they must precede the SET command.
SVA	[SDL=n][,PSIZE=nK][,GETVIS=nK]	SDL=n Specifies the decimal number of entries in the system directory list to be reserved for user phases and non SVA-eligible IBM-supplied phases, in addition to the phases loaded automatically during IPL. For a list of those phases that are automatically loaded into the SVA during IPL, refer to DOS/VSE System Generation. Do not specify entries for these phases, as this is done by IPL. The maximum number that can be specified is 963.

Operation	Operands	Remarks
		PSIZE=nK Specifies the size of the area within the SVA which is to be reserved for user phases. n must be a decimal number and a multiple of 2. The specified size should be large enough for the user phases and for a maintenance area which is required when a phase in the system core image library, with a copy in the SVA, is replaced. Do not specify space for the phases loaded automatically into the SVA during IPL, as IPL will reserve the necessary space. GETVIS=nK Indicates the size of the additional system GETVIS area which you can specify beyond the minimum size allocated by the system. n must be a decimal number and a multiple of 2. Note that SET, DPD, and DEF may be entered at any time during IPL. ADD and DEL must precede any DEF, DLA, DPD, or SVA command(s). SVA must be the last IPL command.

JOB CONTROL - AND ATTENTION ROUTINE COMMANDS

Job Control Overview

Type of Command or Statement	Operation	Valid for		
		JCS	AR	JCC
Job Identification	JOB	x		
	/ &	x		
	/	x		
User Identification	ID	x		
File Definition	DLBL	x		
	EXTENT	x		
	TLBL	x		
	/*	x		
Pass Information to Operator	*	x		
Pass Information to Program	DATE	x		.
	LBLTYP	x		.
	OPTION	x		.
	OVEND	x		x
	UPSI	x		
Job Stream Control	BATCH	.	x	.
	CANCEL	.	x	x
	PAUSE	x	x	x
	PRTY		x	.
	START		x	.
	STOP		.	x
	TPBAL		x	.
	UNBATCH			Note
Setting System Parameters	ALLOC		x	x
	ALLOCR		x	x
	SET		.	x
	SIZE		x	x
	STDOPT	x	.	x

Note: Valid only in a Foreground Partition

Name	Operation	Operand	Remarks	Accepted by
	ALLOC	Fn=mK [,Fn=mK]	n Indicates the number of the foreground partition. m Indicates the amount of storage to be allocated to the specified foreground partition. m must not be smaller than 128.	JCC AR
	ALLOCR	PARTITION=mK [,PARTITION=mK]...	partition Indicates the partition (BG, F1, F2, ...) to which storage is to be allocated. For compatibility reasons, the partition specifications BGR and FnR will also be accepted. m Indicates the amount of storage to be allocated to the specified partition. m should be an even integer. any uneven specification is rounded up to the nearest even integer. m may also be zero.	JCC AR
	ALTER	XXXXXX	Alters 1 to 16 bytes of virtual storage. XXXXXX is the hex address where alteration is to start.	AR

Name	Operation	Operand	Remarks	Accepted by
[//]	ASSGN	For any device: SYSxxx, cuu / UA / IGN / (address-list) / SYSyyy	For remarks see end of this statement [,TEMP / ,PERM]	JCS JCC
		For disks: cuu / (address-list) / SYSyyy / DISK / FBA / 2311 / 2314 / 3330 / 3330B / 3340 / 3340R / 3350 SYSxxx,	[,TEMP / ,PERM] [, VOL=volserno][,SHR]	
		For diskettes: SYSxxx, cuu / (address-list) / SYSyyy / DISKETTE / 3540	[,TEMP / ,PERM] [,VOL=volserno]	
		For tapes: SYSxxx, cuu / (address-list) / SYSyyy / TAPE / 2400T7 / 2400T9 / 3410T7 / 3410T9 / 3420T7 / 3420T9 / 8809	[,ss / ,ALT] [,TEMP / ,PERM] [,VOL=volserno]	
	ASSGN (Cont'd)	For printers: SYSxxx, cuu / (address-list) / SYSyyy / PRINTER / PRT 1 / 1403 / 1403U / 1443 / 3203 / 3211 / 3800 / 3800B / 3800C / 3800BC / 5203 / 5203U	For remarks see end of this statement [,TEMP / ,PERM]	

Name	Operation	Operand	Remarks	Accepted by
	ASSGN (Cont'd)	For card (read) punches: SYSxxx, { cuu / (address-list) / SYSyyy / PUNCH / 1442N1 / 1442N2 / 2520B1 / 2520B2 / 2520B3 / 2540P / 2560 [,H1 ,H2] / 2596 / 3525P / 3525RP / 5424 [,H1] / 5425 [,H2] } [,TEMP ,PERM] For card readers: SYSxxx, { cuu / (address-list) / SYSyyy / READER / 1442N1 / 2501 / 2520B1 / 2540R / 2560 [,H1 ,H2] / 2596 / 3504 / 3505 / 3525RP / 5424 [,H1] / 5425 [,H2] } [,TEMP ,PERM]		

SYSxxx	can be SYSRDR, SYSIPT, SYSIN, SYSPCH, SYSLST, SYSOUT, SYSLOG, SYSLNK, SYSRLB, SYSSLB, SYSCLB or SYS000-SYS 240. SYSCAT and SYSREC can only be assigned by the DEF command at IPL time. For compatibility reasons, an assignment for SYSREC entered from SYSRDR is ignored and processing continues; if entered from SYSLOG, the assignment is rejected.
cuu	c=channel number uu=unit number
address-list	a list of up to seven device addresses in the form: (cuu, . . . , cuu)
UA	unassign
IGN	unassign and ignore (invalid for SYSCLB, SYSRDR, SYSIPT, SYSIN)
SYSyyy	any system or programmer logical unit.
device-class	READER, PRINTER, PUNCH, TAPE, DISK, CKD, FBA, or DISKETTE
device-type	device code of any supported device

Name	Operation	Operand	Remarks	Accepted by
	ASSGN (Cont'd)		X ss density (magn. tape only) ss BPI Parity Transl. feat Conv. feat 10 200 odd off on 20 200 even off off 28 200 even on off 30 200 odd off off 38 200 odd on off 50 556 odd off on 60 556 even off off 68 556 even on off 70 556 odd off off 78 556 odd on off 90 800 odd off on A0 800 even off off A8 800 even on off B0 800 odd off off B8 800 odd on off C0 800 single dens. 9 tr. C0 1600 single dens. 9 tr. C0 1600 dual dens. 9 tr. C8 800 dual dens. 9 tr. D0 6250 single/dual dens. 9 tr. D0 6250 dual dens. 9 tr. 90 1600 IBM 8809 STREAMING: High speed and long gap 30 1600 IBM 8809 STREAMING: High speed and short gap 50 1600 IBM 8809 START STOP: Low speed and long gap 60 1600 IBM 8809 START STOP: Low speed and short gap ALT specifies alternate tape unit. (Invalid for SYSIPT) H1 specifies input hopper 1 for input on 2560, 5424 or 5425; If neither H1 nor H2 is specified, H1 is assumed H2 specifies input hopper 2 for input on 2560, 5424 or 5425; (invalid for programmers units) PERM the assignment is permanent TEMP the assignment is temporary VOL= volserno volume serial number of the tape, DISK or DISKETTE. SHR indicates the shared option for disk devices	
	BATCH	[BG/Fn]	Start or continue processers	AR
	CANCEL	{BG | Fn}[,NODUMP|,PARTDUMP|,DUMP][,SYSDMP|,NOSYSDMP]	Cancels execution of current job in specified area BG Indicates that the background job is to be canceled. Fn Indicates that the specified foreground job is to be canceled. NODUMP Indicate the dump option which is PARTDUMP to be in effect until end of job. DUMP The meaning of the keywords is the NOSYSDMP same as in the OPTION statement.	AR

Name	Operation	Operand	Remarks	Accepted by
			cuu — Indicates that the I/O request for the specified device and the associated task is to be canceled. Warning: Use only in emergency cancel situations (if, for example, 1I40D REQUEST CANCEL has been issued), because you may also cancel POWER/VS. blank — Cancels execution of current job	JCC
[//]	CLOSE	[SYSxxx / ,cuu [,ss] / ,UA / ,IGN / ,ALT / ,SYSyyy]	SYSxxx for magnetic tape SYSPCH SYSLST SYSOUT SYS000-SYS240 for DASD (JCC only) SYSIN SYSRDR SYSIPT SYSPCH SYSLST cuu, ss, UA, IGN, ALT Values as described in ASSGN command.	JCS JCC
//	DATE	mm/dd/yy or dd/mm/yy	mm month (01-12) dd day (01-31) yy year (00-99)	JCS
//	DLBL	filename, [file-ID], [date], [codes], [,DSF][,BUFSP=n] [,CAT=filename] [,BLKSIZE=n] [,CISIZE=n]	filename — One to seven alphameric characters, the first of which must be alphabetic file-ID — One to forty-four alphameric characters (one to eight alphameric characters for the 3540 diskette) date — One to six characters (yy/ddd) codes — Two to four alphabetic characters (SD, DA, DU, ISC, ISE, VSAM) DSF — specifies that a data secured file is to be created or processed BUFSP=n — specifies, for a VSAM file to be processed, the number of bytes of vitual storage (0-999999) to be allocated as bufferspace CAT=filename — specifies filename (1 to 7 alphameric characters) of the DLBL statement for the catalog owing this VSAM file. BLKSIZE=n — a number from 1 to 32,768 (only valid for SAM files on 3350 and 3330-11) CISIZE=n — This operand permits specification of an FBA control interval size for SAM and DAM files or FBA devices in order to improve space utilization on such devices. A number from 1 to 32,768.	JCS
	DSPLY	ADDRESS	Displays 16 bytes of virtual storage	AR
	DUMP	[S / BG / Fn / BGS / FnS / SVA / SVAS / BUFFER / addr,addr / ALL] ,cuu[RUN]	The DUMP command allows the operator to DUMP part or all of the real and/or virtual address area contents on a printer, tape, or disk device (CKD or FBA) Note: There must be no blank between the operands. S — Same as above however, the contents of the supervisor area are also dumped.	

Name	Operation	Operand	Remarks	Accepted by
	DUMP (Cont'd)		BG / Fn — The contents of the specified partition and its associated registers are dumped. If a program is running in real mode in the specified partition, only the real, not the associated virtual partition, is dumped. If a program is running in virtual mode in the specified partition, the virtual parition is dumped, including any fixed pages. The areas acquired through GETVIS in the partition are also dumped. BGS / FnS — Same as if BG or Fn was specified; however, the contents of the supervisor area are also dumped. SVA — The contents of the SVA are dumped. SVAS — The contents of the SVA and of the supervisor are dumped. BUFFER — The contents of the SDAID buffer are dumped on tape or disk. This operand is rejected in case of output to the printer. address,address — Specifies storage are between the two hexadecimal addresses and associated registers. cuu — Specifies the device on which the output is to be written. run — Specifies that the system is to continue to run during DUMP command execution. ALL — The contents of the supervisor, the SVA, and all partitions are dumped.	
	DVCDN	cuu	cuu — c=0-6, uu=00-FE(0-254) in hex	JCC
	DVCUP	cuu	cuu — c=0-6, uu=00-FE(0-254) in hex	JCC
	END or ENTER	blank	End of SYSLOG communications END for the 3210 and 3215 printer keyboards ENTER for DOC	JCC AR
[//]	EXEC	{[[PGM=] progname][,REAL][,SIZE=size]} / {PROC=procname [,OV]}	PGM=progname — one to eight alphameric characters. Used only if the program is in the core image library REAL — The respective program is to be executed in real mode SIZE=size — can be nK, AUTO, or (AUTO,nK) nK: size of area required AUTO: take program size (AUTO,nK): take program size plus nK PROC=procname — Name of cataloged procedure to be retrieved. One to eight alphameric characters, the first of which must be alphabetic OV — Indicates that overwrite statements follow EXEC statement	JCC JCS

Name	Operation	Operand	Remarks	Accepted by
//	EXTENT	[symbolic-unit], [serial-number],[type]. [sequence-number], [relative-track | block], [number-of-tracks | blocks], [split-cylinder-track]	symbolic unit: Six alphameric characters serial number: One to six alphameric characters type: One numeric character sequence number: One to three numeric characters relative track block: One to five numeric characters (CKD-DEVICES) number of tracks/blocks: One to five numeric characters (CKD-DEVICES) split cylinder track: One or two numeric characters	JCS
	HOLD	Fn [,Fn] . . .	Causes the assignments for the specified foreground partition(s) to remain in affect until the end of the next job	JCC
	IGNORE	blank	Ignore abnormal condition	AR JCC
//	JOB	jobname [accounting information]	jobname: One to eight alphameric characters accounting information: One to sixteen characters	JCS
//	LBLTYP	{ TAPE[(nn)] } { NSD (nn) }	TAPE Used when tape files requiring label information, are to be processed and no nonsequential disk files are to be processed (nn) Optional and is present only for future expansion (ignored by job control) NSD Nonsequential disk files are to be processed (nn) Largest number of extents per single file	JCS
	LFCB	cuu, phasename [,FORMS=xxxx] [,LPI=n][,NULMSG]	Causes the FCB of printer cuu to be loaded	AR
[//]	LISTIO	{ ASSGN SYS PROG Fn ALL SYSxxx UNITS DOWN UA cuu BG }	Causes listing of I/O assignments on SYSLST for JCS and SYSLOG for JCC	JCS JCC
	LOG	blank	Causes logging of job control statements on SYSLOG	JCC AR
	LUCB	cuu, phasename [,FOLD][,NOCHK] [,TRAIN=xxxxxx] [,NULMSG]	Causes the UCB of printer cuu to be loaded	AR
	MAP	blank	Causes a map of area in real and virtual storage to appear on SYSLOG	JCC AR

Name	Operation	Operand	Remarks	Accepted by
	MODE	{ IR CR CE, cuu [,I [,xx,y] ,D [,xx,y] ,N] R STATUS HIR ECC [,{M C}] { [,{R Q TH}] [,E=eeee][,T=tttt] } }	Changes the mode of operation, changes the EFL threshold values and gives status information. Note: When HIR or ECC is specified, at least one of the optional operands within these braces must be selected. TH is inly valid for the Model 145 when ECC, C is specified with the MODE command.	AR
	MSG	{BG | Fn}	Transfers control to message routine	AR
//	MTC	opcode, {SYSxxx cuu [,nn]}	opcode BSF, BSR, DSE, ERG, FSF, FSR, REW, RUN, or WTM SYSxxx Any logical unit cuu Specifies the channel and unit number (in hex) nn dec. number (01-99)	JCS JCC
	NEWVOL	[BG|Fn][,ignore]	Indicates that a new volume has been mounted for the specified partition	AR
	NOLOG	blank	Suppresses logging of job control statements on SYSLOG	JCC AR
//	OPTION	option [,option] . . .	option: can be any of the following LOG Log control statements on SYSLST NOLOG Suppress LOG option DUMP Dumps the registers, supervisor area, partition, the used part of the system GETVIS area, and the SVA phase in error if the error occured in the SVA. The dump will be recorded on SYSLST, if assigned, in the case of an abnormal program end (such as program check). PARTDUMP Dump registers, selected supervisor control blocks, and temporary real or virtual partition on SYSLST in case of abnormal program end. NODUMP Suppress DUMP or PARTDUMP option LINK Write output of language translator on SYSLNK for linkage editing NOLINK Suppress LINK option	JCS

Name	Operation	Operand	Remarks	Accepted by
	OPTION (Cont'd)		DECK Output object module on SYSPCH NODECK Suppress DECK option EDECK Punch source macro definitions on SYSPCH NOEDECK Suppress EDECK option ALIGN Align constants and date areas on boundaries NOALIGN Suppress ALIGN option LIST Output listing of source module on SYSLST NOLIST Suppress LIST option LISTX Output of object module on SYSLST SYM Produce symbol table or data division map on SYSLST NOSYM Suppress SYM option XREF Output symbolic cross-reference list on SYLST SXREF The assembler writes the simbolic cross-reference list on SYSLST; printing of all unreferenced labels is suppressed NOXREF Suppress XREF or SXREF option ERRS Output listing of all errors in source program on SYSLST NOERRS Suppress ERRS option RLD Output listing of RLD information on SYSLST NORLD Suppress RLD option ACANCEL Cancel job if attempt to assign device is unsuccessful NOACANCEL Await operator action if a device cannot be assigned CATAL Catalog program or phase in core image library after completion of Linkage Editor run STDLABEL Causes all DASD or tape labels to be written on the standard label track USRLABEL Causes all DASD or tape labels to be written on the user label track PARSTD Causes all DASD or tape labels to be written on the partition standard label track 48C 48 - character set 60C 60 - character set	

Name	Operation	Operand	Remarks	Accepted by
	OPTION (Cont'd)		SYSPARM=string specifies a value for assembler system variable symbol and SYSPARM SUBLIB=DF Causes assembler and ESERV program to retrieve nonedited macros and copy-books from D-sublibrary and edited macros from the F-sublibrary of the source statement library SUBLIB=AE Causes assembler and ESERV program to retrieve nonedited macros from the E-sublibrary and edited macros from the E-sublibrary of the source statement library ONLINE Causes fetching of all programs for execution from the system core image library, although a private core image library is assigned SYSDMP Indicates that dumps are to be written on SYSDMP NOSYSDMP Indicates that dumps are to be written on SYSLST	
[//]	OVEND	[comments]	Indicates end of overwrite statements for a cataloged procedure	JCS JCC
[//]	PAUSE	[comments]	Causes pause immediately after processing this statement. PAUSE statement is always printed on SYSLOG. If no 3210, 3215 or DOC is available the statement is ignored.	JCS JCC
	PAUSE	{BG / Fn} [,EOJ] where n=1,2,3 or 4	Causes pause at end of current job step or at end of job	AR
	PRTY	blank partition [,partition]	Pn=BG, F1, F2, F3 or F4. Allows the operator to display or change the priority of partitions	AR
[//]	RESET	{SYS / PROG / ALL / SYSxxx}	Resets I/O device assignments	JCS JCC
	ROD	blank	Causes all SDR counters for all non-teleprocessing devices on the recorder file on SYSREC to be updated from the SDR counters in main storage	JCC
//	RSTRT	SYSxxx,nnnn[,filename]	SYSxxx Symbolic unit name of the device on which the checkpoint records are stored. Can be SYS000-SYSnnn nnn four character identification of the checkpoint record to be used for restart filename symbolic name of the DASD file to be used for restarting	JCS

Name	Operation	Operand	Remarks	Accepted by
	SET	[,UPSI=value 1] [,LINECT=value 2] [,RCLST=value 3] [,RCPCH=value 4] [,RF=value 5] [,DATE=value 6] [,HC=value 7]	value 1: 0, 1 or X value 2: standard number of lines for output on each page of SYSLST value 3: decimal number indicating minimum number of SYSLST disk records remaining to be written before operator warning value 4 decimal number indicating minimum number of SYSPCH disk records remaining to be written before operator warning value 5 defines to the system the status of the recorder file (IJSYSREC) on SYSREC used by the RMSR feature RF={YES / CREATE} YES-file exists, CREATE-create file value 6 in one of the following formats: mm/dd/yy or dd/mm/yy mm: month (01-12) dd: day (01-31) yy: year (00-99) value 7 HC={YES / NO / CREATE} YES: hard-copy file exists NO: no recording performed CREATE: create a hard-copy file	JCC
		[,SDL]	This operand makes it possible to ADD phase names to the system directory list and, optionally, to load phases into the SVA	
	SETDF	{3800 | cuu} [,BURST=[YIN]] [,CHARS=[table name]] [,FCB=[fcb name]] [,FLASH=[overlay name]] [,FORMS=[forms name]] [,LIST] [,MODIFY=[copymod name]] [,RESET]		AR
	SETMOD	cuu [,MODE]	The SETMOD command, valid for the 3401 magnetic tape unit cuu Specifies the channel and unit number of the 3401 mode Can be one of the following: 90 or HL — high speed and long gap (streaming) 30 or HS — high speed and short gap (streaming) 50 or LL — low speed and long gap (start-stop) 60 or LS — low speed and short gap (start-stop) If the mode operand is omitted, the default mode setting of 60 (or LS) is assumed.	AR

Name	Operation	Operand	Remarks	Accepted by
	SETPRT	SYSxxx[,BURST={N|Y|*}] [,CHARS={table name|* (table name, . . .)}] [,COPIES=number][,DCHK={B|U}] [,DEBUG={NORM TERM DUMP TRAC}] [,DFLT={N|Y}] [,FCB={fcb name|* (fcb name, V) (*,V)}] [,FLASH={overlay name|*|(,count) (overlay name[,{count|255}]) (*[{count|255}])}] [,FORMS={forms name|*}][,INIT={N|Y}] [,MODIFY={copymod name|* (copymod name, table name)}] [,SEP=O][,TRC={N|Y}]		JC
	SIZE	partition=mK [,partition=mK] . . .		JCC, AR
	START	{BG Fn}	Same as BATCH	AR
	STDOPT	option [,option] ...	The STDOPT command can be used to reset in all partitions the permanent job control options which were established at system initialization. The command can only be given in the background partition. The options, which can appear in any order, are as follows (the first keyword is always the default value): ALIGN yes/no Specifies if the assembler is to align data on half-word or fullword boundaries, according to the type of instruction used. A supervisor must be assembled with ALIGN=YES to avoid hard waits. ACANCEL No/yes Specifies if job control is to cancel jobs automatically (ACANCEL=YES) or to wait for operator intervention (ACANCEL=NO) after an unsuccessful attempt to assign a device. (Note that the LOG command suppresses the ACANCEL function). CHARSET 48C|60C Specifies either the 48- or 60-character set for PL/I translator input on SYSIPT DATE MDY|DMY Specifies the format of the date: MDY=month/date/year. DMY=day/month/year.	JCC JCS

Name	Operation	Operand	Remarks	Accepted by
	STDOPT (Cont'd)		DECK YES|NO Specifies if language translators are to produce object modules on SYSPCH DUMP YES|NO|PART Specifies if a dump of the registers and virtual storage is to be written on SYSLST in case of an abnormal program end. PART specifies that a dump of the supervisor control blocks and the virtual storage of the partition is to be written on SYSLST EDECK NO|YES Specifies if the assembler is to create and punch edited macros on SYSPCH ERRS YES|NO Specifies if compilers are to summarize all errors in source programs on SYSLST. Assembler and PL/I always assume ERRS=YES. LINES 56|nn Specifies the number of lines per page on SYSLST. The minimum is 30, the maximum is 99. (If job control is running in another partition at the same time, the new value becomes effective in that partition when the next page is started.) LIST YES|NO Specifies if language translators are to write source module listings and diagnostics on SYSLST. LISTX NO|YES Specifies if compilers are to write hexadecimal object module listings on SYSLST. LOG YES|NO Specifies if all job control statements are to be listed on SYSLST. Invalid statements and commands will be listed on SYSLST if it is assigned. RLD NO|YES Specifies if the relocation dictionary information is to be printed.	

Name	Operation	Operand	Remarks	Accepted by
	STDOPT (Cont'd)		SXREF NO|YES Specifies whether the assembler is to print short cross-reference lists on SYSLST. The printing of unreferenced labels is suppressed instead. SXREF=YES forces XREF=NO. SYM NO|YES SYM=YES specifies that the PL/I compiler is to produce a symbol and offset table listing on SYSLST, or that American National Standard Cobol is to produce a data division glossary. SYSDMP NO|YES YES indicates that dumps are to be written on SYSDMP and NO that they are to be written on SYSLST. TERM NO|YES Specifies whether messages from a compiler are to be displayed on SYSLOG. XREF YES|NO XREF=YES specifies that the assembler is to write symbolic cross-reference lists on SYSLST, or that American National Standard COBOL is to produce a cross-reference listing. XREF=YES forces SXREF=NO.	
	STOP	blank	Stops batched-job progr. processing	JCC
//	TLBL	filename, ['file-ID'], [date], [file serial number], [volume sequence number], [file sequence number], [generation number], [version number] Note: For ASCII file processing the fourth and fifth operands are called set identifier and file section number, respectively	filename: One to seven alphameric characters, the first of which must be alphabetic 'file-ID': One to seventeen alphameric characters date: One to six characters (yy/ddd or d-dddd) {[file serial number (EBCDIC): One to six alphameric characters] [set identifier (ASCII): Six alphameric characters]} {[volume sequence number (EBCDIC)] [file section number (ASCII)]} One to four numeric characters file sequence number: One to four numeric characters generation number: One to four numeric characters version number: One to two numeric characters	JCS

Name	Operation	Operand	Remarks	Accepted by
	TPBAL	[n]	n = number of partitions in which processing can be delayed (0, 1, 2, . . , number of partitions minus one). Allows the operator to display or alter the status of the Teleprocessing Balancing function.	AR
	UCS	SYSxxx, phasename [,FOLD][,BLOCK] [,NULMSG]	Causes the 240-character universal character set contained in the core image library phase specified by phasename to be loaded as buffer storage in the IBM 2821 CU. SYSxxx must be assigned to a 1403 Printer with the UCS feature.	JCC
	UNBATCH	blank	Terminates foreground processing	JCC
//	UPSI	nnnnnnnn	n 0, 1 or X	JCS
//	VOLUME	[c|cu|cuu]	The VOLUME command provides the operator with a short summary of the volumes mounted on DASD devices, together with an indication of whether or not a volume is in use.	AR
//	ZONE	{EAST / WEST} /hh/mm	EAST A geographical position east of Greenwich WEST A geographical position west of Greenwich hh/mm A decimal value which indicates difference in hours and minutes between local time and Greenwich Mean Time. hh may be in the range 0-12; mm in the range 0-59.	JCS
/+	ignored	[comments]	Indicates end of procedure	JCS
/*	ignored	ignored	Columns 1 and 2 are the only columns checked	JCS
/&	ignored	[comments]	Columns 1 and 2 are the only columns checked. Comments appear on SYSLOG and SYSLST at EOJ.	
*		comments	Column 2 must be blank	

JOB CONTROL- AND ATTENTION ROUTINE COMMANDS (. . . . Cont'd)

Type of Command or Statement	Operation	Valid for		
		JCS	AR	JCC
Operator Communications	ALTER	.	x	.
	DSPLY	.	x	.
	DUMP	.	x	.
	END or	.	.	.
	ENTER key	.	x	x
	IGNORE	.	x	x
	LOG	.	x	x
	MAP	.	x	x
	MSG	.	x	.
	MODE	.	x	.
	NEWVOL	.	x	.
	NOLOG	.	x	x
	REPLID	.	x	.
	SETMOD	.	x	.
	ZONE	x		
Control of I/O System	ASSGN	x	.	x
	CLOSE	x	.	x
	DVCDN	.	.	x
	DVCUP	.	.	x
	HOLD	.	.	x
	LFCB	.	x	.
	LISTIO	x	.	x
	LUCB	.	x	.
	MTC	x	.	x
	RESET	x	.	x
	ROD	.	.	x
	SETDF	.	x	.
	SETPRT	x	.	x
	UCS		.	x
	VOLUME		x	
Execution of Program	EXEC	x		x
	RSTRT	x		

JOB CONTROL STATEMENTS SUMMARY

ASSGN	Used at execution time to assign a specific device address to the symbolic unit name used.
CLOSE	Closes either a system or a programmer logical unit assigned to tape, disk, or diskette.
DATE	Contains a date that is put in the communications region.
DLBL	Contains file label information for DASD or diskette label checking and creation.
EXEC	Indicates the end of job control statements for a job step and that the job step is executed.
EXTENT	Defines each area, or extent, of a DASD file or diskette volume.
ID	Used to specify user identification and password.
JOB	Indicates the beginning of control information for a job.
LBLTYP	Defines the amount of storage to be reserved at link-edit time for processing tape and nonsequential DASD file labels in the partition.
LISTIO	Used to get a listing of I/O assignments on SYSLOG or SYSLST.
MTC	Controls operations on logical units to IBM 2400/3400 series magnetic tapes.
OPTION	Specifies one or more of the job control options.
OVEND	Indicates that no more overwrite statements will follow for the respective procedure.
PAUSE	Causes a pause immediately after processing this statement.
RESET	Resets I/O assignments to the standard assignments.
RSTRT	Restarts a checkpointed program.
SETPRT	Loads the IBM 3800 buffers.
STDOPT	Resets systems defaults.
TLBL	Contains file label information for tape label checking and writing.
UPSI	(User Program Switch Indicators.) Allows the user to set program switches that can be tested.
ZONE	Initializes the zone field in the communications region.
/*	Indicates the end of a data set or the end of a job step.
/&	Indicates the end of a job.
*	Job control comments.
/+	Indicates the end of a procedure.

Programming support continues for the following job control statements provided in previous versions of the system (they should, however, not be used for new applications):

DLAB	Contains file label information for DASD label checking and creation.
TPLAB	Contains file label information for tape label checking and writing.
VOL	Used when a set of label information for a magnetic tape file or a DASD file is specified. It is not required with the current DLBL, EXTENT, or TLBL statements.
XTENT	Defines each area, or extent, of a DASD file. It is used in conjunction with the VOL and DLAB statements.

Name	Operation	Operand	Remarks
blank	PHASE	name, origin[,NOAUTO][,SVA][,PBDY]	name: Symbolic name of the phase. One to eight alphameric characters origin: Specifies the load address of the phase. Load address can be in one of the following formats: 1) symbol[(phase)] [±relocation] 2) *[± relocation] } 3) S[+ relocation] } * * addresses relativ to begin of virtual partition 4) ROOT 5) +displacement } 6) F +address } ** ** absolute addresses Note: A phase is eligible for relocation by the relocating loader if its origin is specified as a relative address (formats 1-4 above). However, if a phase is relative to another phase whose origin is specified as an absolute address (formats 5 or 6 above), none of the phases can be made relocatable during this linkage editor execution. Refer to ACTION statement for additional information about the relocating loader. NOAUTO Indicates that the automatic library lookup (AUTOLINK) feature is suppressed for both the private- and system relocatable libs. SVA Indicates that the phase is SVA-eligible. PBDY Indicates that the phase is to be link-edited on a page boundery.
blank	INCLUDE	[modulename][,(name list)]	If both operands are omitted the object module to be included is assumed to be on SYSIPT. If the first operand is present, the object module is assumed to be in either the private- or the system relocatable library. If the first operand is omitted and the second operand is present, the object module to be included is assumed to be in the input stream (SYSLNK). modulename: Symbolic name of the module as used when cataloged in the relocatable library. It consists of one to eight alphameric characters. (namelist): The Linkage Editor constructs a phase from only the control sections specified. The namelist is in the following format: (cs name 1, cs name 2, ..) Entries within the parentheses are the names of the control sections that are used to constitute the phase.

Name	Operation	Operand	Remarks
blank	ENTRY	[entrypoint]	entrypoint Symbolic name of an entry point. If the operand field is blank, the Linkage Editor uses as transfer address the first significant address provided in an END record encountered during generation of the first phase.
blank	ACTION	{REL / NOREL} [,CLEAR] [,MAP / ,NOMAP] [,NOAUTO] [,Cancel] [,BG / ,Fn]	Indicates Linkage Editor options: CLEAR Indicates that the unused portion of the core image library will be set to binary zero before the beginning of the Linkage Editor function. MAP Indicates that SYSLST is available for diagnostic messages. In addition, a virtual storage map is printed on SYSLST. NOMAP Indicates that SYSLST is not available when performing the linkedit function. NOAUTO Indicates that the AUTO LINK function is to be suppressed. CANCEL Cancels the job automatically if any of the errors 2100I through 2170I occur. BG Fn Sets the end-of-supervisor address used in Linkage Editor calculation to the beginning of the partition specified, plus the length of the label area and of the save area. If none of these operands are present, the program is linkedited to execute in the virtual partition in which linkedit takes place, unless otherwise specified in the PHASE statement. REL Indicates that the phase(s) produced during this execution of the Linkage Editor is to be made relocatable if possible. Refer to origin operand in PHASE statement. Note: If support for the relocating loader was generated in the supervisor, ACTION REL is the default. NOREL Indicates that the phase(s) produced during this execution of the Linkage Editor is not to be made relocatable. Note: If support for the relocating loader was not generated in the supervisor, ACTION NOREL is the default.

Maintenance Functions

Function	Unit	Element	Control Statements

Catalog	Cor Image Library	Phase	// OPTION CATAL
			(Linkage Editor control statements and if in card form, the phase to be cataloged)
			/*
			// EXEC LNKEDT

	Relocatable Library	Module	// EXEC MAINT CATALR modulename [,v,m] (module to be cataloged)
	Source statem. Library	Book	// EXEC MAINT CATALS sublib.bookname [,v,m],c]] (book to be cataloged)
	Procedure Library	Proce-dure	// EXEC MAINT CATALP procedurename [,VM=v.m] [,EOP=yy] [,DATA= NO / YES] (procedure to be cataloged) /+(or delimiter as specified in EOP parameter)
Delete	Core Image Library	Phase	// EXEC MAINT DELETC phase 1 [,phase 2, ...]
		Program	// EXEC MAINT DELETC prog1.ALL[,prog2.ALL, ...]
	Relocatable Library	Module	// EXEC MAINT DELETR module 1[,module2, ...]
		Program	// EXEC MAINT DELETR prog1.ALL[,prog2.ALL, ...]
		Library	// EXEC MAINT DELETR ALL
	Source Statement Library	Book	// EXEC MAINT DELETS sublib.book1[,sublib.book2, . .]
		Sub Library	// EXEC MAINT DELETS sublib.ALL
		Library	// EXEC MAINT DELETS ALL
	Procedure Library	Proce-dure	// EXEC MAINT DELETP procedurename[,procedure-name2, ...]
		Library	// EXEC MAINT DELETP ALL
Rename	Core Image Library	Phase	// EXEC MAINT RENAMEC oldname, newname[,oldname, newname, ...]
	Relocatable Library	Module	// EXEC MAINT RENAMR oldname, newname [,oldname, newname, ...]
	Source Statement Library	Book	// EXEC MAINT RENAMS sublib.oldname,sublib.newname [,sublib.oldname,sublib.newname, ...]
	Procedure Library	Proce-dure	// EXEC MAINT RENAMP oldname,newname[,oldname, newname, ...]
Update	Source Statement Library	Book	// EXEC MAINT UPDATE sublib.bookname,[s.book1], [v.m],[nn]) ADD,) DEL, or) REP statements as required with source statements to be added) END [v.m],C]]

Function	Unit	Element	Control Statements
Condense	Core Image Library	Library	// EXEC MAINT CONDS CL

	Relocatable Library	Library	// JOB jobname
			// EXEC MAINT
			CONDS RL

	Source Statement Library	Library	// EXEC MAINT CONDS SL
	Procedure Library	Library	// EXEC MAINT CONDS PL
	Libraries	All	// EXEC MAINT CONDS CL, RL, SL, PL
Set Parameter for Automatic Condense	Libraries	Any or All	// EXEC MAINT CONDL lib=nnnnn[,lib=nnnnn[,lib=nnnnn]] Notes: Values to be substituted for lib: CL – Core image library RL – Relocatable library SL – Source statement library PL – Procedure library Values to be substituted for nnnnn: for CKD devices one to five decimal digits with a maximum value of 65536 for FBA devices, one to nine digits.

Reallocation	System	Library	// DLBL IJSYSRS, 'DOS SYSTEM RESIDENCE FILE' date, code
			// EXTENT SYSRES, balance of extent information
			// EXEC MAINT
			ALLOC CL=cylin(tracks),RL=cylin(tracks), SL=cylin(tracks),PL=cylin(tracks)
			Notes:
			CL – Core image library
			RL – Relocatable library
			SL – Source ststement library
			PL – Procedure library
			For FBA devices, CYLIN and TRACKS are to be substituted by blocks.

Note: // JOB, /* and /& must be included where needed

Service Functions

Display Unit	Element	Control Statements
Core Image Library	Phase	// EXEC CSERV DSPLY phase1[,phase2, ...]
	Program	// EXEC CSERV DSPLY prog1.ALL[,prog2.ALL, ...]
	Library	// EXEC CSERV DSPLY ALL
	Directory	// EXEC DSERV DSPLY CD or DSPLYS CD
	Phase(s) with Version and Modification Level	In the standard position: // EXEC DSERV DSPLY[S] CD(phasename) or CD(phasename) In the nonstandard position or higher than DSERV in use: // EXEC DSERV DSPLY[S] CD(phasename,nn) or CD(phasename,nn)

Display Unit	Element	Control Statements
Relocatable Library	Module	// EXEC RSERV DSPLY module1[,module2, ...]
	Program	// EXEC RSERV DSPLY prog1.ALL[.prog2.ALL, ...]
	Library	// EXEC RSERV DSPLY ALL
	Directory	// EXEC DSERV DSPLY RD or DSPLYS RD
Source Statement Library	Book	// EXEC SSERV DSPLY sublib.book1[.sublib.book2, ...]
	Sublibrary	// EXEC SSERV DSPLY sublib1.ALL[,sublib2.ALL, ...]
	Library	// EXEC SSERV DSPLY ALL
	Directory	// EXEC DSERV DSPLY SD or DSPLYS SD
	Macro-Sublibrary	// EXEC ESERV GENEND DSPLY E.book 1 (,E.book2, . . .)
Procedure Library	Procedure	// EXEC PSERV DSPLY procedurename1[,procedurename2, ...]
	Library	// EXEC PSERV DSPLY ALL
	Directory	// EXEC DSERV DSPLY PD or DSPLYS PD
Transient Directory	Directory	// EXEC DSERV DSPLY TD or DSPLYS TD
System Directory	Directory	// EXEC DSERV
Directories	All	// EXEC DSERV DSPLY ALL or DSPLYS ALL
Punch Unit	Element	Control Statements
Core Image Library	Phase	// EXEC CSERV PUNCH phase 1[,phase2, . . .]
	Program	// EXEC CSERV PUNCH prog1.ALL[,prog2.ALL, ...]
	Library	// EXEC CSERV PUNCH ALL
Relocatable	Module	// EXEC RSERV PUNCH module 1[,module2, ...]
	Program	// EXEC RSERV PUNCH prog1.ALL[.prog2.ALL, ...]
	Library	// EXEC RSERV PUNCH ALL
Source Statement Library	Book	// EXEC SSERV PUNCH sublib.book1[,sublib.book2, . . .][,CMPRSD]
	Sublibrary	// EXEC SSERV PUNCH sublib1.ALL[,sublib2.ALL, ...][,CMPRSD]
	Library	// EXEC SSERV PUNCH ALL[,CMPRSD]
	Macro Sublibrary	// EXEC ESERV GENEND PUNCH E.book 1 (,E.book 2, . . .)
Procedure Library	Procedure	// EXEC PSERV PUNCH procedurename1[,procedurename2, ...]
	Library	// EXEC PSERV PUNCH ALL

Display and Punch Unit	Element	Control Statements
Core Image Library	Phase	// EXEC CSERV DSPCH phase1[,phase2, ...]
	Program	// EXEC CSERV DSPCH prog1.ALL[,prog2.ALL, ...]
	Library	// EXEC CSERV DSPCH ALL
Relocatable Library	Module	// EXEC RSERV DSPCH module1[,module2, ...]
	Program	// EXEC RSERV DSPCH prog1.ALL[,prog2.ALL, ...]
	Library	// EXEC RSERV DSPCH ALL
Source Statement Library	Book	// EXEC SSERV DSPCH sublib.book1[,sublib.book2, ...][,CMPRSDI,HEX| ,CMPHEX]
	Sublibrary	// EXEC SSERV DSPCH sublib.1.ALL[,sublib2.ALL, ...][,CMPRSDI,HEX| ,CMPHEX]
	Library	// EXEC SSERV DSPCH ALL[,CMPRSDI,HEX|,CMPHEX]
	Macro Sublibrary	// EXEC ESERV GENEND DSPCH e.book 1 (,E.book 2, . . .)
Procedure Library	Procedure	// EXEC PSERV DSPCH procedurename 1[,procedurename2, ...]
	Library	// EXEC PSERV DSPCH ALL

Note: //JOB,/* and /& must be included where needed

Copy Functions

Copy Unit	Element	Control Statements

Core Image Library	Phase	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks)
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYC phase1[,phase2, ...]

	Program	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002,balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks), RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPY prog1.ALL[,prog2.ALL, ...]

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date,code
		// EXTENT SYS002,balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYC ALL

Copy Unit	Element	Control Statements

Core Image Library (Cont'd)	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPY NEW

Relocatable Library	Module	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYR module1[,module2, ...]

	Program	// ASSGN SYS002,cuu
		// DLBL KJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYR prog1.ALL[,prog2.ALL, ...]

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYR ALL

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYR NEW

Source Statement Library	Book	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYS sublib.book1[,sublib.book2, ...]

	Sublibrary	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYS sublib1.ALL[,sublib2.ALL, ...]

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS, DOS SYSTEM RESIDENCE FILE ,
		date, code
		// EXTENT SYS002, balance of extent information

LIBRARIAN (. . . . Cont'd)

Copy Unit	Element	Control Statements
Source Statement Library	Library (Cont'd)	// EXEC CORGZ ALLOC CL=cylin(tracks),RL=cylin(tracks), SL=cylin(tracks),PL=cylin(tracks) * ALLOC STATEMENT MUST ALWAYS BE PRESENT COPYS ALL

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYS NEW

Procedure Library	Procedure	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYP procedurename1[,procedurename2, ...]

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks)
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYP ALL

	Library	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPYP NEW

Libraries	All	// ASSGN SYS002,cuu
		// DLBL IJSYSRS,'DOS SYSTEM RESIDENCE FILE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		ALLOC CL=cylin(tracks),RL=cylin(tracks),
		SL=cylin(tracks),PL=cylin(tracks)
		* ALLOC STATEMENT MUST ALWAYS BE PRESENT
		COPY ALL

Definition of a Private Library (See note 2)	Core Image	// ASSGN SYS003,cuu
		// DLBL IJSYSPC,'user identification of private library',
		date, code
		// EXTENT SYS003, balance of extent information
		// EXEC CORGZ
		NEWVOL CL=cylin(tracks)

	Relocatable	// ASSGN SYSRLB,cuu
		// DLBL IJSYSRL,'user identification of private library',
		date, code
		// EXTENT SYSRLB, balance of extent information
		// EXEC CORGZ
		NEWVOL RL=cylin(tracks)

	Source Statement	// ASSGN SYSSLB,cuu
		// DLBL IJSYSSL,'user identification of private library',
		date, code
		// EXTENT SYSSLB, balance of extent information
		// EXEC CORGZ
		NEWVOL SL=cylin(tracks)

LIBRARIAN (. . . . Cont'd)

Copy Unit	Element	Control Statement

Definition and Creation of a Private Library (See note 2)	Core Image	// ASSGN SYS003,cuu
		// DLBL IJSYSPC,'user identification of private library',
		date, code
		// EXTENT SYS003, balance of extent information
		// EXEC CORGZ
		NEWVOL CL=cylin(tracks)
		COPYC operands

	Relocatable	// ASSGN SYSRLB,cuu
		// DLBL IJSYSRL,'user identification of private library',
		date, code
		// EXTENT SYSRLB,balance of extent information
		// EXEC CORGZ
		NEWVOL RL=cylin(tracks)
		COPYR operands

	Source Statement	// ASSGN SYSSLB,cuu
		// DLBL IJSYSSL,'user identification of private library',
		date, code
		// EXTENT SYSSLB, balance of extent information
		// EXEC CORGZ
		NEWVOL SL=cylin(tracks)
		COPYS operands

Merge System Residence to New System Residence		// ASSGN (statements as required)
		// DLBL IJSYSRS,'NEW SYSTEM RESIDENCE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		MERGE RES,NRS
		COPY statements (COPYC,COPYR,COPYS,COPYP,COPYI)
		as required

Merge New System Residence to System Residence		// ASSGN (statements as required)
		// DLBL IJSYSRS,'NEW SYSTEM RESIDENCE',
		date, code
		// EXTENT SYS002, balance of extent information
		// EXEC CORGZ
		MERGE NRS, RES
		COPY statements (COPYI,COPYC,COPYR,COPYS,COPYP,
		COPY) as required

Merge System Residence to Private Libraries		// ASSGN (statements as required)
		// DLBL IJSYSRL,'PRIVATE RELOCATABLE LIBRARY',
		date, code
		// EXTENT SYSRLB, balance of extent information
		// DLBL IJSYSSL,'PRIVATE SOURCE STATEMENT
		LIBRARY', date, code
		// EXTENT SYSSLB, balance of extent information
		// DLBL IJSYSCL,'PRIVATE CORE IMAGE LIBRARY',
		date, code
		// EXTENT SYSCLB, balance of extent information
		ASSGN SYSCLB, cuu
		// EXEC CORGZ
		MERGE RES, PRV
		COPY statements (COPYI,COPYR,COPYS,COPYC)
		as required

Merge New System Residence to Private Libraries		// ASSGN (statements as required)
		// DLBL IJSYSRS,'NEW SYSTEM RESIDENCE', date, code
		// EXTENT SYS002, balance of extent information
		// DLBL IJSYSRL,'PRIVATE RELOCATABLE LIBRARY',
		date, code
		// EXTENT SYSRLB, balance of extent information
		// DLBL IJSYSSL,'PRIVATE SOURCE STATEMENT
		LIBRARY', date, code
		// EXTENT SYSSLB, balance of extent information
		// DLBL IJSYSCL,'PRIVATE CORE IMAGE LIBRARY',
		date, code
		// EXTENT SYSCLB, balance of extent information
		ASSGN SYSCLB,cuu
		// EXEC CORGZ
		MERGE NRS,PRV
		COPY statements (COPYR,COPYS,COPYC) as required

Copy Unit	Element	Control Statements

Merge Private Libraries to System Residence		// ASSGN (statements as required)
		// DLBL IJSYSPR,'PRIVATE RELOCATABLE LIBRARY', date, code
		// EXTENT SYS001, balance of extent information
		// DLBL IJSYSPS,'PRIVATE SOURCE STATEMENT LIBRARY', date, code
		// EXTENT SYS000, balance of extent information
		// DLBL IJSYSPC,'PRIVATE CORE IMAGE LIBRARY', date, code
		// EXTENT SYS003, balance of extent information
		// EXEC CORGZ
		MERGE PRV,RES
		COPY statements (COPYR,COPYS,COPYC) as required

Merge Private Libraries to New System Residence		// ASSGN (statements as required)
		// DLBL IJSYSRS,'NEW SYSTEM RESIDENCE', date, code
		// EXTENT SYS002, balance of extent information
		// DLBL IJSYSPR,'PRIVATE RELOCATABLE LIBRARY', date, code
		// EXTENT SYS001, balance of extent information
		// DLBL IJSYSPS,'PRIVATE SOURCE STATEMENT LIBRARY', date, code
		// EXTENT SYS000, balance of extent information
		// DLBL IJSVSPC,'PRIVATE CORE IMAGE LIBRARY', date, code
		// EXTENT SYS003, balance of extent information
		// EXEC CORGZ
		MERGE PRV,NRS
		COPY statements (COPYR,COPYS,COPYC) as required

Merge Private Libraries to Private Libraries		// ASSGN (statements as required)
		// DLBL IJSYSRL,'NEW PRIVATE RELOCATABLE LIBRARY', date, code
		// EXTENT SYSRLB, balance of extent information
		// DLBL IJSYSPR,'EXISTING PRIVATE RELOCATABLE LIBRARY', date, code
		// EXTENT SYS001, balance of extent information
		// DLBL IJSYSSL,'NEW PRIVATE SOURCE STATEMENT LIBRARY', date, code
		// EXTENT SYSSLB, balance of extent information
		// DLBL IJSYSPS,'EXISTING PRIVATE SOURCE STATEMENT LIBRARY', date, code
		// EXTENT SYS000, balance of extent information
		// DLBL IJSYSCL,'NEW PRIVATE CORE IMAGE LIBRARY', date, code
		// EXTENT SYSCLB, balance of extent information
		ASSGN SYSCLB, cuu
		// DLBL IJSYSPC,'EXISTING PRIVATE CORE IMAGE LIBRARY', date, code
		// EXTENT SYS003, balance of extent information
		// EXEC CORGZ
		MERGE PRV,PRV
		COPY statements (COPYR,COPYS,COPYC) as required
		To define the private library in the same jobstep, precede MERGE with NEWVOL statement (except for COPYC statements).

Notes: 1) //JOB, /* and /& must be included where needed
2) The private library can be updated with either a MAINT or CORGZ MERGE function

Direction of Transfer for Merge Operations

Logical Unit		IJSYSRS	IJSYSRL	IJSYSPR	IJSYSSL	IJSYSPS	IJSYSCL	IJSYSPC
File Name	SYSRES	SYS002	SYSRLB	SYS001	SYSSLB	SYS000	SYSCLB	SYS003
Merge RES to NRS	from	to						
Merge NRS to RES	to	from						
Merge RES to PRV	from		to		to		to	
Merge NRS to PRV		from	to		to		to	
Merge PRV to RES	to			from		from		from
Merge PRV to NRS		to		from		from		from
Merge PRV to PRV			to	from	to	from	to	from

ESERV

Function: De-editing and/or updating of edited macro's.
(Refer to: Guide to DOS/VS Assembler (GC33-4024))

Control Statement	Meaning
// EXEC ESERV	
GENEND	Causes generation of an assembler END statement, and a /* statement. GENEND, when present, must be the first statement after the // EXEC ESERV statement.
GENCATALS	Causes generation of the appropriate CATALS statement before each macro in the stream, and a /* statement after the last macro. GENCATALS, when present, must be the first statement after the // EXEC ESERV statement. Note: If neither GENEND nor GENCATALS is used, GENCATALS is assumed.
DSPLY sublibl.mac1[,sublibl.mac2, ...]	Produces a printout of the de-edited macro on the device assigned to SYSLST.
PUNCH sublibl.mac1[,sublibl.mac2, ...]	Produces a de-edited deck on the device assigned to SYSPCH.
DSPCH sublibl.mac1[,sublibl.mac2, ...]	Produces a deck and printout of the de-edited macro on the devices assigned to SYSPCH and SYSLST respectively.
) COL start col,n	Specifies the columns containing the sequence numbers in the statement of a macro definition. startcol A decimal number within the range 73-80, which identifies the start column of the sequence number. n A decimal number within the range 1-8, specifying the number of columns used by the sequence number. COL, when present must be the first statement following DSPLY, PUNCH or DSPCH statement. If the COL statement is omitted startcol receives a default value of 73 and n a default value of 6.
) VER seqno[+rel],l	Specifies the source statement of a macro definition which contents are to be verified. seqno+rel Identifies the source statement which is to be compared with the statement following the VER statement. seqno: The sequence number of a source statement. rel: A decimal number of 1-4 digits in length. If omitted, 0 is assumed. l A decimal number within the range 1-80. If omitted, 72 is assumed. Only the first l characters are used in the comparison.
) ADD seqno[+rel]	Add statements to a source macro definition. seqno+rel Identifies the source statement after which the new statements following the ADD statement are to be inserted. (seqno: See VER statement) (rel: See VER statement)
) DEL seqno[+rel][,seqno[+rel]]	Delete statements from a macro source definition. seqno+rel,seqno+rel Identifies the first and the last source statement of the section to be deleted. If the second operand is omitted, only the source statement identified by the first operand is deleted. (seqno: See VER statement) (rel: See VER statement)

Control Statement	Meaning
) REP seqno[+rel][,seqno[+rel]]	Replace statements in a source macro definition. seqno+rel,seqno+rel Identifies the first and the last source statement of the section which is to be replaced by the statements following the REP statement. If the second operand is omitt . omitted, only the source statement identified by the first operand is replaced. (seqno: See VER statement) (rel: See VER statement)
) RST seqno[+rel]	Specifies a new sequence number serial starts. seqno+rel Identifies the source statement after which the new serial starts. (seqno: See VER statement) (rel: See VER statement)
) END	Indicates the end of an update to a macro definition. Required statement for all updating.

Sample coding for de-editing without updating a macro definition

```
// JOB NOUPDATE
// EXEC ESERV
 PUNCH E.MAC1,E.MAC2
/*
/&
```

Sample coding for de-editing and updating a macro definition

```
// JOB UPDATE
// EXEC ESERV
 GENEND
 DSPCH E.MAC1
) COL 77,3
) VER 72 + 1,5
.PP9
) ADD 72 + 1
   AIF (&PCH NE 1400) D4
) DEL 102, 103
) REP 245
JOYCE CLC 0(4,REG6),BLANKS
) END
/*
// PAUSE CHECK LIST, MOVE DECK TO READER
// OPTION EDECK, NODECK
// EXEC ASSEMBLY

        deck produced by ESERV

// PAUSE MOVE SYSPCH DECK TO READER
// EXEC MAINT

        deck produced by assembler

/*
/&
```

LSERV

The label information cylinder is on the first full cylinder after the last system library on SYSRES. A display of all labels on the cylinder, with the exception of Data Set Secured labels, can be obtained by executing LSERV. Illustrations in this section show the location of the label information cylinder on SYSRES, and the layout of label information and record format.

System requirements

LSERV may be executed in any partition, with a minimum of 8192 bytes of the real or virtual address areas. LSERV assumes that the SYSRES label cylinder is formatted as described in DOS/VSE DASD Labels.

Executing LSERV

The control statements necessary to execute LSERV in a virtual partition are:

From the console:

```
// EXEC LSERV
```

From the reader:

```
// JOB jobname
// EXEC LSERV
/*
/&
```

LSERV can also be executed in a real partition. The output of LSERV shows the contents of the label cylinder on the device assigned to SYSRES. The output is directed to the device assigned to SYSLST.

When and How to use

1. Operator action given in DOS/VSE Messages indicates when LSERV must be executed. Programmer action, also given in DOS/VSE Messages, explains how to use the LSERV printout. For example, under the message:
   OP36NO REC FND
2. It is useful to execute LSERV prior to running a program that is known to have been run sometime in the past, but whose workfile assignments and partition allocations are unknown.
3. LSERV can be used for error analysis. LSERV displays the TLBL and the DLBL and EXTENT information contained on the SYSRES label cylinder. Information about secured data files is not displayed.

Summary of information provided

The printout of LSERV will show you the following details about the previous run:

- Whether the correct DLBL/EXTENT information is still on the label cylinder
- The permanent files
- The temporary files
- Extent type
- File type

For more information,
Refer to DOS/VSE Serviceability Aids and Debugging Procedures (GC33-5380).

SUPERVISOR MACROS

Supervisor Macro Instruction

Operation	Operand	Explanation
SUPVR	ASCII = {NO / YES}	Indicates whether the supervisor is to support American National Standard Code for Information Interchange (ASCII). YES adds 512 Bytes to the supervisor size.
	EU = {NO / YES}	YES must be specified if mixed parity tape processing is required for the emulators, the tape preprocessor and the tape postprocessor programs.
	ID = {1 / c}	Specifies whether the Supervisor Select option is to be used. By specifying an alphameric character (A-Z, 1-9), a unique name will be assigned to the supervisor being generated. For example, ID=A generates a supervisor named $$A$SUPA. It this parameter is omitted the supervisor will be named $$A$SUP1.
	MICR = {NO / 1419 / 1419D}	Indicates support for magnetic ink or optical character reader/sorters. The specification 14 indicates support for 1419s with Single Address Adapter, 1255s, 1259s, or 1270s. 1419D specification gives support for 1419s with Dual Address adapter, or 1275s. Burst mode and MICR devices cannot run concurrently on the same byt multiplexor channel.
	MODE = {370 / E}	Specifies whether 370 or 4300 hardware is supported.
	NPARTS = {3 / n}	Specifies the number of partitions to be supported. The minimum is 2. The maximum value for N is 5. The default value is 3.
	PAGEIN = {NO / n}	Specifies that paging activity is to be controlled by means of the macros PAGEIN, RELPAG, and FCEPGOUT. The value n indicates the maximum no. of page-in requests that can be queued for execution. Each PAGEIN macro issued in a program represents a page-in request. To obtain the support for the 3 macros, the value n must be 1 or higher. Although the maximum number is not restricted, n should not exceed 15.
	RMS = {NO / YES} (models 115 and 125 only)	Specifies whether the entire RMS (MCAR, CCH, and RMSR) is to be generated. RMS=YES is required if the string switch feature is installed. If RMS=NO is specified and TP=VTAM, then RMS=YES is forced.
	TP = BTAM/VTAM	Specification of VTAM forces AB=YES, FCPREAL =YES, IT=YES, OC=YES, PC=YES, PFIX=YES, AP=YES, TOD=YES, WAITM=YES, and for Models 115 and 125 RMS=YES.

Describe the Hardware Features

Operation	Operand	Explanation
CONFG	FP = {NO / YES}	Floating Point Feature
	MODEL = modnumber	Specify the model number of the processor.

Specify Optional Support in the Supervisor

Operation	Operand	Explanation
FOPT	CBF={NO / n}	Specifies whether output to a console printer assigned as SYSLOG is to be buffered. n specifies the number of buffers to be generated. Accepted values are from 1 to 50. An invalid specification causes one buffer to be generated.
	DASDFP={NO / (n^1,n^2)}	Specifies support for protection of DASD files, where n^1,n^2 indicates the whole range of channels to which the devices may be attached. Specification or n^1,n^2 provides file protection for disk devices.
	DOC={NO / 125D / 3277}	Only valid in 370 mode. Specifies whether support is required for Display Operator Console (DCC). The necessary supervisor routines are generated to allow to assign SYSLOG either to a 125D or 1 3277. The default taken is dependent on the CPU model number as specified by the MODEL parameter of the CONFG macro. If MODEL=115, or 125, 125D will be used as a default. If MODEL=138, 148, 158, 3011, or 4300, 3277 will be the default. For any other MODEL the default will be N0. Note: A Display Operator Console must not be attached to the same selector or multiplexor channel as SYSRES.
	ERRQ={5 / n}	Specify the number of entries for the error queue. With multiprogramming support n may be from 5 to 50. Lhe lower value is the default value in each case.
	EVA={NO / r,w}	Specifies support for error by volume analysis for 2400 and 3400 series tapes. r is the read error threshold. w is the write error threshold.
	FASTTR={NO / YES}	Specifies whether fast CCW translation is to be supported. Specifying FASTTR=YES affects the maximum, minimum and default values of BUFSIZE=n in VSTAB.
	JA={NO / YES / (n1,n2,n3, ...)}	Generates Job Accounting Interface support. YES specifies basic support (SIOs not counted). (n1,n2,n3, . . .) specifies the number of devic es per partition for which SIOs are to be counted. The maximum value of n for any partition is 255; the default value is 0. n1 always specifies the number for the background partition. The partitions to which n2,n3 . . . refer depend on the number of partitions in the system. The table below shows the pattern for partition association to values n2 through n5. n2 n3 n4 n5 2 partitions F1 3 partitions F2 F1 4 partitions F3 F2 F1 5 partitions F4 F3 F2 F1
	JALIOCS={NC / n1,n2}	Generates a user save area and an alternative label area for job accounting. n1 specifies the number of bytes for the save area for user-written job-accounting routines. The maximum acceptable value is 1024 and the default value is 16. n2 specifies the number of bytes for the alternative label area. The maximum value is 244 and the default value is 0.
	PRTY=(partion, partition, . . .	Specifies the dispatching priorities of partitions in ascending order. The number of operands must be equal to the number of partitions in the system. The default sequence is BG, Fn, . . ., that is, BG has the lowest priority and F1 has the highest priority.

Specify Optional Support in the Supervisor (. . . . Cont'd)

Operation	Operand	Explanation
FOPT (cont'd)	PSLD= {5 / n}	Specifies the number of entries in the Private Second Level Directories. A PSLD is created for each partition. The minimum value that may be specified for n is 5. The upper limit is not restricted but the value specified should not exceed 18 for CKD devices and 30 for FBA devices. 5 is assumed in case of an incorrect specification. A performance decrease will result if the number specified is less than the number of actually used directory tracks of a Private Core Image Library.
	RPS= {NO / YES}	Provides support for the Rotational Position Sensing (RPS) capabilities of DASD devices supporting the feature.
	SLD= {5 / n}	Specifies the number of entries in the Second Level Directory. The minimum value for n is 5. This value is assumed in the case of an incorrect specification. A performance decrease will result if the number specified is less than the number of actually used directory tracks of the System Core Image Library. An upper limit for CKD is 18, for FBA it is 30, although the number is not restricted.
	SYNCH= {NO / YES}	Specifies whether the synchronous exit function is to be supported. SYNCH=YES should be specified if problem programs use the synchronous exit facility (SVC screening) to support multiple users in one partition.
	SYSFIL= {NO / YES / (YES,n1,n2)}	Specifies if system input and system output files (SYSRDR, SYSIPT, SYSLST, SYSPCH) in any partition may be assigned to a disk device or diskette, or if extended support for the procedure library is desired. Specification of YES gives support for all disk devices including diskette. n1 indicates the residual capacity (in records) for beginning of operator notification when SYSLST is assigned to a disk device. n2 indicates the same for SYSPCH. Acceptable values for n1 and n2 are from 100 to 65535; if either is omitted, 1000 is assumed. For diskette this notification is not supported.
	TEBV= {IR / CR}	Specifies whether individual (IR) or combined (CR) recording of error statistics is required for unlabeled or nonstandard 2400 or 3400 series tapes.
	TRKHLD= {NO / n}	Specifies whether the Track-Hold feature is to be supported for DASD in a supervisor that supports multiprogramming. n indicates the maximum no. of tracks/blocks to be held at any one time. Accepted values are from 1 to 255. A Track-Hold Table (THTAB) with n entries is generated in the supervisor. An invalid specification results in 10 entries being generated. The NPARTS parameter in the SUPVR macro must specify more than one partition if TRKHLD=n is specified.
	TTIME= {NO / partition ID}	Specifies whether the timer is to be supported and if so, the partition owning the task timer. Only one partition ID can be specified. The partition ID is BG or Fn (where n is one of the foreground partitions generated into the system). If the partition ID sequence does not correspond to the appropriate NPARTS specification, or if the value specified for NPARTS is invalid, an MNOTE is issued.

Specify Optional Support in the Supervisor (. . . . Cont'd)

Operation	Operand	Explanation
FOPT (Cont'd)	USERID=id	Specifies whether a supervisor id is to be printed as part of the IPL COMPLETE message. id may be up to 16 bytes long. If you specify more than 16 bytes, the id will be truncated on the right. If you specify less than 16 bytes, the id will be padded with blanks on the right. The specified id is placed in a 16 byte area that immediately precedes the BG communication region. The field contains blanks if the parameter is not specified.
	XECB={NO / YES / n}	Specifies whether Cross Partition Event Control is to be supported. If YES is specified, four XECB's are generated per partition (as specified in NPARTS) n must be specified as a numeric value and indicates the number of XECB's for which an entry is to be generated in a supervisor internal table.
	ZONE={NO / EAST / WEST ,hh,mm}	Indicates the difference between Greenwich Mean Time and local time for obtaining the local time of day. NC is assumed if the parameter is omitted cr is invalid, or if TOD=NO is specified.

Define options and Configuration requirements to be included in Physical IOCS

Operation	Operand	Explanation
PIOCS Describes the System I/O Configu-tion		
	CHANSW={NO / YES}	Specifies whether channel switching is to be supported for tape or disk control units on selector cr block multiplexor channels. If CHANSW=YES and TAPE=NO is specified, TAPE=YES is forced.
	DISK=(device-list)	Specifies the type of disk devices to be supported. Specification of DISK=xxxx gives support to all attachable DASD devices of the type specified. If support for more than one type of DASD device is required, each type must be specified. For example, to provide support to 3330s and 3340s, specify DISK=(3330, 3340). For Model 115, 3340 is forced. For Model 125, 3330 is forced if neither 3330 nor 3340 is specified. If MODEL specifies 4300-hardware and MODE=370, 3340 is default.
VSTAB Describes Real and Virtual Storage and the Channel Program Trans-lation Copy Blocks		
	VSIZE=nK	Only valid in 370 mode. Specifies the size of virtual storage including SVA. n must be at least 128K times the number of partitions specified by the NPARTS parameter and a multiple of 2, plus the size of the SVA. The minimum size for the SVA, for system purposes, is 100K (including the system GETVIS area of 20K). This size and the amount of the SVA space to be used for user modules have to be added to the size for the partitions.
	BUFSIZE=n	Specifies the number of copy blocks to be generated for the Channel Program Translation routine. The default and minimum values are shown in the DOS/VSE SYSGEN manual.

Define options and Configuration requirements to be included in Physical IOCS (. . . . Cont'd)

Operation	Operand	Explanation
IOTAB	BGPGR={10 / n}	Specifies the number of programmer logical units (SYSnnn) for the BG partition. The minimum value for n is 10, and the maximum is 241. A partition LUB table is generated with a two-byte entry for each system logical unit and additional entries for the number of units specified by n.
	BSCLNS={1 / n}	Specifies the maximum number of BSC lines for Model 115 or 125 with Integrated Communications Adapter (ICA). The minimum specification is 0 and the maximum is 6. The specified value is entered in the second byte of the Line Mode Table (MODTAB) and the table is generated with this number of entries for BSC line mode settings. If the parameter is omitted, n defaults to 1.
	CHANQ={10 / 8 / 6 / n}	Specifies the number of 8-byte entries to be generated for the channel queue. At least six entries are always generated, but if more than three partitions are specified by the NPARTS parameter of the SUPVR macro, then this minimum is increased by two for each additional partition and is further increased by the number of console buffers specified by the CBF parameter in the FOPT macro.
	D2311={0 / n} D2314={0 / 2 / n} D2400={0 / n} D3330={0 / 2 / n} D3340={0 / 2 / n} D3350={0 / 2 / n} D3410={0 / 2 / n} D3420={0 / n} D3540={0 / n} D3800={0 / n} D3886={0 / n} D3895=n D8809=n DFBA=n	Specifies the number of each type of special device attached to the system. The number specified for each device may be more than the actual number attached, but if the total number of special devices specified exceeds the number of devices specified in the IODEV parameter the latter number is set to the total number of special devices plus 5. Dxxxx can have two functions: 1. For each special device specified, a PUB table extension (PUB2) entry is generated of appropriate length for that device. 2. For device on a block multiplexer channel running in block multiplex mode additional CCWs are generated, for DASDFP. For defaults and values to be specified refer to DOS/VSE System Generation Manual. Only valid in ECPS:VSE mode. Specifies the number of FBA devices attached to the system.

Define options and Configuration requirements to be included in Physical IOCS (. . . Cont'd)

Operation	Operand	Explanation
	FnPGR={10/m}	Specifies for foreground partition n the number of programmer logical units. The minimum value for m is 10, and the maximum value is 241.
	IODEV={10/n}	Specifies the number of I/O devices attached to the system. The maximum specification allowed is 254, the minimum is 4. Each unit requiring an ADD entry must be included in n.
	JIB={5xnparts/n}	Specifies the number of Job Information Blocks (JIBs) for the system (the minimum is 5, the maximum is 255). Requirements are: 1. One JIB for each temporary logical unit assignment. 2. One JIB for each alternate logical unit assignment.
	NRES={d/n}	Indicates the number of Resource Usage Records (RURs). The default value d is equal to the no. of partitions supported plus 10. The minimum value accepted is the no. of partitions increased by 2.
	SSLNS={4/n}	Specifies the maximum number of Start/Stop lines for the Model 115 or 125 with the Integrated Communications Adapter (ICA). The minimum specification is 0 and the maximum is 16. The specified value is entered in the first byte of the Line Mode Table (MODTAB) and the table is generated with this number of entries for Start/Stop line mode settings. It the parameter is omitted and MODEL=115 or 125, then n defaults to 4.

DEVICE TYPE CODES

Card Code	Actual IBM Device	Device-Type X'nn'	Device Type
2400T9	9-track Magnetic Tape units	50	Magnetic Tape devices
2400T7	7-track Magnetic Tape units	50	
3410T9	9-track 3410 Magnetic Tape units	53	
3410T7	7-track 3410 Magnetic Tape units	53	
3420T9	9-track 3420 Magnetic Tape units	52	
3420T7	7-track 3420 Magnetic Tape units	52	
8809	8809 Magnetic Tape unit	5A	
1442N1	1442N1 Card Read Punch	30	Card Read Punches
2520B1	2520B1 Card Read Punch	31	
2560	2560 Multifunction Card machine	33	
2596	2596 Card Read Punch	30	
3525RP	3525 Card Punch (witch optional read feature)	32	
5425	5424/5425 Multifunction Card Unit	34	
2501	2501 Card Reader	10	Card Readers
2540R	2540 Card Reader	11	
3504	3504 Card Reader	12	
3505	3505 Card Reader	12	
2540P	2540 Card Punch	21	Card Punches
2520B2	2520B2 Card Punch	20	
1442N2	1442N2 Card Punch	22	
2520B3	2520B3 Card Punch	20	
3525P	3525 Card Punch	23	
PRT 1	3211, 3203-4, 3203-5 and 3289-4 Printer	43	Printers
1403	1403 Printer	40	
1403U	1403 Printer with UCS feature	42	
1443	1443 Printer	41	
2245	2245 KANJI Printer (Supported in Real Mode only)	44	
3203	3203 Printer	4A	
3211	SAME AS PRT 1	43	
3277 (local 3270)	3284, 3286, 3287, 3288 Printers with 3277 Control Unit or 3284, 3286, 3287, 3288, and 3289 Printers with 3274 Control Unit. MODE operand must be entered as X'01'.	B0	
3277B (local 3270)	3284, 3286, 3287, 3288 Printers with 3277 Control Unit or 3284, 3286, 3287, 3288, and 3289 Printers with 3274 Control Unit, attached in burstmode to a multiplexor chan. (MODE operand must be entered as X'01').	B0	
3800	3800 Printing Subsystem	45	
3800B	3800 Printing Subsystem with Burster Trimmer-Stacker (BTS)	45	
3800BC	3800 Printing Subsystem (BTS and Additional CGS	45	
3800C	3800 Printing Subsystem with Additional Character Generation Storage (CGS)	45	
5203	5203 Printer	4C	
5203U	5203 Printer with UCS feature	4D	
1050A	3210, 3215 Console Printer Keyboards	00	Printer Keyboards
	3286-2 in Printer Keyboard Mode	00	
125D	Model 115 or Model 125 Integrated Display Operation Console	B2	Display Operator Console
	Model 138/148 Console in 115/125 Console Display Emulation Mode		
125DP	Model 115 or Model 125 Integrated Display Operator Console	B2	
	Model 138/148 Console in 115/125 Console with 5213 Console Printer attached		
3277	3277 Display Operator Console	B0	
3277	3284, 3286 or 3287 Console Printer (the MODE operand must be entered as X'02')	B0	

DEVICE TYPE CODES (. . . . Cont'd)

Card Code	Actual IBM Device	Device-Type X nn	Device Type
UNSP	Unsupported Device	FF	Unsupported Device
UNSPB	Unsupported Device	FF	
FBA	3310/3370 Direct Access Storage Device	90	DASD
2311	2311 Disk Storage Device	60	
2314	2314 Direct Access Storage Facility	62	
2314	2319 Disk Storage Facility	62	
3330	3330 Disk Storage, Model 1 and 2, or 3333-1	63	
3330B	3330 Disk Storage Modell 11	65	
3340	3340 Disk Storage (General)	68	
3340R	3340 Disk Storage with RPS Feature (with or without 3340 Data Module Model 35)	68	
3340R	3340 Disk Storage with RPS Feature (with or without 3340 Data Module Model 70)	68	
3340	3344 Direct Access Storage	68	
3340R	3344 Direct Access Storage with RPS	68	
3340	3340 Disk Storage without RPS Feature (with or without 3340 Data Module Model 35)	69	
3340	3340 Disk Storage without RPS Feature (with or without 3340 Data Module Model 70)	6A	
3350	3350 Disk Drive	67	
1419	1255 Magnetic Character Reader	72	MICR-Magnetic Ink Character Recognition Devices
1419	1259 Magnetic Character Reader	72	
1419	1419 Magnetic Character Reader	72	
1419P	1419 Dual Address Adapter Primary Control Unit	73	
1419S	1419 Dual Address Adapter Secondary Control Unit	74	
3895	3895 Document Reader/Inscriber	7D	Reader/Inscriber
2701	2701/2715 Data Adapter unit	D0	Teleprocessing lines
2701	Integrated Communications Adapter (Model 135)	D2	
2702	2702 Transmission Control unit	D1	
2703	2703 Transmission Control unit	D2	
2703	Integrated Communication Adapter (Model 115/125)	D2	
2703	3704/3705 Communication Controller in Emulation Mode	D2	
3704	3704 Communication Controller	DC	
3705	3705 Communication Controller	DC	
3791L 3274-1A	3791 Local Communication Controller	DE	
3705	SDLC ICA on 4300 Processors (Mode must be entered AS10)		
LCA2	3791 Cluster Controller, Locally Attached	DE	
1419	1270 Optical Reader/Sorter	72	Optical Readers
1419P	1275 Optical Reader Sorter Primary Control Unit	73	
1419S	1275 Optical Reader Sorter Secondary Control Unit	74	
1287	1287 Optical Reader	77	
1288	1288 Optical Page Reader	77	
3881	3881 Optical Mark Reader	11	
3886	3886 Optical Character Reader	7C	
3540	3540 Diskette Input/Output Unit	80	Diskette
7443	7443 System Recording File	88	
2260	2260 Display Station	C0	Display Stations
3277 (local 3270)	3277 Display Station; MODE operand need not be entered	B0	
3277B (local 3270)	3277 Display Station; attached in burst mode to a multiplexor channel. MODE operand need not be entered	B0	

DEVICE TYPE CODES (. . . . Cont'd)

Card Code	Actual IBM Device	Device-Type X nn	Device Type
7770	7770 Audio Response Unit	D3	Audio Response Units
7772	7772 Audio Response Unit	D4	
1017	1017 Paper Tape Reader with 2826 Control Unit M.1	78	Paper Tape Readers
1017TP	1017 Paper Tape Reader with 2826 Control Unit Model 2	D5	
2671	2671 Paper Tape Reader	70	
1018	1018 Paper Tape Punch with 2826 Control Unit M.1	79	Paper Tape Punch
1018TP	1018 Paper Tape Punch with 2826 Control Unit Model 2	D6	

FORMAT OF THE ESD CARD

Card Columns	
1	Multiple punch (12-2-9). Identifies this as a loader card.
2-4	ESD - External Symbol Dictionary card.
11-12	Number of bytes of information contained in this card.
15-16	External symbol identification number (ESID) of the first SD, PC, CM or ER on this card. Relates the SD, PC, CM or ER to a particular control section.
17-72	Variable information. 8 positions - Name 1 position - Type code hex 00 , 01 , 02 , 04 , 05 , or 0A to indicate SD, LD, ER, PC, CM, or WX respectively. 3 positions - Assembled origin 1 position - Blank 3 positions - Lenghth, if an SD type, CM type, or a PC type. If an LD type, this field contains the external symbol identification number (ESID) of the SD containing the label.
73-80	May be used by the programmer for identification.

FORMAT OF THE TXT CARD

Card Columns	
1	Multiple punch (12-2-9). Identifies this as a loader card.
2-4	TXT - Text card.
6-8	Assembled origin (Address of first byte to be loaded from this card).
11-12	Number of bytes of text to be loaded.
15-16	External symbol identification number (ESID) of the control section (SD or PC) containing the text.
17-72	Up to 56 bytes of text — data or instructions to be loaded.
73-80	May be used for program identification.

FORMAT OF THE RLD CARD

Card Columns	
1	Multiple punch (12-2-9). Identifies this as a loader card.
2-4	RLD — Relocation List Dictionary Card.
11-12	Number of bytes of information contained in this card.
17-72	Variable information (multiple items). a) Two positions - (relocation identifier) pointer to the ESID number of the ESD item on which the relocation factor of the contents of the address constant is dependent. b) Two positions - (position identifier) pointer to the ESID number of the ESD item on which the position of the address constant is dependent. c) One position - flag indicating type of constant, as follows: Bits 0-2 Ignored 3 0 - a non branch type load constant 1 - a branch type load constant 4-5 00 - load constant length = 1 byte 01 - load constant length = 2 bytes 10 - load constant length = 3 bytes 11 - load constant length = 4 bytes 6 0 - relocation factor is to be added 1 - relocation factor is to be subtracted 7 0 - Next load constant has different R and P identifiers; therefore, both R and P must be present. 1 - Next load constant has the same R and P identifiers; therefore they are both omitted. Five significant bits of this byte are expanded in the RSERV printout. d) Three positions — assembled origin of load constant.
73-80	May be used for program identification.

FORMAT OF THE END CARD

Card Columns	
1	Multiple punch (12-2-9). Identifies this as a loader card.
2-4	END
6-8	Assembled origin of the label supplied to the Assembler in the END card (optional).
15-16	ESID number of the control section to which this END card refers (only if 6-8 present).
17-22	Symbolic label supplied to the Assembler if this label was not defined within the assembly.
29-32	Control section length (if not specified in the last SD or PC).
73-80	Not used.

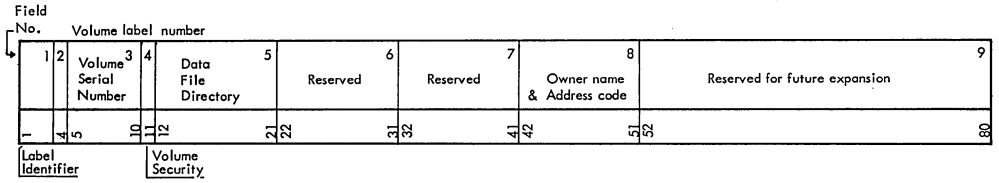
FORMAT OF THE REP (User Replace) CARD

Card Columns	
1	Multiple punch (12-2-9). Identifies this as a loader card.
2-4	REP — Replace text card.
5-6	Blank.
7-12	Assembled address of the first byte to be replaced (hexadecimal). Must be right justified with leading zero's if needed to fill the field.
13	Blank
14-16	External symbol identification number (ESID) of the control section (SD) containing the text (hexadecimal). Must be right justified with leading zero's if needed to fill the field.
17-70	From 1-11 4-digit hexadecimal fields separated by comma's, each replacing two bytes. A blank indicates the end of information in this card.
71-72	Blank
73-80	May be used for program identification.

CHAPTER III
DOS/VSE IOCS (GENERAL/SAM/DAM/ISAM)

STANDARD VOLUME LABEL, TAPE OR DASD

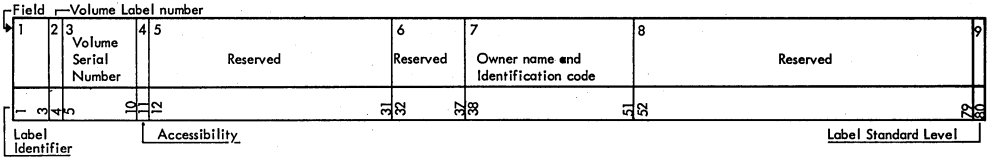
IBM Standard Volume Label Format (80 bytes) for EBCDIC Tape or DASD



FIELD	NAME AND LENGTH	DESCRIPTION
1	LABEL IDENTIFIER 3 bytes	Must contain VOL to indicate that this is a Volume Label.
2	VOLUME LABEL NR 1 byte	Indicates the relative position (1-8) of a volume label within a group of volume labels.
3	VOLUME SERIAL NR 6 bytes	A unique identification code which is assigned to a volume when it enters an installation. This code may also appear on the external surface of the volume for visual identification. It is normally a numeric field 000001 to 999999, however any or all of the 6 bytes may be alphameric.
4	VOLUME SECURITY 1 byte (OS/VS only)	Indicates security status of the volume: 0: no further identification for each file of the volume is required. 1: Further identification for each file of the volume is required before processing.
5	DATA FILE DIRECTORY 10 bytes	For DASD only. The first 5 bytes contain the starting address(CCHHR) of the VTOC. The last 5 bytes are blank. For tape files this field is not used and should be recorded as blanks.
6	RESERVED 10 bytes	Reserved
7	RESERVED 10 bytes	Reserved
8	OWNER NAME AND ADDRESS CODE 10 bytes	Indicates a specific customer, installation and/or system to which the volume belongs. This field may be a standardized code, name, address etc. (OS/VS only).
9	RESERVED 29 bytes	Reserved

Note: All reserved fields should contain blanks to facilitate their use in the future. Any information appearing in these fields at the present time will be ignored by the DOS/VS and OS/VS programs.

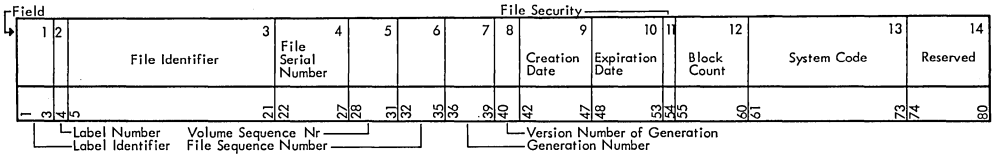
ANSI Standard Volume Label format and contents for ASCII tapes



FIELD	NAME AND LENGTH	DESCRIPTION
1	LABEL IDENTIFIER 3 bytes	Must contain VOL to indicate that this a Volume Label.
2	VOLUME LABEL NR 1 byte	Must be 1. If any other standard volume labels are present (indicated by an entry other than 1 in this field)they are ignored.
3	VOLUME SERIAL NR 6 bytes	Uniquely identifies this volume. Must consist of 6 bytes and may be any character exept a quote(').
4	ACCESSIBILITY 1 byte	Indicates accessibility protection: Space: No accessibility protection. Nonspace: Accessibility protection.
5	RESERVED 20 bytes	Reserved for future use as required by the American National Standards Institute, Inc. Should contain spaces.
6	RESERVED 6 bytes	Reserved for future use as required by American National Standards Institute, Inc. Should contain spaces.
7	OWNER NAME AND IDENTIFICATION CODE 14 bytes	Indicates a specific customer, installation and/or system to which the volume belongs. This field may be a standardized code, name, address etc.
8	RESERVED 28 bytes	Reserved for future use as required by the American National Standards Institute, Inc. Should contain spaces.
9	LABEL STANDARD LEVEL 1 byte	Indicates whether this volume observes the American National Standards: Dec. 1: Volume observes the standards (1 is also the default value) Space: Volume does not observe the standards, but it follows an agreed format.

STANDARD MAGNETIC TAPE FILE LABEL

IBM Standard Tape File Label Format and Contents

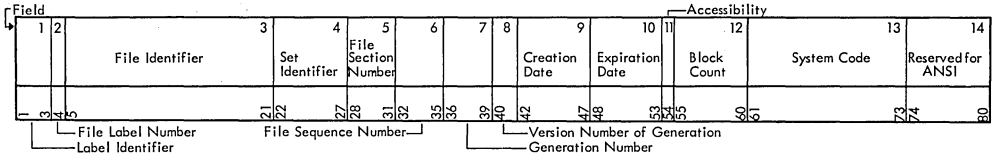


FIELD	NAME AND LENGTH	DESCRIPTION
1	LABEL IDENTIFIER 3 bytes EBCDIC	Identifies the type of label: HDR: Header--beginning of data file. EOF: End of File--end of a set of data. EOV: End of Volume--end of the physical reel.
2	FILE LABEL NUMBER 1 byte EBCDIC	Always a 1
3	FILE IDENTIFIER 17 bytes EBCDIC	Uniquely identifies the entire file, may contain only printable characters.
4	FILE SERIAL NUMBER 6 bytes EBCDIC	Uniquely identifies a file-volume relationship. This field is identical to the Volume Serial Number in the volume label on the first or only volume of a multi-volume file or a multi-file set. This field will normally be numeric(000001 to 999999) but may contain any six alphameric characters.
5	VOLUME SEQUENCE NUMBER 4 bytes	Indicates the order of a volume in a given file or multi-file set. This number must be numeric(0000-9999). Multiple volumes of an output file will be numbered in consecutive sequence.
6	FILE SEQUENCE NUMBER 4 bytes	Assign numeric sequence to a file within a multi file set.
7	GENERATION NUMBER 4 bytes	Numerically identifies the various editions of the file.
8	VERSION NUMBER OF GENERATION 2 bytes	Indicates the version of the generation of a file.

FIELD	NAME AND LENGTH	DESCRIPTION
9	CREATION DATE 6 bytes	Indicates the year and the day of the year that the file was created: Position / Code / Meaning 1 / blank / none 2-3 / 00-99 / year 4-6 / 001-366 / day of year (e.g., January 31, 1965 would be entered as 65031)
10	EXPIRATION DATE 6 bytes	Indicates the year and the day of the year when the file may become a scratch tape. The format of this field is identical to field 9. On a multi-file reel, processed sequentially, all files are considered to expire on the same day.
11	FILE SECURITY 1 byte	Indicates the security status of the file. 0: No security protection. 1: Security protection. Additional identification of the file is required before it can be processed.
12	BLOCK COUNT 6 bytes	Indicates the number of data blocks written on the file from the last header label to the first trailer label, exclusive of tape marks. Count does not include checkpoint records. This field is used in trailer labels.
13	SYSTEM CODE 13 bytes	Uniquely identifies the programming system.
14	RESERVED 7 bytes	Reserved

STANDARD MAGNETIC TAPE FILE LABEL

ANSI Standard Tape File Label Format and Contents

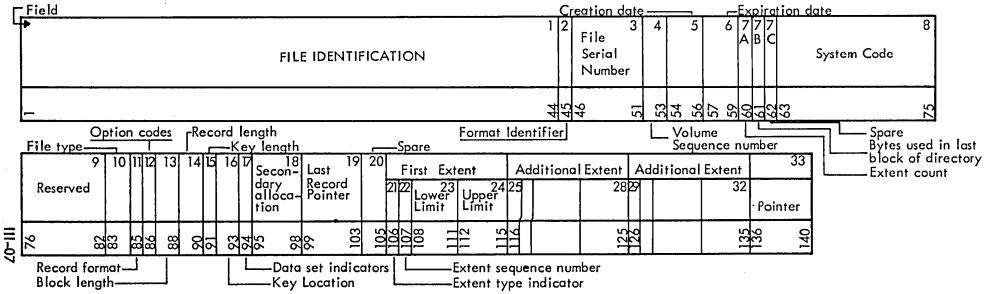


FIELD	NAME AND LENGTH	DESCRIPTION
1	LABEL IDENTIFIER 3 bytes, ASCII	Identifies the type of label: HDR: Header--beginning of a data file. EOF: End of File--end of a set of data. EOV: End of Volume--end of the physical reel.
2	FILE LABEL NUMBER 1 byte, ASCII	Indicates the sequence of this label within a label group (HDR, EOF, EOV). DOS/VS supports File Label 1 only and ignores subsequent numbers.
3	FILE IDENTIFIER 17 bytes, ASCII	Identifies the entire file. May be any character exept a quote(').
4	SET IDENTIFIER 6 bytes, ASCII	Identifies the volume-file relationship. Generally, this field is identical to the volume serial number from the VOL label or the first or only volume of the logical file.
5	FILE SECTION NUMBER 4 bytes	Indicates the order of a volume in a given file or multi-file set. (The first file must be numbered 0001).
6	FILE SEQUENCE NUMBER 4 bytes	Assigns numeric sequence to a file within a multi-file set. (The first file must be numbered 0001).
7	GENERATION NUMBER 4 bytes	Numerically identifies this edition of the file. (Must be numerical or blank).
8	VERSION NUMBER OF GENERATION 2 bytes	Indicates this version of the generation in field 7. (Must be numerical or blank).
9	CREATION DATE 6 bytes	Indicates the year and the day of the year that this file was created (byyddd), where: b= blank yy= year (00-99) ddd= day (001-366)

FIELD	NAME AND LENGTH	DESCRIPTION
10	EXPIRATION DATE 6 bytes	Indicates the year and the day of the year that this file may become a scratch tape. Same format as above (Field 9).
11	ACCESSIBILITY 1 byte	Indicates the accessibility protection of the file. Space: no accessibility protection. Nonspace: accessibility protection.
12	BLOCK COUNT 6 bytes	Indicates the number of data blocks (physical records) written for this logical file.
13	SYSTEM CODE 13 bytes	Uniquely identifies the programming system.
14	RESERVED 7 bytes	Reserved for future use as required by ANSI.(American National Standards Institute,Inc.). Should be recorded as spaces.

STANDARD DASD FILE LABEL , FORMAT 1

(Format 1: This format is common to all data files on Direct Access Storage Devices)



FIELD	NAME AND LENGTH	DESCRIPTION
1	FILE NAME 44 bytes, alphameric EBCDIC	This field serves as the key portion of the file label. Each file must have a unique file name. Duplication of file names will cause retrieval errors. The file name can consist of three sections: 1 File ID is an alphameric name assigned by the user and identifies the file. Can be 1-35 bytes if generation and version numbers are used, or 1-44 bytes if they are not used. 2 Generation number. If used, this field is separated from File ID by a period. It has the format Gnnnn, where G identifies the field as the generation number and nnnn (in decimal) identifies the generation of the file. 3 Version Number of Generation. If used, this section immediately follows the

STANDARD DASD FILE LABEL , FORMAT 1 (....Cont'd)

FIELD	NAME AND LENGTH	DESCRIPTION
1	FILENAME (Cont'd)	generation number and has the format Vnn, where V identifies the field as the version of generation number and nnn (in decimal) identifies the version of generation of the file. Note: DOS/VS compares the entire field against the filename given in the DLBL card. The generation and version numbers are treated differently by OS/VS.

The remaining fields comprise the DATA portion of the file label:

2	FORMAT IDENTIFIER 1 byte, EBCDIC numeric	1 = Format 1
3	FILE SERIAL NR 6 bytes, EBCDIC alphameric	Uniquely identifies a file/volume relationship. It is identical to the Volume Serial Number of the first or only volume of a multivolume file.
4	VOLUME SEQUENCE NR, 2 bytes, binary	Indicates the order of a volume relative to the first volume on which the data file resides.
5	CREATION DATE 3 bytes, discontinuous binary	Indicates the year and the day of the year the file was created. It is of the form YDD, where Y signifies the year(0-99) and DD the day of the year(1-366).
6	EXPIRATION DATE 3 bytes, discontinous binary	Indicates the year and the day of the year the file may be deleted. The form of this field is the same as that of field 5.
7A	EXTENT COUNT	Contains a count of the number of extents for this file on this volume. If user labels are used, the count does not include the user label track. This field is maintained by the DOS/VS programs.
7B	BYTES USED IN LAST BLOCK OF DIRECTORY 1 byte, binary	Used by OS/VS
7C	SPARE 1 byte	Reserved
8	SYSTEM CODE 13 bytes	Uniquely identifies the programming system. The character codes that can be used in this field are limited to EBCDIC characters. On input, IOCS ignores this field. On output, IOCS writes the information supplied in DLBL.
9	RESERVED 7 bytes	Reserved
10	FILE TYPE 2 bytes	The contents of this field uniquely identify the type of data file: Hex 4000: Consecutive organization Hex 2000: Direct access organization Hex 8000: Indexed sequential organization Hex 0200: Library organization Hex 0000: Organization not defined in the file label Hex 0008: VSAM

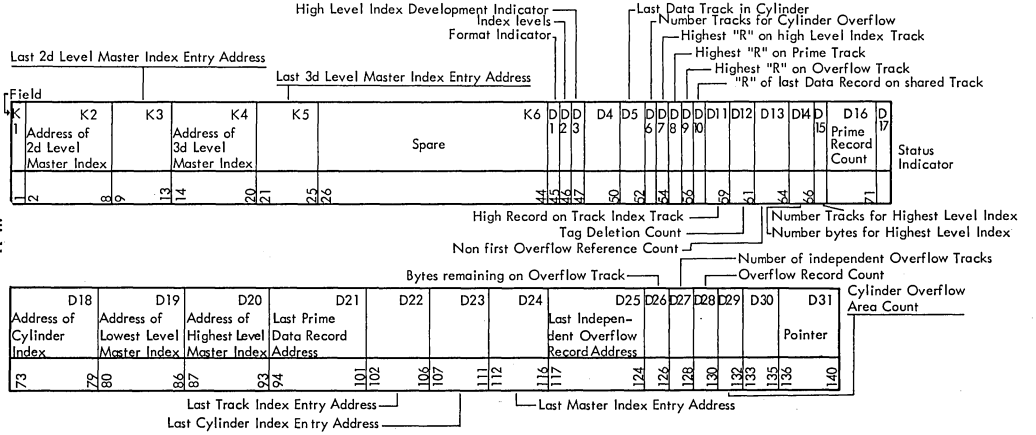
STANDARD DASD FILE LABEL , FORMAT 1 (....Cont'd)

FIELD	NAME AND LENGTH	DESCRIPTION
11	RECORD FORMAT 1 byte	Used by OS/VS
12	OPTION CODES	Bits within this field indicate various options used in building the file: Bit 0: 0 Bit 1: Reserved Bit 2: Master index present (ISAM) Bit 3: Independent overflow present (ISAM) Bit 4: Cylinder overflow present (ISAM) Bit 5: Reserved Bit 6: Delete record (OS/VS) Bit 7: Reorganize (OS/VS)
13	BLOCK LENGTH 2 bytes, binary	Indicates the block length for fixed length records or maximum block size for variable length blocks.
14	RECORD LENGTH 2 bytes, binary	Indicates the record length for fixed length records or the maximum record length for variable length records.
15	KEY LENGTH 1 byte, binary	Indicates the length of the key portion of of the data records in the file.
16	KEY LOCATION 2 bytes, binary	Indicates the high order position of the data records
17	DATA SET INDICATORS 1 byte	Bits within this field are used to indicate the following: Bit 0: If on, indicates that this is the last volume on which this file normally resides. Bit 1,2,4,6,7: 0 for DOS/VS-Used by OS/VS. Bit 3: If on, data set security is invoked. Bit 5: Used by DOS/VS and OS/VS.
18	SECONDARY ALLOCATION 4 bytes, binary	Used by OS/VS
19	LAST RECORD POINTER 5 bytes, discontinuous binary	Used by OS/VS
20	SPARE 2 bytes	Reserved
21	EXTENT TYPE INDICATOR 1 byte	Indicates the type of extent with which the following fields are associated: HEX CODE 00: Next three fields do not indicate any extent. 01: Data area (SAM, DAM), Prime data area (ISAM), Data Space (VSAM). 02: Overflow area of an indexed sequential file. 04: Cylinder index or master index area of an indexed sequential file. 40: User label track area. 80: Shared cylinder indicator.

STANDARD DASD FILE LABEL , FORMAT 1 (....Cont'd)

FIELD	NAME AND LENGTH	DESCRIPTION
22	EXTENT SEQUENCE NR 1 byte, binary	Indicates the extent sequence in a multi-extent file.
23	LOWER LIMIT 4 bytes, discontinuous binary	The cylinder and the track address specifying the starting point(lower limit) of this extent component. This field has the format CCHH.
25-28	ADDITIONAL EXTENT 10 bytes	These fields have the same format as the fields 21-24 above.
29-32	ADDITIONAL EXTENT 10 bytes	These fields have the same format as the fields 21-24 above.
33	POINTER TO NEXT FILE LABEL WITHIN THIS LABEL SET 5 bytes, discontinuous binary	The address (format CCHHR) of a continuation label if needed to further describe the file. If field 10 indicates Indexed Sequential organization, this field points to a Format 2 file label within this label set. Otherwise, it points to a Format 3 file label, and then only if the file contains more than three extent segments. This field contains all binary zeros if no additional file label is pointed to.
24	UPPER LIMIT 4 bytes	The cylinder and the track address specifying the ending point (upper limit) of this extent component. This field has the format cchh.

STANDARD DASD FILE LABEL , FORMAT 2



SEE NEXT PAGE FOR FURTHER EXPLANATION

STANDARD DASD FILE LABEL , FORMAT 2 (....Cont'd)

FIELD	NAME AND LENGTH	DESCRIPTION
K1	KEY IDENTIFICATION 1 byte	This byte contains the hex code 02 in order to avoid conflict with a file name.
K2	ADDRESS OF 2d LEVEL MASTER INDEX 7 bytes, discontinuous binary	This field contains the address of the first track of the second level of the master index, in the form MBBCCHH. (OS/VS only)
K3	LAST 2d LEVEL MASTER INDEX ENTRY 5 bytes, discontinuous binary	This field contains the address of the last index entry in the second level of the master index, of the form CCHHR. (OS/VS only)
K4	ADDRESS OF 3d LEVEL MASTER INDEX 7 bytes, discontinuous binary	This field contains the address of the first track of the third level of the master index, in the form MBBCCHH.
K5	LAST 3d LEVEL MASTER INDEX ENTRY 5 bytes, discontinuous binary	This field contains the address of the last entry in the third level of the master index, in the form CCHHR. (OS/VS only)
K6	SPARE 19 bytes	Reserved
D1	FORMAT IDENTIFIER 1 byte, EBCDIC numeric	2: Format 2
D2	NUMBER OF INDEX LEVELS 1 byte, binary	The contents of this field indicate how many levels of index are present with an Indexed Sequential File.
D3	HIGH LEVEL INDEX DEVELOPMENT INDICATOR 1 byte, binary	This field contains the number of tracks determining development of Master Index. (OS/VS only)
D4	FIRST DATA RECORD IN CYLINDER 3 bytes	This field contains the address of the last data track on each cylinder in the form HHR.
D5	LAST DATA TRACK IN CYLINDERS 2 bytes	This field contains the address of the last data track on each cylinder, in the form HH.
D6	NUMBER OF TRACKS FOR CYLINDER OVERFLOW 1 byte, binary	This field contains the number of tracks in cylinder overflow area. (OS/VS only)
D7	HIGHEST "R" ON HIGH LEVEL INDEX TRACK 1 byte	This field contains the highest possible R on track containing high-level index entries.
D8	HIGHEST "R" ON PRIME TRACK 1 byte	This field contains the highest possible R on prime data tracks for form F records.
D9	HIGHEST "R" ON OVERFLOW TRACK 1 byte	This field contains the highest possible R on overflow data tracks for form F records.

STANDARD DASD FILE LABEL , FORMAT 2 (....Cont'd)

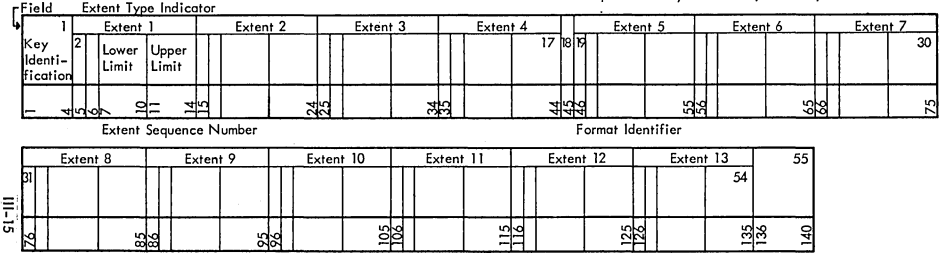
FIELD	NAME AND LENGTH	DESCRIPTION
D10	"R" OF LAST DATA RECORD ON SHARED TRACK -1 byte	This field contains the R of the last data record on a shared track.
D11	HIGH RECORD ON TRACK INDEX TRACK 2 bytes	The first byte of this 2-byte field indicates the high(0-256) record on the track index track. The second byte is reserved.
D12	TAG DELETION COUNT 2 bytes, binary	This field contains the number of records that have been tagged for deletion.
D13	NONFIRST OVERFLOW REFERENCE COUNT 3 bytes, binary	This field contains a count of the number of random references to a nonfirst overflow record.
D14	NUMBER OF BYTES FOR HIGHEST LEVEL INDEX-2 bytes binary	The contents of this field indicate how many bytes are needed to hold the highest level index in main storage.
D15	NUMBER OF TRACKS FOR HIGHEST LEVEL INDEX 1 byte, binary	This field contains a count of the number of tracks occupied by the highest level index.
D16	PRIME RECORD COUNT 4 bytes, binary	This field contains a count of the number of records in the prime data area.
D17	STATUS INDICATOR 1 byte	The eight bits of this byte are used for the following indications: Bit 0-1 : must remain off Bit 2 : file closed for ADD or ADDRTR Bit 3-5 : must remain off Bit 6 : last block full-Bit 7: last track full
D18	ADDRESS OF CYLINDER INDEX 7 bytes	This field contains the address of first track of the cylinder index, in the form MBBCCHH.
D19	ADDRESS OF LOWEST LEVEL MASTER INDEX 7 bytes	This field contains the address of the first track of the lowest-level index of the high level indexes, in the form MBBCCHH.
D20	ADDRESS OF HIGHEST LEVEL INDEX 7 bytes	This field contains the address of the first track of the highest level master index, in the form MBBCCHH.
D21	LAST PRIME DATA RECORD ADDRESS 8 bytes	This field contains the address of the last data record in the prime data area, in the form MBBCCHHR.
D22	LAST TRACK INDEX ENTRY ADDRESS 5 bytes	This field contains the address of the last normal entry in the track index on the last cylinder in the form CCHHR
D23	LAST CYLINDER INDEX ENTRY ADDRESS 5 bytes	This field contains the address of the last index entry in the cylinder index in the form CCHHR.
D24	LAST MASTER INDEX ENTRY ADDRESS 5 bytes	This field contains the address of the last index entry in the master index, in the form CCHHR.
D25	LAST INDEPENDENT OVERFLOW RECORD ADDRESS 8 bytes	This field contains the address of the last record written in the current independent overflow area, in the form MBBCCHHR.

STANDARD DASD FILE LABEL , FORMAT 2 (....Cont'd)

FIELD	NAME AND LENGTH	DESCRIPTION
D26	BYTES REMAINING ON OVERFLOW TRACK 2 bytes,binary	This field contains the number of bytes remaining on current independent overflow track.(OS/VS only)
D27	NUMBER OF INDEPEN-OVERFLOW TRACKS 2 bytes,binary	This field contains the number of tracks remaining in independent overflow area.
D28	OVERFLOW RECORD COUNT 2 bytes,binary	This field contains a count of the number of records in the overflow area.
D29	CYLINDER OVERFLOW AREA COUNT 2 bytes,binary	This field contains the number of cylinder overflow areas full.
D30	DUMMY TRACK INDEX ENTRY 3 bytes	This field contains the HHR portion of the dummy track index entry. (OS/VS only)
D31	POINTER TO FORMAT 3 FILE LABEL 5 bytes	This field contains the address(in the form CCHHR) of a Format 3 file label if more than 3 extent segments exist for the data file within this volume. Otherwise it contains binary zeros. (OS/VS only)

STANDARD DASD FILE LABEL , FORMAT 3

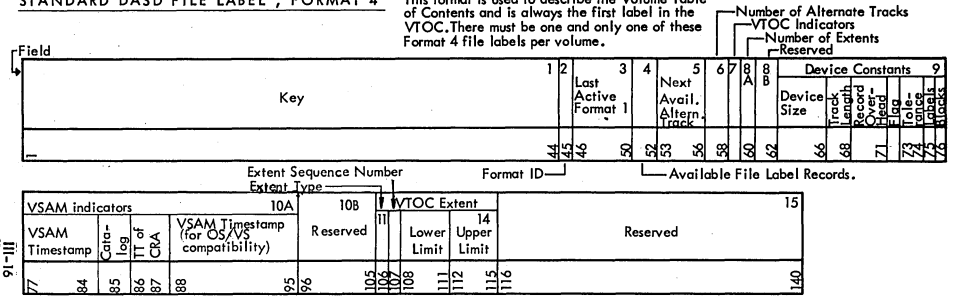
Format 3: This format is used to describe extra extent segments on the volume if there are more than can be described in the Format 1 (and Format 2 if it exists) file label. This file label is pointed to by a Format 1, Format 2, or another Format 3 file label.



FIELD	NAME AND LENGTH	DESCRIPTION
1	KEY IDENTIFICATION 4 bytes	Each byte of this field contains the Hex code 03 in order to avoid conflict with a data file name.
2-17	EXTENTS (in KEY) 40 bytes	Four groups of fields identical in format to fields 21-24 in the Format 1 label are contained here.
18	FORMAT IDENTIFIER 1 byte, EBCDIC numeric	3 : Format 3
19-54	ADDITIONAL EXTENTS 90 bytes	Nine groups of fields identical in format to fields 21-24 in the Format 1 label are contained here.
55	POINTER TO NEXT FILE LABEL 5 bytes	This field contains the address(in the form CCHHR) of another Format 3 label if additional extents must be described. Otherwise, it is all binary zeros.

STANDARD DASD FILE LABEL , FORMAT 4

This format is used to describe the Volume Table of Contents and is always the first label in the VTOC. There must be one and only one of these Format 4 file labels per volume.



FIELD	NAME AND LENGTH	DESCRIPTION
1	KEY FIELD 44 bytes, binary	Each byte of this field contains the Hex code 04 in order to provide a unique key.
2	FORMAT ID 1 byte, EBCDIC numeric	4 : Format 4 .
3	LAST ACTIVE FORMAT 1 5 bytes	Contains the address(in the form CCHHR) of the last active Format 1 file label. It is used to stop a search on a file name. (OS/VS only)
3	NEXT AVAILABLE ALTERNATE TRACK	Contains the address(in the form CCHH) of the next track to be assigned as an
5	4 bytes	alternate for a bad track.
6	NUMBER OF ALTERNATE TRACKS 2 bytes, binary	Contains the number of alternate tracks available.
7	VTOC INDICATORS	Bit 0, if on indicates no DADSM(Format 5) label, or DADSM label does not reflect true status of volume. Bits 1,2 not used. Bit 3, if on indicates a stacked pack for the 1401/1440/1460 Emulator Program. Bits 4-7 are not used.

STANDARD DASD FILE LABEL , FORMAT 4 (....Cont'd)

FIELD	NAME AND LENGTH	DESCRIPTION
8A	NUMBER OF EXTENTS 1 byte	Contains the hexadecimal constant 01, to indicate one extent in the VTOC.
8B	RESERVED 2 bytes	Reserved
9	DEVICE CONSTANTS 14 bytes	This field contains constants describing the device on which the volume was mounted when the VTOC was created. The following describes each of the subfields:

Device Size: (4 bytes)-The number of cylinders(CC) and tracks per cylinder(HH).

Track Length: (2 bytes)-The number of available bytes on a track exclusive of home address and record zero. (record zero is assumed to be a nonkeyed record with an eight bytes data field)

Record Overhead: (3 bytes)-The number of bytes required for gaps, check bits, and count field for each record. This value varies according to the record characteristics and thus is broken down into three subfields.

three

I: Overhead required for a keyed record other than the last record on the track.

L: Overhead required for a keyed record that is the last record on the track.

K: Overhead bytes to be subtracted from I or L if the record does not have a key field.

Flag: (1 byte)-Further defines unique characteristics of the device.

Bits: 0-5 : Reserved
6/7 : CC and HH must be used as 1-byte values, as in the case of the 2321. A tolerance factor must be applied to all but the last record on the track.

Tolerance: (2 bytes)-A value that is to be used to determine the effective length of the record on the track. The effective length of a record is calculated in the following manner :
1 Add the key length to the data length of the record
2 Test bit 7 in the flag byte :
a. If 0, go to step 3
b. Multiply value from step 1 by the tolerance factor
c. Shift result 9 bits to the right
3. Add overhead bytes to the result.
Note : Step 2 is not required if the calculation is for the last record on the track.

Labels/track: (1 byte)-A count of the number of labels that can be written on each track in the VTOC. (Number of full records of 44-bytes key and 96-bytes data lengths that can be contained on one track of this device)

Directory Blocks/Track: (1 byte)-A count of the number of directory blocks that can be written on each track for an OS/VS partitioned data set. (Number of full records of 8-bytes key and 256-bytes data lengths that can be contained on one track of this device)

FIELD	NAME AND LENGTH	DESCRIPTION
9	DEVICE CONSTANTS (...Cont'd)	The following illustrates the device constants field for the various direct access devices:

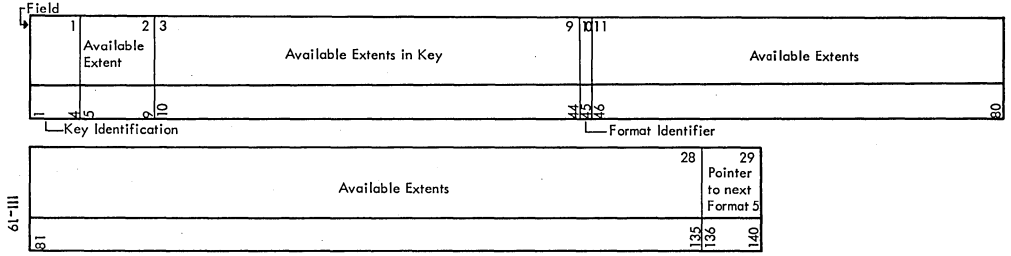
Device	CC	HH	Track Length	I	L	K	Flag	Tol.	Labels Track
2311	203	10	3625	81	20	20	1	537	16
2314/2319	203	20	7294	146	45	45	1	534	25
3330	411	19	13165	191	191	56	0	512	39
3330-11	815	19	13165	191	191	56	0	512	39
3340/35MB	350	12	8535	242	242	75	0	512	22
3340/70MB	700	12	8535	242	242	75	0	512	22
3350	555	30	19254	267	267	82	8	512	46

Note : Labels per track = $\frac{\text{track length}}{\text{I + label length}}$

FIELD	NAME AND LENGTH	DESCRIPTION
10A	VSAM INDICATORS (19 bytes)	The VSAM indicators are present on any volume that contains space allocated to VSAM. If a volume does not contain any VSAM-owned space, these fields are set to zeros.
	VSAM TIMESTAMP (8 bytes)	Is the time the most recent VSAM data space was added to the volume.
10A (Cont'd)	VSAM CATALOG (1 byte)	Bit0-1 : This volume is owned by a VSAM catalog. Bit 1-7: Unused
	TT of CRA (2 bytes)	Relative track number of first track of catalog recovery area
	VSAM TIMESTAMP (8 bytes)	
10B	RESERVED (10 bytes)	Reserved
11-14	VTOC EXTENT	These fields describe the extent of the VTOC and are identical in format to fields 21-24 of the Format 1 file label. Extent type 01 (prime date area).
15	RESERVED (25 bytes)	Reserved

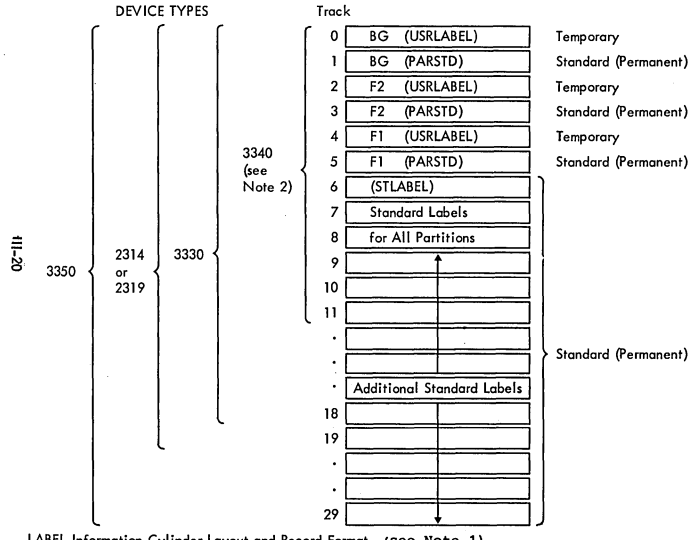
STANDARD DASD FILE LABEL , FORMAT 5

(Note: Format 5 Label used by OS/VS only)



FIELD	NAME AND LENGTH	DESCRIPTION
1	KEY IDENTIFICATION 4 bytes	Each of these four bytes is an hex 05.
2	AVAILABLE EXTENT 5 bytes	This field indicates an extent of space available for allocation to a data file. The first two bytes are relative track address. The next two are the number of full cylinders included in the extent. The last byte is the number of tracks in addition to the cylinders in the extent.
3-9	AVAILABILITY EXTENTS IN KEY 35 bytes	These fields are identical to field 2. They are in relative track address sequence.
10	FORMAT IDENTIFIER 1 byte EBCDIC numeric	5 : Format 5
11-28	AVAILABLE EXTENTS 90 bytes	These fields are the same as field 2. There are 26 available extents fields in the Format 5 label.
29	POINTER TO NEXT FORMAT 5	Contains the address(in the form CCHHR) of the next Format 5 file label if one exists.

LABEL INFORMATION CYLINDER



LABEL Information Cylinder Layout and Record Format (see Note 1)

Note 1: The layout of the label information cylinder depends on the number of partitions defined in your system. This example assumes that three partitions are present. For more information see **DOS/VSE Serviceability Aids and Debugging Procedures**.

Note 2: For the 3340, a second label information cylinder is provided containing 12 standard label tracks for all partitions.

LIOCS MODULE NAME VERSUS OPTIONS

character # ➡	1	2	3	4	5	6	7	8
CDMOD	I	J	C	F RECFORM=FIXUNB (always for TYPEFLE=INPUT, TYPEFLE=CMBND or FUNC=I files) V RECFORM=VARUNB U RECFORM=UNDEF	A CTLCHR=ASA(not specified if CMBND) Y CTLCHR=YES C CONTROL=YES Z CTLCHR or CONTROL not specified	B RDONLY=YES and TYPEFLE=CMBND C TYPEFLE=CMBND H RDONLY=YES and TYPEFLE=INPUT I TYPEFLE=INPUT N RDONLY=YES and TYPEFLE=OUTPUT O TYPEFLE=OUTPUT	Z WORKA and IOAREA2 not specified W WORKA=YES I IOAREA2=YES B WORKA and IOAREA2 Z WORKA=YES not specified (CMBND file only)	0 DEVICE=2540, 3881 1 DEVICE=1442, 2596 2 DEVICE=2520 3 DEVICE=2501 4 DEVICE=2540 and CRDERR 5 DEVICE=2520 and CRDERR 6 DEVICE=3505 or 3504 7 DEVICE=3525 and FUNC=R/P or omitted 8 DEVICE=2560 and FUNC=R/P or omitted A DEVICE=3525 and FUNC=RP B DEVICE=3525 and FUNC=RW C DEVICE=3525 and FUNC=PW D DEVICE=3525 and FUNC=I E DEVICE=3525 and FUNC=RPW F DEVICE=2560 and FUNC=RP

character # ➔	1	2	3	4	5	6	7	8
CDMOD (Cont'd)								G DEVICE=2560 and FUNC=RW H DEVICE=2560 and FUNC=PW I DEVICE=2560 and FUNC=I J DEVICE=2560 and FUNC=RPW K DEVICE=5425 and FUNC=RP L DEVICE=5425 and FUNC=RW M DEVICE=5425 and FUNC=PW N DEVICE=5425 and FUNC=I O DEVICE=5425 and FUNC=RPW
DIMOD	I	J	J	F always	C non-RPS version of module V RPS data set	B TYPEFLE=OUTPUT (processes both in put and output) I TYPEFLE=INPUT	I IOAREA2=YES Z IOAREA2=YES is not specified	C RDONLY=YES D RDONLY=YES is not specified
DRMOD	I	J	M	Z	S SETDEV=YES Z SETDEV=YES not specified	R RDONLY=YES Z RDONLY=YES not specified	D always	0 always
DUMOD	I	J	N	D	I =DUMODFI O =DUMODFO	C ERROPT=YES and ERREXT=YES E ERROPT=YES Z neither is specified	Z	Y RDONLY=YES Z RDONLY not specified

character # ➡	1	2	3	4	5	6	7	8
MRMOD	I	J	U	S Address=SINGLE D Address=DUAL	Z	Z	Z	Z
MTMOD (GET/PUT)	I	J	F	F RECFORM=FIXUNB (or FIXBLK) (EBCDIC mode) X RECFORM=FIXUNB (or FIXBLK) (ASCII mode) V RECFORM=VARUNB (or VARBLK) (EBCDIC mode) R RECFORM=VARUNB (or VARBLK) (ASCII mode) S RECFORM=SPNUNB (or SPNBLK) (spanned records) U RECFORM=UNDEF (EBCDIC code) N RECFORM=UNDEF (ASCII mode)	B READ=BACK Z READ=FORWARD or if READ is not specified	C CKPTREC=YES Z CKPTREC=YES is not specified	W WORKA=YES Z WORKA=YES is not specified	M ERREXT=YES and RDONLY=YES N ERREXT=YES Y RDONLY=YES Z ERREXT and RDONLY not specified
MTMOD (WORKFILE)	I	J	F	W always	E ERROPT=YES Z ERROPT is not specified	N NOTEPNT=YES S NOTEPNT=POINTS Z NOTEPNT is not specified	Z always	M ERREXT=YES and RDONLY=YES N ERREXT=YES Y RDONLY=YES Z ERREXT and RDONLY not specified

character # ➡	1	2	3	4	5	6	7	8
ORMOD	I	J	M	F RECFORM=FIXUNB X RECFORM=FIXBLK U RECFORM=UNDEF D RECFORM=UNDEF and BLKFAC=YES	C CONTROL=YES Z CONTROL=YES is not specified	I IOAREA2=YES W WORKA=YES B both are specified Z neither is specified	T device is in tape mode D device is in document mode	Z always
PRMOD	I	J	D	F RECFORM=FIXUNB V RECFORM=VARUNB U RECFORM=UNDEF	A CTLCHR=ASA Y CTLCHR=YES C CONTROL=YES S STLIST=YES Z none of these is specified T DEVICE=3525 with 2-line printer U DEVICE=2560 V DEVICE=5425	B ERROPT=YES and PRINTOV=YES P PRINTOV=YES, DEVICE is not 3525 and ERROPT is not specified I PRINTOV=YES, DEVICE=3525 and FUNC=WE[T] or omitted F PRINTOV=YES, DEVICE=3525 and FUNC=RW[T] C PRINTOV=YES, DEVICE=3525 and FUNC=PW[T] D PRINTOV=YES, DEVICE=3525 and FUNC=RPW[T] Z PRINTOV=YES and ERROPT not specified and DEVICE is not 2560, 3525, or 5425	I IOAREA2=YES Z IOAREA2=YES is not specified	V RDONLY=YES and WORKA=YES W WORKA=YES Y RDONLY=YES Z neither is specified

LIOCS MODULE NAME VERSUS OPTIONS (....Cont'd)

character # ➡	1	2	3	4	5	6	7	8
PRMOD (Cont'd)						O PRINTOV=YES not specified, DEVICE= 3525 and FUNCT= W[T] or omitted R PRINTOV=YES not specified, DEVICE= 3525 and FUNCT= RW[T] S PRINTOV=YES not specified, DEVICE= 3525 and FUNCT= PW[T] T PRINTOV=YES not specified, DEVICE= 3525 and FUNCT= RPW[T] E ERROPT=YES and PRINTOV=YES is not specified U FUNC=W or omitted and DEVICE=2560 or 5425 V FUNC=RW and DEVICE=2560 or 5425 W FUNC=PW and DEVICE=2560 or 5425 X FUNC=RPW and DEVICE=2560 or 5425		

LIOCS MODULE NAME VERSUS OPTIONS (....Cont'd)

character # ➡	1	2	3	4	5	6	7	8
PTMOD	I	J	E	S SCAN=YES Z SCAN=YES is not specified	T TRANS=YES (SCAN) =YES is not specified) Z TRANS=YES is not specified	F RECFORM=FIXUNB and SCAN=YES U RECFORM=UNDEF and SCAN=YES Z SCAN=YES is not specified and/or DEVICE=1018	1 DEVICE=1017 2 DEVICE=1018 Z DEVICE=2671 or if this entry is omitted	Z always
SDMOD (GET/PUT)	I	J	G	C SDMODFx specifies HOLD=YES F SDMODFx does not specify HOLD=YES R SDMODUx specifies HOLD=YES U SDMODUx does not specify HOLD=YES P SDMODVx specifies HOLD=YES and RECFORM=SPNBLK/SPNUNB Q SDMODVx does not specify HOLD=YES and specifies RECFORM=SPNBLK/SPNUNB S SDMODVx specifies HOLD=YES V SDMODVx does not specify HOLD=YES	I SDMODxI (non-RPS version of module) O SDMODxO (non-RPS version of module) U SDMODxU (non-RPS version of module) W SDMODxI (RPS input data set) X SDMODxO (RPS output data set) Y SDMODxU (RPS update data set)	C ERROPT=YES and ERREXT=YES E ERROPT=YES Z neither is specified	M TRUNCS=YES and FEOVD=YES T TRUNCS=YES W FEOVD=YES Z neither is specified	B CONTROL=YES and RDONLY=YES C CONTROL=YES Y RDONLY=YES Z neither is specified

LIOCS MODULE NAME VERSUS OPTIONS (....Cont'd)

Character # ➡	1	2	3	4	5	6	7	8
SDMOD (WORK FILES)	I	J	G	T HOLD=YES W HOLD=YES not specified	C ERROPT=YES and ERREXT=YES E ERROPT=YES W RPS version of module Z neither is specified	N NOTEPNT=YES R NOTEPNT=POINT RW Z NOTEPNT is not specified	C CONTROL=YES Z CONTROL=YES is not specified	T RDONLY=YES and UPDATE=YES U UPDATE=YES Y RDONLY=YES Z neither is specified
DAMOD	I	J	I	F RECFORM=FIXUNB B RECFORM=UNDEF handles both UNDEF and FIXUNB S RECFORM=SPNUNB V RECFORM=VARUNB	A AFTER=YES W RPS version of module Z AFTER is not specified	E IDLOC=YES and FEOVD=YES I IDLOC=YES R FEOVD=YES Z neither is specified	H ERREXT=YES and RELTRK=YES P ERREXT=YES R RELTRK=YES Z neither is specified	W HOLD=YES and RDONLY=YES X HOLD=YES Y RDONLY=YES Z neither is specified
ISMOD	I	J	H	A RECFORM=BOTH, IOROUT=ADD or ADDRTR B RECFORM=FIXBLK, IOROUT=ADD or ADDRTR U RECFORM=FIXUNB, IOROUT=ADD or ADDRTR Z RECFORM is not specified (IOROUT=LOAD or RETRVE)	A IOROUT=ADDRTR (non-RPS version of module) I IOROUT=ADD L IOROUT=LOAD (non-RPS version of module) R IOROUT=RETRVE V IOROUT=ADDRTR (RPS version of module) X IOROUT=LOAD (RPS version of module)	B TYPEFLE=RANSEQ G IOAREA2=YES TYPEFLE=SEQNTL or IOROUT=LOAD R TYPEFLE=RANDOM S TYPEFLE=SEQNTL Z neither is specified (IOROUT=LOAD or ADD)	B CORINDX=YES and HOLD=YES C CORINDX=YES O HOLD=YES Z neither is specified	F CORDATA=YES, ERREXT=YES, RDONLY=YES G CORDATA=YES and ERREXT=YES O CORDATA=YES and RDONLY=YES P CORDATA=YES S ERREXT=YES and RDONLY=YES T ERREXT=YES Y RDONLY=YES Z neither is specified

Bytes		Bits	Contents	Function
Dec	Hex			
0-15	00-FF			CCB
8	8			1= 2501 double - CCW support
16	10	0		1= OMR [1]; 0= Omitted
		1		1= ERROPT [2]; 0= Omitted
		2		COBOL open; ignore option
		3		1= GET issued [3]; 0= GET not issued [7]
		4		DTF table address constants relocated by OPENR
		5-7		File association:
				000= READ only
				010= READ/PRINT [4]
				101= READ/PUNCH/PRINT [5]
				001= READ/PUNCH [5]
17-19	11-13			Address of logic module
20	14		X'02'	DTF type
			X'05'	DTF type for 2560 or 5424/5425
21	15	0		1= Open; 0= Closed
		1		First time switch
		2		1= 1442 or 2596; 0= Other
		3		1= 2560, 3525 or 5424/5425; 0=Other
		4		1= 3504, 3505; 0= Other
		5		1= 2 I/O areas; 0= 1 I/O area
		6		1= 2520; 0= Other
		7		1= 2540; 0= Other
22	16		B'SSF0X010'	Normal command code (not for 2560 or 5424/5425):
				SS: 00= pocket 1; 01= pocket 2; 10= pocket 3 [6]
				F : 1= Column binary [3]; 0= EBCDIC
				X: 1= OMR or RCE [3]; 0= neither
			B'H0B00010'	Read command code (2560):
				H : 0= hopper 1; 1= hopper 2
				B : 0= EBCDIC; 1= column binary
			B'HMMM0011'	SS command code (5425).
				H: 0= hopper1, 1=hopper 2.
				MMM: 001=stacker 1,010=stacker 2. 011=stacker 3, 100=stacker 4.
23	17		B'H0B00010'	Control command code (not for 2560 or 5424/5425).
				Read command code (2560, 5425)
24-27	18-1B			Address of IOAREA2
28	1C	0		1= 2560; 0= Other
		1		1= 5424/5425; 0= Other
		2-7		Not used
29-31	1D-1F			Address of EOF routine
32-39	20-27			Read CCW
Bytes 40-49 as used for all files except 2560 and 5424/5425 files				
40-43	28-2B		LA &IOREG,0(14) NOP 0	Load user pointer register
44-49	2C-31		MVC 0(&BLKSIZE, 13),0(14) NOP 0 DC X'0000'	Move IOAREA to WORKA

Bytes Dec	Bytes Hex	Bits	Contents	Function
The following bytes (50-105) are used for 2501 double-CCW support.				
50-55	32-37			Unused CCB
56-71	38-47			
72	48	0		1= OMR[1], 0= omitted.
		1		1= ERROPT[2], 0= omitted.
		2		COBOL open; ignore option
		3		1= GET issued[3], 0 = GET not issued[7].
		4		DTF table address constants relocated by OPENR.
		5-7		File Association 000 = READ only 010 = READ/PRINT[4] 101 = READ/PUNCH/PRINT[5] 001 = READ/PUNCH[5].
73-75	49-4B			Address of logic module.
76	4C		X'02'	DTF type.
			X'05'	DTF type for 2560 or 5424/5425.
77	4D	0		1=open; 0= closed
		1		First time switch
		2		1= 1442 or 2596; 0 = other.
		3		1= 2560, 3525, or 5424/5425; 0=Other.
		4		1= 3504 or 3505; 0= other.
		5		1= 2 I/O areas; 0 = 1 O/I area.
		6		1= 2520; 0 = other.
		7		1= 2540; 0 = other.
78	4E		B'SSF0X010'	Normal command code (not for 2560 or 5424/5425. SS : 00 = pocket1, 01 = pocket2, 10 = pocket 3[6]. F : 1= column binary[3], 0= EBCDIC. X : 1 = OMR or RCE[3], 0 = neither.
			B'H0B00010'	Read command code (2560, 5424/5425) H : 0= hopper 1, 1 = hopper2. B : 0 = EBCDIC, 1 = column binary.
79	4F			Control command code (not for 2560 or 5424/5425).
			B'H0B00010'	Read command code (2560, 5424/5425).
80-83	50-53			Address of IOAREA2. (If IOAREA2 is not specified, address of IOAREA1).

Bytes Dec	Bytes Hex	Bits	Contents	Function
84	54	0 1 2-7		1 = 2560; 0 = other. 1 = 5424/5425; 0 = other. Not used.
85-87	55-57			Address of EOF routine.
88-95	58-5F			Read CCW.
96-99	60-63		LA &IOREG, 0(14) NOP 0	Load user pointer register.
100-103	64-67		MVC 0 (&BLKSIZE, 13), 0(14) NOP 0	Move IOAREA to WORKA
104-105	68-69		DC X'0000'	
The following bytes (50-57) are used for 3504, 3505, and 3525 associated files.				
50-53	32-35		DC A (name) B 16(15) B 20(15) DC F'0'	If ERROPT=name[2]. If ERROPT=SKIP. If ERROPT=IGNORE. If ERROPT=omitted.
54-57	36-39		DC A(ASOCFLE)	Address of associated DTF table[7]. (3525 only).
Bytes 40 onward as used for 2560 and 5425 files.				
40-47	28-2F			Stacker select CCW (2560). Read CCW (5424/5425).
48-51	30-33		LA &IOREG,0(14) NOP 0	
52-57	34-39		MVC 0(&BLKSIZE, 13), 0(14) NOP 0 DC X'0000'	Move IOAREA to WORKA
58-63	3A-3F		CLC 0(L, 14), 64(1)	Test for end of file. L=4 if MODE=C; L=2 in other cases.
64-67	40-43		DC C'/* ' DC X' 0C001022'	End-of-file indicator if MODE=E In other cases.

Bytes Dec	Bytes Hex	Bits	Contents	Function
68-71	44-47		DC A(name) B 16(15) B 20(15) DC F'0'	If ERROPT=name[2]. If ERROPT=SKIP. If ERROPT=IGNORE. If ERROPT=omitted.
72-75	48-4B		DC A(ASOCFLE)	Address of associated DTF table[7].
76-81	4C-51		MVC 0 (&BLKSIZE, 14), 82(1)	Move card image to IOAREA1
82	52		DC &BLKSIZE.C' '	Buffer for card image.

1 OMR only for 3504 and 3505.
2 ERROPT for 2560, 3504, 3505, 3525, 5424/5425 READ file.
3 3504, 3505, and 3525 with or without CONTROL=YES specified.
4 2560, 3525, or 5424/5425 with or without CONTROL=YES specified.
5 2560, 3525, or 5424/5425 without CONTROL=YES specified.
6 Defaults to pocket2 for 3504, 3505, and 3525.
7 Present only when 2560, 3525, or 5424/5425 associated files are specified for the input DTF.

DTFCD (Punch)

Bytes Dec	Bytes Hex	Bits	Contents	Function
0-15	00-FF			CCB
16	10	0		Not used
		1		1= ERROPT [3]; 0= Omitted
		2		COBOL open; ignore option
		3		1= PUT issued [2]; 0= PUT not issued
		4		DTF table address constants relocated by OPENR
		5-7		File Association: 000= PUNCH only 011= PUNCH/PRINT [3] 001= READ/PUNCH [3] 101= READ/PUNCH/PRINT [3] 100= PUNCH/INTERPRET [3]
17-19	11-13			Address of logic module
20	14		X'04'	DTF type
21	15	0		1= Open; 0= Closed
		1		First time switch
		2		1= CTLCHR
		3		1= Fixed unblocked
		4		1= Variable unblocked
		5		1= 2 I/O areas
		6		1= Workarea
		7		1= 2 CCWs in table; 0= 1 CCW in table
22	16		B'SSF00001'	Normal command code: SS : 00= pocket 1; 01= pocket 2; 10= pocket 3 [4] F : 1= column binary; 0= EBCDIC
			B'HSSS0011'	Normal stacker select command code (2560 or 5424/5425). H : 0= hopper 1; 1= hopper 2 SSS: stacker information
23	17		B'HSSS0011'	Control command code (not for 2560 or 5424/5425). Actual stacker select command code (2560 or 5424/5425)
24-27	18-1B		DC A(IOAREA1+x)	Address of data in IOAREA1
28-31	1C-1F			Bucket [1]
32-33	20-21		LR 12,(RECSIZE)	Undefined records only
34-37	22-25		LA &IOREG,4(14) NOPR 0	Load user pointer register
38	26	0-2		Not used
		3		1= 5424/5425
		4		1= 2560
		5		1= 3525
		6		1= 1442 or 2596
		7		1= 2520B1
39	27		DC C' '	Blank for eject last card

Bytes Dec	Bytes Hex	Bits	Contents	Function
For all files except 2560 and 5425 files:				
40-47	28-2F			Punch CCW
48-55	30-37			Eject CCW for last card if 2520
For 2540 files if CRDERR is specified				
48-55	30-37			Retry CCW
56-135	38-87		DC CL80' '	Savearea card image
For 3525 PUNCH/INTERPRET files				
48-55	30-37			Load CCW
56-63	38-3F			Print CCW
64-127	40-7F		DC 64C' '	Print buffer
For 3525 Associated files				
48-51	30-33		DC A(ASOCFLE)	Pointer to associated file
For 2560 and 5424/5425 files				
40-47	28-2F		DC D'0'	Eject CCW If FUNC= RP or RPW
48-55	30-37			Stacker select CCW
56-63	38-3F			Punch and Feed CCW
For 2560 PUNCH/INTERPRET files				
64-71	40-47			Load print head buffer 1 CCW
72-79	48-4F			Load print head buffer 2 CCW
80-87	50-57			Print CCW
88-151	58-97		DC 64C' '	Save area for printing line 2
For 5425 PUNCH/INTERPRET files				
64-71	40-47			Print CCW
For 2560 and 5424/5425 Associated files				
64-67	40-43		DC A(ASOCFLE)	
68	44		DC C' '	If mode is EBCDIC
			DC X'00'	If mode is Column Binary
69-	45-		DC &BLKSIZE.C' '	Buffer for card image

1) The bucket bytes handle undefined length records
2) Valid for 2560 or 3525 READ/PUNCH, PUNCH/PRINT, and READ/PUNCH/PRINT files
3) Valid for 2560 or 3525 only. (3504, 3505, 5425) non-associated files
4) Defaults to pocket 2 for 3525

Bytes		Bits	Contents	Function
Dec	Hex			
0 -15	00-0F			CCB
16	10	0-1 2 3 4 5-7		Not used COBOL open; ignore option Not used OPENR relocates DTF address constants Not used
17-19	11-13			Address of logic module
20	14		X'00'	DTF type
21	15			Command code (X'02' for 1442, X'C2' for 2520, 2540)
22	16			Command code (X'01' for 1442, X'09' for 2520, 2540)
23	17			Command code (X'01' for 1442, X'09' for 2520, 2540)
24-31	18-1F			CCW
32-35	20-23			Input area address
36-39	24-27			Output area address
40-41	28-29			Input blocksize
42-43	2A-2B			Ouput blocksize
44-49	2C-31		MVC 0 (&BLKS, 13,0(14)	
50-55	32-37		MVC 0 (&OUBL, 14),0(13)	
56-59	38-3B			End-of-file address
60-67	3C-43			Save area
68-73	44-49		MVC 1 (&OUBL-1, 13),0(13)	
74-77	4A-4D		MVI 0(13),X'40'	
78-79	4E-4F			Constant (blanks)
80-83	50-53			Constant address (bytes 78-79)

Bytes Dec	Bytes Hex	Bits	Contents	Function
0 -15	00-0F			CCB
16	10	0		1= 2-line printer (3,4); 0= Other
		1		1= ERROPT (3,4); 0= Omitted
		2		COBOL open; ignore option
		3		1= 3525; 0= Other
		4		OPENR relocates DTF address constants
		5-7		000= PRINT only 011= PUNCH/PRINT 3) 010= READ/PRINT 3) 101= READ/PUNCH/PRINT 3)
17-19	11-13			Address of logic module
20	14		X'08' X'07'	DTF type DTF type for 2560 and 5424/5425
21	15	0		1= Open; 0= Closed
		1		First time switch
		2		1= Control character
		3		1= Fixed unblocked records
		4		1= Variable unblocked records
		5		1= 2 I/O areas
		6		1= Workarea
		7		1= Print overflow channel 9
For Printer and Card Punch devices				
22	16		X'09'	Normal command code 5)
23	17		X'09'	Control command code 5)
24-27	18-1B		DC A(IOAREA1+x)	Address of data in IOAREA1
28-31	1C-1F			Bucket 1)
32-33	20-21		LR 12,(RECSIZE) NOPR 0	For undefined records only
34-37	22-25		LA &IOREG,4(14) NOP 0	Only if IOREG= (r)
38-39	26-27			Bucket 2)
40-47	28-2F		11,*,X'60',1 9,IOAREA,X'20',121	CCW- Set up Selective Tape List Control STLIST not specified 6)
48-55	30-37		9,IOAREA,X'20',121	CCW- STLIST specified 6)
			A (Name)	Address of user error routine (for all the 3211-compatible printers identified by device type code PRT1)
			DC A(ASOCFLE)	If ASOCFLE= filename 3)
For the 2560 and 5424/5425 Multi Function Card Machine				
22	16		X'00'	Not used
23	17		B'HHHHHH00'	Print head selection byte H= 1 specifies the corresponding head
24-27	18-1B			Address of IOAREA1

Bytes Dec	Bytes Hex	Bits	Contents	Function
For the 2560 Multi Function Card Machine (Cont'd)				
28-31	1C-1F			Bucket
32-33	20-21		LR 12,(RECSIZE) NOPR 0	For undefined records only
34-37	22-25		LA &IOREG,4(14) NOP 0	Only if IOREG= (r)
38-39	26-27			Number of bytes to be printed by the last specified print head
40-43	28-2B		DC A(ASOCFLE) DC F'0'	If FUNC= RW,PW or RPW In all other cases
44	2C	0 1 3 4-7		1= 2560 Not used 1= Print control switch for 2560 associated files Not used
45-47	2D-2F		DC 3X'00'	Reserved for future use
For 2560 simple files				
48-55	30-37			Eject CCW
56-63	38-3F			Load print head buffer CCW
64-71	40-47			Print CCW
For 2560 associated files				
48-55	30-37			Load print head buffer CCW
56-63	38-3F			Print CCW
For 5424/5425 files				
48-55	30-37			Print CCW

1) The bucket bytes handle undefined records. Bit 0 of byte 28 at open time determines the mode set of a printer with UCS. If bit 0= 1, the mode is set so that data checks occur if an invalid character is printed. Otherwise, mode is set to suppress data checks. The use of the UCS parameter determines the setting of this bit. If STLIST= YES, byte 31 saves the STLIST control byte provided by the PUT macro.

2) The 2 byte bucket saves print overflow conditions if CTLCHR= ASA. If STLIST= YES, byte 38 contains the current STLIST control byte. Byte 39 is set by the PUT macro to indicate spacing or skipping. (X'00' no spacing, no skipping; X'01' spacing; X'02' skipping).

3) Valid for 2560, 3525 READ/PRINT, PUNCH/PRINT and READ/PUNCH/PRINT files.

4) Valid for 3525 PRINT only files.

5) X'05' for 3525; X'09' for other devices

6) Valid for 1403 only

Bytes Dec	Bytes Hex	Contents	Function
0 -15	00-0F		CCB
16	10	X'20' X'08'	COBOL open; ignore option DTF table address constants relocated by OPENR
17-19	11-13		Address of logic module: GET and PUT logic if TYPEFLE= INPUT; PUT logic if TYPEFLE= OUTPUT; GET, PUT and PUTR logic if TYPEFLE= CMBND
20	14	X'03'	DTF type
21-23	15-17		For input and output: not used For combined: byte 21 contains X'01' and bytes 22-23 contain INPSIZE
24-31	18-1F	X'09', IOAREA1, X'00', BLKSIZE	CCW
End of table if RECFORM= FIXUNB and WORKA not specified. The following bytes are added if WORKA is specified.			
32-35	20-23	DC A(IOAREA1)	Address of I/O area
36-39	24-27	DC F'0'	Register save area
40-43	28-2B	DC F'0'	Register save area
.End of table if RECFORM= FIXUNB. The following bytes are added if RECFORM= UNDEF			
		DC F'0'	Register save area
		DC F'0' DC H'BLKSIZE' DC AL2(BLKSIZE-1)	Register save area I/O area size For input files only
The following bytes are added to the table if TYPEFLE= CMBND			
32-35	20-23	DC A(IOAREA1+BLKSIZE)	I/O area address for input
36-37	24-25	DC H'BLKSIZE'	Blocksize

Bytes Dec	Bytes Hex	Bits	Contents	Function
0 -15	00-0F			CCB
16	10	0-1		Not used
		2		COBOL open; ignore option
		3		Not used
		4		OPENR relocates DTF table addresses
		5-7		Not used
17-19	11-13			Address of logic module
20	14		X'C0'	DTF type
21	15			PIOCS switches:
		0		1= open; 0= closed
		1	B'1'	Input
		2-5	B'0000'	Not used
		6	B'1'	Device is 3886
		7	B'0'	Not used
22	16			Error indicator byte
23	17			LIOCS switches:
		0-4	B'00000'	Not used
		5		1= SETDEV
		6		1= Control passed to COREXIT
		7		1= FR loaded from disk
24-31	18-1F			FR phasename at open time
32-39	20-27			Phasename of currently used FR
40-43	28-2B		X'00000000'	Not used
44-47	2C-2F			Start address of FR area in DTF
48-51	30-33			Address of four-byte pointer at the end of the FR area in the DTF
52-55	34-37			EOF routine address
56-63	38-3F			Scan CCW
64-71	40-47			Read CCW
72-79	48-4F			Read CCW
80-87	50-57			Control CCW
88-95	58-5F			Load format record CCW
96-99	60-63			COREXIT routine address
100-103	64-67			IOAREA1 area address
104-107	68-6B			Header area address
108-111	6C-6F			Exit indicator address
112	70			Start of FR area

Bytes Dec	Bytes Hex	Bits	Function
0 -15	00-0F		Dummy CCB
16	10	0-1	Not used
		2	COBOL open; ignore option
		3	Not used
		4	DTF table address constants relocated by OPENR
		5-7	Not used
17-19	11-13		Address of logic module
20	14		DTF type, (X'09') DTF type, (X'0A' if HEADER= YES)
21	15		PIOCS switches:
		0	1= Open; 0= closed
		1	1= Input
		2	1= Control
		3	1= Device is 1287
		4	1= Header
		5	Reserved for future use
		6	1= RDLNE
		7	Not used
22	16		Not used
23	17	0-6	Not used
		7	1= LIOCS posts a hopper empty condition to DTF
24-39	18-27		CCB
40-47	28-2F		Sense CCW
48-51	30-33		Lost lines (equipment check)
52-55	34-37		After 9 retries for journal tape, or after 2 retries for documents
56-59	38-3B		Wrong length records
60-63	3C-3F		After 4 retries for journal tape, or after 2 retries for documents
64-67	40-43		Keyboard corrections
68-71	44-47		Count of data check errors
72-75	48-4B		Lines marked
76-79	4C-4F		Total lines read (CCW chains executed)
80	50		Error indicators:
		0	1= EOP
		1	1= Lost reference mark indicator
		2	1= Late stacker selection
		3	1= Non-recovery error
		4	1= Equipment check
		5	1= Wrong length record
		6	1= Hopper empty
		7	1= Data check
81	51		LIOCS switches:
		0	1= First time
		1	1= 2 I/O areas

Bytes		Bits	Function
Dec	Hex		
81 (Cont'd)		2 3 4 5-7	1=WORKA=YES 1=RECFORM=FIXUNB 1=RECFORM=UNDEF Not used
82	52		Normal command code
83	53		Control command code
84-87	54-57		IOAREA2 address
88-95	58-5F		Read CCW
96-103	60-67		Go to next line CCW
104-111	68-6F		Control CCW
112-115	70-73		EOF address
116-119	74-77		Correction exit address
120-123	78-7B		IOAREA1 address
124-127	7C-7F		DC A(&BLKS-1)
128-129	80-81		SR 13,&RECS
130-131	82-83		LR &RECS,13
132-133	84-85		LR &IOR,13
134-135	86-87		Sense

Bytes Dec	Bytes Hex	Bits	Function
0 -5	00-05		CCB indicators
6 -7	06-07		Logical class and unit numbers (primary if DUAL addressing)
8	08		Zero
9 -11	09-0B		CCW address
12-15	0C-0F		Zeros
16	10	0-1	Not used
		2	COBOL open; ignore option
		3	Not used
		4	DTF table address constants relocated by OPENR
		5-7	Not used
17-19	11-13		Address of logic module
20	14		DTF type= X'0B'
21	15		Logic module option switches:
		0	User disengage 0= off; 1= on
		1	Program sort mode 0= no; 1= yes
		2	First time switch (after engage) 0= no; 1= yes
		3	Addressing= DUAL 0= no; 1= yes
		4	Waiting 0= no; 1= yes
		5	Read logic indicator 0= no; 1= yes
		6	Not used
		7	Supervisor initial read (after open) 0= no; 1= yes
22-29	16-1D		Symbolic filename
30	1E		Open/Close switch:
		0	0= closed; 1= open
31-33	1F-21		Open/Close option switches
34-35	22-23		Logic module option switches
36-39	24-27		Error information status
40-41	28-29		Length of DTF table
42-43	2A-2B		Device type indicator
44-45	2C-2D		Record type
46-49	2E-31		Reserved for future use
50-51	32-33		I/O register
52-55	34-37		End-of-file address
56-59	38-3B		IOAREA2/1 address
60-63	3C-3F		Document buffer size
64-65	40-41		Blocking factor/Number of buffers
66-67	42-43		I/O area size
68-71	44-47		Record length
72-76	48-4C		Sense information

Bytes Dec	Bytes Hex	Bits	Function
77	4D		Supervisor switch
78-79	4E-4F		Logical class and unit numbers (secondary, for DUAL addressing only)
80-81	50-51		Register alignment bytes
82-83	52-53		Logical class and unit numbers (primary, for DUAL addressing)
84-87	54-57		Document buffer size
88	58		Command code (4C)
89-91	59-5B		Address of last byte of first document buffer
92	5C		Command code (4C)
93-95	5D-5F		Address of last byte of last document buffer
96-99	60-63		Stacker select routine address
100-103	64-67		Address of stacker select CCW chain
104-107	68-6B		Current buffer address pointer (Supervisor)
108-111	6C-6F		Supervisor count
112-113	70-71		Number of buffers minus 7
114-115	72-73		Message indicator
116-119	74-77		ERROPT routine address
120-121	78-79		Logical class and unit numbers (secondary, for DUAL addressing only)
122-123	7A-7B		Reserved for future use
124-127	7C-7F		Address of last buffer given to user
128-131	80-83		Address of first byte of last buffer
132-139	84-8B		Channel status word (CSW)
140-143	8C-8F		Address of active GET record
144-147	90-93		GET counter
148-159	94-9F		Reserved for future use
For single addressing			
160-167	A0-A7		CCW - Engage
168-175	A8-AF		CCW - Read
176-183	B0-B7		CCW - Sense
184-191	B8-BF		CCW - NOP
192-199	C0-C7		CCW - Stacker select
200-207	C8-CF		CCW - TIC
208-215	D0-D7		CCW - Control
216-223	D8-DF		CCW - BN

Bytes		Bits	Function
Dec	Hex		
224-231	E0-E7		CCW - Read
232-239	E8-EF		CCW - Sense
240-247	F0-F7		CCW - Disengage
For DUAL Address Adapter			
160-167	A0-A7		CCW - Engage
168-175	A8-AF		CCW - Read buffer 1
176-183	B0-B7		CCW - Sense
184-191	B8-BF		CCW - NOP
192-199	C0-C7		CCW - Read buffer 2
200-207	C8-CF		CCW - MOD Sense
208-215	D0-D7		CCW - Read buffer 1
216-223	D8-DF		CCW - MOD Sense
224-231	E0-E7		CCW - TIC to NOP
232-239	E8-EF		CCW - NOP
240-247	F0-F7		CCW - MOD CTL
248-255	F8-FF		CCW - Stacker select
256-263	100-107		CCW - MOD Sense

Bytes Dec	Bytes Hex	Bits	Contents*	Function	Record Format
0 -15	00-0F			CCB	
8	08		Input: X'00'-X'63' Output: X'00'-X'04' (variable) X'00' (undefined)	Buffer offset length, ASCII	
16	10	0		First time entered MTMOD for a file	
		1		Not used	
		2		COBOL open; ignore option	
		3		American National Standard COBOL	
		4		DTF table address constants relocated by OPENR	
		5		1= spanned records	
		6		1= ASCII - 0= EBCDIC	V-V,S
		7		ASCII input: 1= Length check	V
				ASCII output: 1= Buffer offset length = 4	V
17-19	11-13			Address of logic module	
20	14		X'11' X'12' X'13' X'14'	Nonstandard or unlabeled Standard labeled, output Standard labeled, input, backwards Standard labeled, input, forwards	
21	15	0		First time switch: 1= not first-time entry 0= first-time entry	
		1		1= blocked 0= unblocked	
		2		1= 2 I/O area's 0= 1 I/O area	
		3		1= workarea 0= no workarea 0= workarea, spanned	F,U,V F,U,V S
		4		1= input 0= output	
		5		1= backwards 0= forwards	
		6		1= checkpoint 0= no checkpoint	
		7		1= TRUNC required during Close	
22-29	16-1D			Symbolic filename	
30	1E			Same as command code in CCW; (X'01', X'02' or X'0C')	
31	1F	0-4		Bits 0-4 are used as displacements by OPEN to determine the location of variable fields of the DTF.	

Bytes Dec	Bytes Hex	Bits	Contents *	Function	Record Format
31 (Cont'd)	1F		B'01110'	Input	F
			B'01100'	Output	F
			B'10001'	Input	V
			B'01111'	Output	V
			B'01101'	Input	U
			B'01011'	Output	U
		5		1= Tape label information included in DTF (see bytes 88-95) 0= Tape label information not included in DTF	
		6		Used by COBOL	
		7		1= Header label and EOV information wanted 0= No header label and EOV information wanted	
32	20	0		Standard labels: 1= yes; 0= no	
		1		Labels: 1= nonstandard; 0= unlabeled	
		2		Rewind unload: 1= yes; 0= no	
		3		Rewind option: 1= no rewind; 0= rewind	
		4		Drive direction: 1= backwards; 0= forwards	
		5		User label address: 1= yes; 0= no	
		6		Tapemark option: 1= no; 0= yes	
		7		EOF-EOV switch (used by IBM SORT): 1= yes; 0= no	
33-35	21-23			User label routine address	
36	24	0		DTFPH: 1= yes; 0= no	
		1		COBOL indicator: 1= yes; 0= no	
		2		File type: 1= input; 0= output	
		3		FEOV switch: 1= yes; 0= no	
		4		EOF-EOV switch (output): 1= EOF 0= EOV	
		5		Open indicator: 1= open; 0= closed	
		6		1= variable or spanned records	V, S
		7		1= undefined records	U
37-39	25-27			EOF address	
40-43	28-2B			Block count	
44-47*	2C-2F		BXH 11,12, 24(15)	Forward	F
			BXLE 11,12 24(15)	Backward	F
			L &VARBLD, DEBLOCKER	If VARBLD parameter is used	V
			NOP 0(0)		S
			DC F'0'	DEBLOCKER1	U

Bytes Dec	Bytes Hex	Bits	Contents*	Function	Record Format
48-51	30-33		LA 14,1(14)		F,V,S
			BCTR 14,0 + NOPRO	Backward	F,V,S
			L &RECSIZE, DEBLOCKER1	If RECSIZE given	U
			NOP 0(0)	For input if not NOP	U
52-55	34-37		L &IOREG, DEBLOCKER1	If IOREG specified	F
			L &IOREG, DEBLOCKER5	If IOREG specified	V
			L EIOREG, DEBLOCKER2	If IOREG specified	U
			NOP 0(0)	If no IOREG	
			L &RECSIZE, IJFVSREC	If spanned input	S
			ST &RECSIZE, IJFVSREC	If spanned output	S
56-63	38-3F			CCW	
64-67	40-43		DC A(IOAREA1)	One I/O area	
			DC A(IOAREA1 +BLKSIZE-1)	One I/O area, read backward	
			DC A(IOAREA2)	Two I/O area's	
			DC A(IOAREA2 +BLKSIZE-1)	Two I/O area's, read backward	
68-71	44-47		DC F'0'	Input	F
			DC A(IOAREA1 +BLKSIZE-RECSIZE)	Input backward: DEBLOCKER1	F
			DC A(IOAREA1)	1 I/O area, output: DEBLOCKER1	F
			DC A(IOAREA2)	2 I/O area's, output: DEBLOCKER1	F
			DC A(BLKSIZE)	DEBLOCKER1: EBCDIC	V,S
			DC A(IOAREA1)	1 I/O area: DEBLOCKER2	U
			DC A(IOAREA2)	2 I/O area's: DEBLOCKER2	U
72-75	48-4B		DC F'RECSIZE'	Forward: DEBLOCKER2	F
			DC F'-RECSIZE'	Backward: DEBLOCKER2	F
			DC A(IOAREA1)	1 I/O area: DEBLOCKER2	V,S
			DC A(IOAREA2)	2 I/O area's: DEBLOCKER2	V,S
			LA 14,1(14)	Forward	U
			BCTR 14,0 + NOPRO	Backward	U
76-79	4C-4F		DC F'0'	Input forward: DEBLOCKER3	F
			DC A(IOAREA1 +BLKSIZE - RECSIZE)	Input backwards: DEBLOCKER3	F
			DC A(IOAREA1 +BLKSIZE-1)	Output, 1 I/O area: DEBLOCKER3	F
			DC A(IOAREA2 +BLKSIZE-1)	Output, 2 I/O area's: DEBLOCKER3	F

Bytes Dec	Bytes Hex	Bits	Contents	Function	Record Format
76-79 (Cont'd)			DC F'0' DC Y(BLKSIZE) DC Y(BLKSIZE -1)	DEBLOCKER3 (Bytes 76-77 only) (Bytes 78-79 only)	V,S U U
80-83	50-53		DC Y(BLKSIZE) +Y(BLKSIZE-1) DC &(BLKSIZE) +Y(BLKSIZE+1) DC F'0' LR 12,RECSIZE DC H'0'	Forward Backward DEBLOCKER4 (Bytes 80-81 only) (Bytes 82-83 only)	F F V,S U U
84-87	54-57		DC Y(RECSIZE -1) DC 2X'00' DC A(IOAREA1 +4) DC A(IOAREA2 +4) DC A(IOAREA1 +BUFOFF) DC A(IOAREA2 +BUFOFF) DC 2X'00' B 28(15) B 24(15) B 28(15) DC A(ERROPT)	(Bytes 84-85) (Bytes 86-87) Output, Standard labels 1 I/O area: DEBLOCKER 5, EBCDIC 2 I/O area's: DEBLOCKER 5, EBCDIC 1 I/O area: DEBLOCKER5, ASCII 2 I/O area's: DEBLOCKER5, ASCII (Bytes 84-85 output only) Standard labels; reserved for OPEN Input only, ERROPT= omitted Input only, ERROPT= SKIP Input only, ERROPT= IGNORE Input only, ERROPT= ADDRESS	F F V,S V,S V V F,U U U U U
88-91	58-5B		DC A(WLRERR) B 24(15) B 28(15) DC 2X'00' DC A(ERROPT)	Input only, WLRERR= ADDRESS Input only, WLRERR omitted and ERROPT= SKIP Input only, WLRERR omitted and ERROPT= IGNORE or omitted Output only, standard labels (bytes 88-89), reserved for OPEN Input only, WLRERR omitted and ERROPT= ADDRESS	For fixed-length records only
90-95	5A-5F		DC 6X'00'	File serial number, Standard labels, Output only	
92-95	5C-5F		DC A(ERROPT) B 28(15) B 24(15) B 28(15)	Input only, ERROPT= ADDRESS Output, nonstand-rd labels only. ERROPT= ADDRESS Input only, ERROPT= omitted Input only, ERROPT= SKIP Input only, ERROPT= IGNORE	
96-99	60-63		DC 4X'00'	Volume sequence number, Standard labels, output only	
96-97	60-61		DC 2X'00'	Standard labels, input only, reserved for OPEN	

Bytes Dec	Bytes Hex	Bits	Contents*	Function	Record* Format
98-103	62-67		DC 6X'00'	File serial number, Standard labels, Input only.	For fixed-length records only
100-103	64-67		DC 4X'00'	File sequence number, Standard labels, output only	
104-107	68-6B		DC 4X'00'	Volume sequence number, Standard labels, input only	
			DC A(ERROPT)	Output only, Standard labels only. ERROPT= ADDRESS	
108-111	6C-6F		DC 4X'00'	File sequence number, Standard label, input only	
88-91	58-5B		DC F'0'	DEBLOCKER6	
92-95	5C-5F		DC A(ERROPT)	Output only, Nonstandard labels only. ERROPT= ADDRESS	
92-93	5C-5D		DC Y(BLKSIZE)	Input only	
			DC Y(BLKSIZE -4)	Output only: EBCDIC	
			DC Y(BLKSIZE -BUFOFF)	Output only: ASCII	(V only)
94-95	5E-5F		DC Y(BLKSIZE -1)		
96-97	60-61		DC Y(RECSIZE -1)		For variable-length and spanned records
98-99	62-63		DC H'0'	Input only: Residual count	
100-103	64-67		DC A(WLRERR)	Input only, WLRERR= ADDRESS	
			B 24(15)	Input only, WLRERR= omitted and ERROPT= SKIP	
			B 32(15)	Input only, WLRERR= omitted and ERROPT= IGNORE or omitted	
100-101	64-65		DC 2X'00'	Output only, Standard labels, reserved for OPEN	
104-107	68-6B		DC A(ERROPT)	Input only, ERROPT= ADDRESS	
			B 28(15)	Input only, ERROPT= omitted	
			B 24(15)	Input only, ERROPT= SKIP	
			B 28(15)	Input only, ERROPT= IGNORE	
			DC A(ERROPT)	Output, Nonstandard labels only (version 3 onward); ERROPT=ADDRESS	
108-111	6C-6F		DC 4X'00'	Volume sequence number; Standard labels, output only	
108-109	6C-6D		DC 2X'00'	Standard labels, input only; Reserved for OPEN	
110-115	6E-73		DC 6X'00'	File serial number; Standard labels, input only	
112-115	70-73		DC 4X'00'	File sequence number; Standard labels, output only	

Bytes Dec	Bytes Hex	Bits	Contents*	Function	Record Format
116-119	74-77		DC A(ERROPT)	Output only, ERROPT= ADDRESS, Standard labels only	For variable length and spanned records
116-119	74-77		DC 4X'00'	Volume sequence number; Standard labels, input only	
120-123	78-7B		DC 4X'00'	File sequence number; Standard labels, input only	
86-91	56-5B		DC 6X'00'	File serial number; Standard labels, output only	For undefined records only
88-91	58-5B		DC A(WLRERR)	Input only; WLRERR= ADDRESS	
			B 24(15)	Input only; WLRERR= omitted and ERROPT= SKIP	
			B 28(15)	Input only; WLRERR= omitted and ERROPT= IGNORE or omitted	
			DC A(ERROPT)	Input only; WLRERR= omitted and ERROPT= ADDRESS. Output only; nonstandard labels, ERROPT= ADDRESS	
92-95	5C-5F		DC 4X'00'	Volume sequence number; Standard labels, output only	
92-93	5C-5D		DC 2X'00'	Standard labels; input only, Reserved for OPEN	
94-99	5E-63		DC 6X'00'	File serial number; Standard labels, input only	
96-99	60-63		DC 4X'00'	File sequence number; Standard labels, output only	
100-103	64-67		DC 4X'00'	Volume sequence number; Standard labels, input only	
100-103	64-67		DC A(ERROPT)	Output only; Standard labels only, ERROPT= ADDRESS	
100-103	64-67		DC 4X'00'	Volume sequence number; Standard labels, output only	
104-107	68-6B		DC 4X'00'	File sequence number; Standard labels, input only	
100-103	64-67		DC A(WLRERR)	Input only; WLRERR= ADDRESS	For spanned records only
			B 24(15)	Input only; WLRERR= omitted and ERROPT= SKIP	
			B 32(15)	Input only; WLRERR= omitted and ERROPT= IGNORE or omitted	
100-101	64-65		DC 2X'00'	Output only; Standard labels, reserved for OPEN	
102-107	66-6B		File serial number	Standard labels, output only	
100-103	64-67		DC 4X'00'	Output only; ERROPT= ADDRESS Nonstandard labels only	

Bytes Dec	Bytes Hex	Bits	Contents*	Function	Record Format
100-123	64-7B		DC 24X'00'	Output only; ERROPT= omitted, nonstandard labels	
104-107	68-6B		DC A(ERROPT) B 24(15) B 24(15) B 28(15)	Input only; ERROPT= ADDRESS Input only; ERROPT= omitted Input only; ERROPT= SKIP Input only; ERROPT= IGNORE	
104-107	68-6B		DC A(ERROPT)	Output only; ERROPT= ADDRESS, nonstandard labels	
108-123	6C-7B		DC 16X'00'	Output only; ERROPT= ADDRESS, nonstandard labels	
108-111	6C-6F		Volume sequence number	Standard labels, output only	
112-115	70-73		File sequence number	Standard labels, output only	
116-119	74-77		DC A(ERROPT)	Output only, ERROPT= ADDRESS, standard labels	
120-123	78-7B		DC 4X'00'	Output only, ERROPT= ADDRESS, standard labels	For spanned records only
108-123	6C-7B		DC 16X'00'	Input only, nonstandard labels	
108-109	6C-6D		DC 2X'00'	Standard labels, input only, reserved for OPEN	
110-115	6E-73		File serial number	Standard labels, input only	
116-119	74-77		Volume sequence number	Standard labels, input only	
120-123	78-7B		File sequence number	Standard labels, input only	
124-127	7C-7F		DC F'0'	Full word for loading and storing USER RECSIZE: IJFVSREC	
128	80		DC X'00'	IJFVSFLG	
		0		Sign bit, not used	
		1		Skip to first segment	
		2		First segment	
		3		Segment out of sequence, input only	
		4		Read back for EOV, output only	
		5		File reversed for logical spacing; Input CNTRL only; Trunc issued, output only	
		6		User Trunc issued, output only	
		7		Multi segment, output only; Skip Get segment, input CNTRL only	
129-131	81-83		DC 3X'00'	Pointer within WORKA	

See Notes on next page

* The format of the tape data file DTF is different starting at byte 44.
The location indicated by the numbers in the left hand column can contain only one of the factors listed under Contents .
The factor used for any given DTF table is determined by whether the file record format is fixed, variable or undefined, and by other DTF parameters as indicated.
A blank in the record column indicates that the contents apply to all record types.

- Record Format explanation

  F = Fixed Record
  V = Variable
  U = Undefined
  S = Spanned (variable format superset)

  The deblockers are scratch areas used by the modules to save data from one GET/PUT macro instruction to another. In the text and listings, they are referred to by the names DEBLOCKER1 to 6. These are not labels; they are comments used to make it easier to follow the listings.

Bytes		Bits	Function
Dec	Hex		
0 -15	00-0F		CCB
16	10	0-1	Not used
		2	COBOL open; ignore option
		3	1= VOL1 label is at user specified density
		4	1= DTF table address constants relocated by OPENR
		5-7	Not used
17-19	11-13		Address of logic module
20	14		DTF type= X'10'
21	15	0	1= No rewind
		1	1= Rewind unload
		2	1= Workfile
		3	1= Read backward
		4	1= Write
		5	1= POINTW
		6	Not used
		7	1= Forward-space file before next operation
22-23	16-17		Not used
24-25	18-19		Record length
26-27	1A-1B		Maximum BLKSIZE
28	1C		Read command code (X'02' for read forward; X'0C' for read backward)
29-31	1D-1F		EOF address
32-39	20-27		CCW
40-43	28-2B		Block count, initialized 00000000 for read forward, 00400000 for read backward
44	2C	0	1= Error routine
		1	1= Ignore
		2	Not used
		3	1= Record fixed unblocked
		4-7	Not used
45-47	2D-2F		DC A(ERROPT) Address of error routine

DTFSD (Data Files)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0 -15	00-0F		Command Control Block (CCB)
	16	10	0	1= Dequeue old volume extents
			1	1= Dummy OPEN to obtain extents from label track
			2	1= File assigned 'IGN' (COBOL)
			3	1= Track hold option specified
			4	1= DTF relocated by OPENR
			5	1= Input trailer labels to be processed at close time (COBOL only)
			6	1= Spanned processing
			7	1= COBOL end-of-extent option specified
	17-19	11-13		Address of logic module
	20	14		DTF type for OPEN/CLOSE (X'20'= sequential access DASD files)
	21	15	0	1= Not used
			1	1= Blocked file
			2	1= Work file
			3	1= Workarea specified
			4	1= Not a Version 1 type table
			5	1= Open; 0= closed
			6	1= Input; 0= output
			7	1= User labels specified
	22-28	16-1C		Filename (DTF name)
	29	1D		Device type code: X'00'= 2311 X'01'= 2314,2319 X'04'= 3330-1,-2 X'05'= 3330-11 X'07'= 3350 X'08'= 3340 general X'09'= 3340 35MB X'0A'= 3340 70MB Note: In previous versions, last byte of filename contains device type code
	30-35	1E-23		Address of Format 1 label in VTOC (BCCHHR)
	36-37	24-25		Volume sequence number
	38	26		Open communication byte: Input file
			0	1= No more extents
			1	1= Update file
			2	1= Process trailer labels
			3	1= Exit to user's EOF routine
			4	1= Next extent on new volume
			5	1= Return to close routine
			6	1= Process header labels
			7	1= Extent switch
				Output file
			0	1= No more extents
			1	1= Extents needed at close time
			2	1= Process trailer labels
			3	1= Process header labels
			4	1= Next extent on new volume
			5	1= Extents entered via console

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	38 (Cont'd)			Output file (Cont'd)
			6	1= Process trailer labels at close
			7	1= Check extent for minimum of 2 tracks
	39	27	0	1= Extent bypassed before file is opened (input only)
			1	1= FEOVD has been issued (input only)
			0-7	Sequence number of current extent opened (Output only)
	40	28		Sequence number of last extent opened
	41-43	29-2B		Address of user's label routine
	44	2C	0	not used
			1	1= Device supports RPS
			2	1= Version 3 DTF
			3-6	not used
			7	1= DTF has been extended into the partition virtual area
	45-47	2D-2F		Address of IOAREA1
	48-51	30-33		CCHH address of user's label track (X'80000000')
&Filename.S	52-53	34-35		Lower head limit (HH)
	54-57	36-39		Extent upper limit (CCHH)
	58-59	3A-3B		Seek address (BB): X'0000' if a disk device
	60-63	3C-3F		Search argument (CCHH)
	64	40		Record number
	65-67	41-43		EOF address if input file ; Key length and data length if output file
	68-71	44-47		CCHH control field :
				CCHH= X'00C80009' if 2311 - type 1
				CCHH= X'00C80013' if 2314 or 2319 - type 1
				CCHH= X'01940012' if 3330 - type 1
				CCHH= X'03280012' if 3330-11 - type 1
				CCHH= X'015C000B' if 3340 35MB
				CCHH=X'02B8000B' if 3340 70MB
				CCHH=X'022B001D' if 3350 - type 1
				where nn= current upper head number
	72	48		Number of records per track (input) or number of records per track - minus one (output)
	73	49		Switch byte used by the logic modules for various switching purposes. Functions indicated are for the ON condition(1) of the respective bit.
				Fixed length Record Modules
			0	Not first entry after Open (INPUT and UPDATE) Not first write after Open (OUTPUT)
			1	Short record (INPUT and UPDATE without truncation)
			2	Partial block written (OUTPUT)
			3	ERROPT= SKIP (INPUT); TRUNC= YES (OUTPUT)
			4	End-of-file record written (OUTPUT) End of extent (UPDATE)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	73 (Cont'd)			Fixed length Record Modules (Cont'd)
			5	Truncation not specified (used by OPEN routines)
			6	Write block of records (UPDATE)
			7	End of file (UPDATE)
				Variable length Record Modules
			0	Not first entry after OPEN (INPUT and UPDATE) Write record (OUTPUT)
			1	Wrong length record (INPUT); TRUNC= YES (OUTPUT); Second GET operation performed (UPDATE)
			2	Return to close routine (OUTPUT) Update specified (UPDATE)
			3	Not first entry after OPEN (OUTPUT)
			4	New extent required by CLOSE
			5	Capacity of I/O area exceeded (OUTPUT) Second GET required (UPDATE)
			6	Not first read (INPUT) Second GET issued (UPDATE)
			7	Unnecessary to read (INPUT) Track capacity exceeded (OUTPUT) Save record count (UPDATE)
				Undefined length Record Modules
			0	Not first entry after OPEN (ALL modules)
			1	Save record count (UPDATE)
			2	Return to close routine (OUTPUT)
			3	Second GET issued (UPDATE)
			4	Not used
			5	PUT command issued (UPDATE)
			6	End of file reached (UPDATE)
			7	Multi-track operation (UPDATE)
	74-75	4A-4B		Block size minus 1
	76-80	4C-50		CCHHR= Extent lower limit and record number. Field is used as a search argument bucket by the logic modules
	81	51	1	1= FEOVD has been issued (output only)
	81-83	51-53		Address of user wrong-length record routine if input file; Track capacity counter if output file
	84-87	54-57		Instruction to load user's register IOREG. (Note: This field is a NOP unless blocked records are processed in one I/O area, or two I/O areas are specified and records are processed in the I/O areas)
	88-91	58-5B		Address of current available input/output area
	92-95	5C-5F		Logical record size
	96-99	60-63		Address of end of input/output area

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	100	64		Logical indicators:
			0	1= ERROPT= address
			1	1= ERROPT= IGNORE
			2	1= ERROPT= SKIP
			3	1= VERIFY= YES
			4	1= 2 I/O areas
			5	1= WLRERR= address (fixed length and variable records) 1= Output file (undefined length records)
			6	1= Fixed-length records 0= Variable or undefined length records
			7	Control parameter specified
	101-103	65-67		Address of user's read error routine
	104-111	68-6F		Seek CCW
	112-119	70-77		Search ID Equal CCW
	120-127	78-7F		TIC CCW
	128-135	80-87		Read/Write Data CCW

This is the end of the common portion of the DTFSD table. The following sections are a added depending on the parameters specified in the operand of the DTFSD macro instruction.

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function	Record Format
If RECFORM= FIXBLK and TRUNCS= YES					Fixed length record input files
	136-143	88-8F		Read count CCW	
	144-151	90-97		Count field input area	
	If CONTROL= YES, the following section is added				
	152-167	98-A7		Control CCB	
	168-175	A8-AF		Control CCW	
If UPDATE= YES					
	136-143	88-8F		Search ID Equal CCW	
	144-151	90-97		TIC CCW	
	152-159	98-9F		Verify CCW	
	If CONTROL= YES, the following section is added				
	160-175	A0-AF		Control CCB	
	176-183	B0-B7		Control CCW	
If RECFORM= FIXBLK, TRUNCS= YES and UPDATE= YES					
	136-143	88-8F		Read Count CCW	
	144-151	90-97		Search ID Equal CCW	

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function	Record Format
If RECFORM=FIXBLK, TRUNCS=YES and UPDATE=YES (Cont'd)					Fixed record length input files
	152-159	98-9F		TIC CCW (Bytes 158-159 contain saved block length if two files are using same logic module)	
	160-167	A0-A7		Verify CCW	
	168-175	A8-AF		Count field input area	
	If CONTROL=YES, the following section is added				
	176-191	B0-BF		Control CCB	
	192-199	C0-C7		Control CCW	
If TRUNCS or UPDATE are not specified, no additions are made to the DTFSD table except when CONTROL=YES is specified, the following section is added.					
	136-151	88-97		Control CCB	
	152-159	98-9F		Control CCW	
	136-143	88-8F		Search ID Equal CCW	Fixed record length output files
	144-151	90-97		TIC CCW	
	152-159	98-9F		Verify CCW	
	If CONTROL is not specified				
	160-163	A0-A3		End-of-extent routine address (primarily used by COBOL compiler)	
	If CONTROL=YES				
	160-175	A0-AF		Control CCB	
	176-183	B0-B7		Control CCW	
	184-187	B8-BB		End-of extent routine address (primarily used by COBOL compiler)	
	136-143	88-8F		Read count CCW	Variable length record, undefined length record, spanned record input files
If UPDATE is not specified:					
	144-151	90-97		Count field input area	
	If CONTROL=YES *				
	152-167	98-A7		Control CCB	
	168-175	A8-AF		Control CCW	
	176-179	B0-B3		Logical record length	
	180-183	B4-B7		RX type instruction	

* These bytes are always generated when spanned processing is specified

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function	Record Format
If UPDATE is not specified: (Cont'd)					Variable length record, undefined length record, spanned record input files
	If CONTROL= YES (Cont'd)				
	184	B8	0 1 2 3-7	Not used 1= Skip segment 1= Spanned first time Not used	
	185-187	B9-BB		Pointer in logical record	
If UPDATE= YES:					Variable length record and undefined length record input files
	144-151	90-97		Search ID Equal CCW	
	152-159	98-9F		TIC CCW	
	160-167	A0-A7		Verify CCW	
	168-175	A8-AF		Count field input area	
	176-183	B0-B7		Count field save area if one I/O area	
	184-191	B8-BF		Count field save area if two I/O areas	
	If CONTROL= YES *				
	192-207	C0-CF		Control CCB	
	208-215	D0-D7		Control CCW	
	216-219	D8-DB		Logical record length	Variable length spanned record update files
	220-223	DC-DF		RX type instruction	
	224	E0	0 1 2 3 4 5 6 7	Not used 1= Skip segment 1= Spanned first time 1= Null segment 1= Spanned PUT return Not used Not used 1= No update	
	225-227	E1-E3		Pointer in logical record	
	228-235	E4-EB		Count save area	
	236-239	EC-EF		Extent status save area	
	136-143	88-8F		Search ID Equal CCW	Variable length record output files
	144-151	90-97		TIC CCW	
	152-159	98-9F		Verify CCW	
	160-163	A0-A3		Space remaining in output area	
	164-165	A4-A5		Track capacity	
	166-169	A6-A9		Instruction to load user's register VARBLD (If VARBLD is not specified, instruction is NO-OP)	

* These bytes are always generated when spanned processing is specified

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function	Record Format
	If CONTROL= YES *				Variable length record output files
	170-172	AA-AC		Not used	
	173-175	AD-AF		End-of-extent routine address (primarily used by COBOL compiler)	
	176-191	B0-BF		Control CCB	
	192-199	C0-C7		Control CCW	
	200-203	C8-CB		Logical record length	Variable length spanned record output files
	204-207	CC-CF		RX type instruction	
	208	D0	0 1 2 3 4 5 6 7	Not used Not used 1= Leading segment 1= Output block truncated 1= End of track 1= Track truncated 1= Save count 1= Volume spanned	
	209-211	D1-D3		Pointer in logical record	
	212-219	D4-DB		Count save area	
	220-223	DC-DF		Extent status save area	
	136-143	88-8F		Search ID Equal CCW	Undefined length record output files
	144-151	90-97		TIC CCW	
	152-159	98-9F		Verify CCW	
	160-161	A0-A1		Track capacity	
	If CONTROL= YES				
	162-164	A2-A4		Not used	
	164-167	A4-A7		End-of-extent routine address (primarily used by COBOL compiler)	
	168-183	A8-B7		Control CCB	
	184-191	B8-BF		Control CCW	

* These bytes are always generated when spanned processing is specified

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0-15	00-0F		Command Control Block (CCB)
	16	10	0-1	Not used
			2	1= File assigned 'IGN' (COBOL)
			3	1= Track hold option specified
			4	1= DTF relocated by OPENR
			5-7	Not used
	17-19	11-13		Address of logic module
	20	14		DTF type for OPEN/CLOSE (X'20' = sequential access DASD files)
	21	15	0	0= Disk device
			1	1= CLOSE macro is not to delete Format 1 and Format 3 file labels
			2	1= Work file
			3	Type of open : 1= Point; 0= Normal
			4	1= Routine entered from close routine
			5	1= File opened; 0= File closed
			6	Not used
			7	1= Re-entry to close routine
	22-28	16-1C		Filename (DTF name)
	29	1D		Device type Code : X'00'= 2311 X'01' =2314, 2319 X'04'= 3330-1, 2 X'05'= 3330-11 X'07'= 3350 X'08'= 3340 general X'09'= 3340 35MB X'0A'= 3340 70MB Note : In previous versions, last byte of filename contains device type code
	30-31	1E-1F		Track capacity counter
	32-35	20-23		Address of Format 1 label in VTOC (CCHR)
	36	24		Extent sequence number
	37	25		Open communication byte
			0	Not used
			1	1= Device supports RPS
			2	1= Version 3 DTF
			3	1= symbolic unit in DTF
			4	1= next extent on new volume
			5	1= extent opened
			6	Not used
			7	1= DTF has been extended into the partition virtual area
	38	26		Lower head limit
	39	27		Upper head limit
&Filename.L	40-41	28-29		Record length
	42-45	2A-2D		Initial extent lower limit
	46-49	2E-31		Current extent lower limit
	50-53	32-35		Extent upper limit

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.S	54-55	36-37		Seek address (BB= X'0000')
	56-59	38-3B		Search address (CCHH)
	60	3C		Record number
	61	3D		Switch byte used by logic module
			0	1= First write entry indicator
			1	1= Write update indicator
			2	1= POINTS macro issued
			3	Not first record of a track (RECFORM= UNDEF)
			4	1= Track upper limit reached
			5	Not used
			6	1= Check after read/write
			7	Not used
	62-63	3E-3F		Maximum record length
	64	40		Verify chain bit
	65-67	41-43		Address of user's EOF routine
	68	44		Logical indicators
			0	1= ERROPT= address
			1	1= ERROPT= IGNORE
			2	1= Fixed-length unblocked records
			3	1= Verify specified
			4	1= ERROPT= SKIP
			5	1= Re-read after read error
			6-7	Not used
	69-71	45-47		Address of user read/write error routine
	72-143	48-8F		CCW chain for work files
	144-151	90-97		Input area for Verify CCW and Read Count CCW

DTF Assembly Label	Module DSECT Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	IJICCB	0-15	00-0F		Command Control Block (CCB)
	IJIMOD	16	10	0	1= Trailer labels
				1	Used by FREE macro
				2	1= COBOL Open/Ignore option
				3	1= Track hold option specified
				4	1= DTF relocated by OPENR
				5	Not used
				6	1= SPNUNB
				7	Used by CNTRL macro
		17-19	11-13		Address of logic module
		20	14		DTF type for OPEN/CLOSE (X'22'= direct access files)
	IJISWI	21	15	0	1= Output; 0= Input
				1	1= Verify option specified
				2	1= Search multiple track (SRCHM) specified
				3	1= WRITE AFTER or WRITE RZERO macro used
				4	1= IDLOC specified
				5	1= Undefined; 0 = FIXUNB, VARUNB or SPNUNB
				6	1= RELTYPE= DEC
				7	1= End of file
	IJIFNM	22-28	16-1C		Filename (DTF name)
	IJIDVTP	29	1D		Device type code: X'05'=3330-11 X'00'=2311 X'07'=3350 X'01'=2314,2319 X'08'=3340 general X'09'=3340 35MB X'04'=3330-1,-2 X'0A'=3340 70MB
	IJIUNT	30-31	1E-1F		Starting logical unit address of the first volume containing the data file. This value is supplied by the OPEN from EXTENT cards (can be initially zero)
	IJIRPS	32	20	0	Not used
				1	1= Device supports RPS
				2-6	Not used
				7	1= DTF has been extended into the partition virtual area
	IJIULB	33-35	21-23		Address of user's label routine
	IJIUXT	36-39	24-27		Address of user's routine for processing EXTENT information
	IJIRELPT	40	28		Pointer to relative address area: $\frac{\&Filename.P - \&Filename}{2}$
	IJIERC	41-43	29-2B		Address of a 2-byte field in which IOCS can store the error condition or status codes
	IJITST	44-45	2C-2D		Macro code switch for internal use: X'0000'= READ ID X'0001' =READ KEY X'0002'= WRITE ID X'0003'= WRITE KEY X'0004'= WRITE RZERO X'0005'= WRITE AFTER

DTF Assembly Label	Module DSECT Label	Bytes Dec	Bytes Hex	Bits	Function
	IJIBPT	46-47	2E-2F		Pointer to channel program build area (&Filename.B) minus 32
	IJICB2	48-63	30-3F		Control seek CCB
&Filename.Z	IJICCW	64-71	40-47		Control Seek CCW for overlap seek routine
	IJIXMD	72-75	48-4B		Channel program builder instruction: XI 36(2),C'0'
	IJIMSZ	76-77	4C-4D		Maximum data length for FIXUNB or UNDEF records; BLKSIZE for VARUNB or SPNUNB records
	IJISPT	78	4E		Pointer to READ ID string (File name.0); X'00' if no READ ID issued
		79	4F		Pointer to READ KEY string (File name.1); X'00' if no READ KEY issued
		80	50		Pointer to WRITE ID string (File name.2); X'00' if no WRITE ID issued
		81	51		Pointer to WRITE KEY string (File name.3); X'00' if no WRITE KEY issued
		82	52		Pointer to WRITE RZERO string (Filename.4); X'00' if no WRITE RZERO issued
		83	53		Pointer to WRITE AFTER string (Filename.5); X'00' if no WRITE AFTER issued
	IJITRK	84-85	54-55		Track constant: 2311: H'0' if key length=0 H'20' if key length≠0 2314/2319: H'0' if key length=0 H'45' if key length≠0 3330: H'135' if key length=0 H'191' if key length≠0 3340: H'167' if key length=0 H'242' if key length≠0 3350: H'185' if key length=0 H'267' if key length≠0
	IJIRIC	86-87	56-57		2311: H'61' 2314/2319: H'101' 3330: H'135' 3340: H'167' 3350: H'185'
	IJILAT	88	58	0 1 2 3 4	Not used 1= Wrong-length record 1= non data transfer error Not used 1= no room found

DTF Assembly Label	Module DSECT Label	Bytes Dec	Bytes Hex	Bits	Function
		88 (Cont'd)		5-6	Not used
				7	1= Record out of extent area
		89	59	0	1= Data check in count area
				1	1= Track overrun
				2	1= End of cylinder
				3	1= Data check when reading key or data
				4	1= No record found
				5	1= End of file
				6	1= End of volume
				7	Not used
	IJILBTK	90-95	5A-5F		Label track address, XBCCHH, where X is the volume sequence number of the device on which the label track is located.
The following section is included if UNDEF, AFTER or RZERO is specified					
&Filename.L	IJILST	96-143	60-8F		Basic CCW's to build channel program
		144-183	90-B7		Basic CCW's for undefined length or formatting macros
	IJIVIT	184-185	B8-B9		Instruction to give record length to user if record length is undefined (NOPR 0 if no RECSIZE specified)
	IJIFRU	186-187	BA-BB		Instruction to get record length from user if record length is undefined. (NOPR 0 if no RECSIZE specified)
&Filename.F	IJIFLD	188-192	BC-C0		Work area (used for R0 address - CCHH0)
&Filename.K	IJICNT	193-200	C1-C8		Work area (used for R0 data field)
&Filename.C	IJICTS	201-208	C9-D0		Work area (included only for spanned or variable records for record count field)
The channel program builder strings are generated following the DTFDA table and preceding the channel program building area					
&Filename.0		Variable			Channel program builder string for READ ID macro. If READ ID is not specified, the string is not generated
&Filename.1		Variable			Channel program builder string for READ KEY macro. If READ KEY is not specified, the string is not generated
&Filename.2		Variable			Channel program builder string for WRITE ID macro. If WRITE ID is not specified, the string is not generated

DTF Assembly Label	Module DSECT Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.3		Variable			Channel program builderstring for WRITE KEY macro. If WRITE KEY is not specified, the string is not generated
&Filename.4		Variable			Channel program builder string for WRITE RZERO macro. If WRITE RZERO or WRITE AFTER is not specified, the string is not generated
&Filename.5		Variable			Channel program builder string for WRITE AFTER macro. If WRITE RZERO or WRITE AFTER is not specified, the string is not generated
The following section contains the channel program build areas and varies in size					
&Filename.B		0-7	00-07		Seek CCW that is generated at program assembly time and used by all channel programs
		Variable			Area to build: 1) Eight CCW's if AFTER is not specified 2) Eight CCW's if spanned or variable length records and AFTER= YES is specified 3) Seven CCW's if undefined or fixed records and AFTER= YES is specified
		Variable			Area to build: 1) Eight CCW's if AFTER is not specified and VERIFY= YES is specified 2) Eight CCW's if spanned or variable length records and AFTER= YES and VERIFY= YES are specified 3) Five CCW's if undefined or fixed records and AFTER= YES and VERIFY= YES are specified
The following section is added for spanned records only					
		8 bytes			Count save area
		8 bytes			SEEKADR save area
		1 byte		0	1= Relative addressing
				1	1= IJIGET switch on
				2	1= Ignore hold switch on
				3	1= Reserved for use by DAMODV
				4	1= New volume SEEKADR
				5-7	Not used

DTF Assembly Label	Module DSECT Label	Bytes Dec	Bytes Hex	Bits	Function
		1 byte			Reserved
		2 bytes			Record size
		12 bytes			Work area
		8 bytes			Control word save area
The following section is added to the DTFDA table if DSKXTNT (relative addressing) is specified					
&Filename.P		3 bytes			3X'00' for padding
&Filename.I		5 bytes			IDLOC record area (bucket used by module)
&Filename.S		8 bytes			SEEKADR in the form: M,B1,B2,C1,C2,H1,H2,R
		4 bytes			DC A(&SEEKADR)
		4 bytes			DC A(&IDLOC)
		8 bytes			Work area for RELTYPE= DEC
&Filename.X		4 bytes			Save area for CCHH portion of actual DASD address
		4 bytes			Alteration factor for C1 in SEEK ADR (see bytes 112-119) 2311 : X'00000001' 2314/2319: X'00000001' 3330 : X'00001300' 3340 : X'00000C00' 3350 : X'00001E00'
		4 bytes			Alteration factor for C2 in SEEK ADR (see bytes 112-119) 2311 : X'0000000A' 2314/2319: X'00000014' 3330 : X'00000013' 3340 : X'0000000C' 3350 : X'0000001E'
		4 bytes			Alteration factor for H1 in SEEK ADR (see bytes 112-119) 2311 : X'00000001' 2314/2319: X'00000001' 3330 : X'00000001' 3340 : X'00000001' 3350 : X'00000001'
		Variable to end of DTF table			DSKXTNT table composed of a variable number of 8-byte entries containing extent information in the following format: Bytes 0-2 TTT2 - cumulative number of tracks in the DSK XTNT table entries up to and including the current entry 3 M- volume sequence number 4 B - bin number (0 for disk devices) 5-7 TTT1 - relative track number of lower limit of this entry A 2-byte end-of-table indicator containing X'FFFF' follows the last entry in the DSKXTNT table

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0 -15	00-0F		Command Control Block (CCB)
	16	10	0	Used by ISAM Interface Program
			1	Not used
			2	1= COBOL open; ignore option
			3	Not used
			4	1= DTF table address constants relocated by OPENR
			5	Not used
			6	1= Data set security
			7	1= Wrong blocksize error during file extension
	17-19	11-13		Address of logic module
	20	14		File type for OPEN/CLOSE (X'24'= LOAD)
	21	15		Option byte:
			0	Not used
			1	Not used
			2	1= Cylinder overflow option
			3	Not used
			4	1= Blocked records (used by previous versions)
			5	1= Verify
			6	Not used
			7	1= 2 I/O areas present
	22-28	16-1C		File name
	29	1D		Prime data device type indicator: X'00'= 2311 ; X'01'= 2314/2319 ; X'04'= 3330; X'08'= 3340 general; X'09'= 3340 35MB; X'0A'= 3340 70MB
&Filename.C	30	1E		Status byte:
			0	1= Uncorrectable DASD error (except WLR)
			1	1= WLR error
			2	1= Prime data area full
			3	1= Cylinder index area not large enough to reference prime data area. Set on only if error detected at SETFL time
			4	1= Master index not large enough to reference prime data area. Set on only if error detected at SETFL time
			5	1= Duplicate record
			6	1= Sequence error
			7	1= No EOF record written in prime data area
	31	1F		High level index device type indicator: X'00'= 2311 ; X'01'= 2314/2319 ; X'04'= 3330; X'08'= 3340 general ; X'09'= 3340 35MB ; X'0A'= 3340 70MB
	32	20		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	33-34	21-22		First prime track in cylinder (HH)
	35	23		First prime data record in cylinder (R)
	36-37	24-25		Last prime track in cylinder (HH)
	38	26		High record on master index/cylinder index track (R)
	39	27		High record on prime data track (R)
	40	28		High record on overflow track (R)
	41	29		High record on last track index track in cylinder (whether shared or unshared)
	42	2A		High record on track index track other than last in cylinder. If only one track index track in cylinder, it is equal to byte 41
	43	2B		Condition code :
			0	1= WLR checks requested (for extension)
			1	1= First record in file
			2	1= Prime data extent full
			3	1= Master index/cylinder index extent too small
			4	1= Prime data upper limit has been increased (for extension)
			5	1= Extension
			6-7	Not used
	44-50	2C-32		Prime data lower limit (MBBCCHH)
	51-57	33-39		Cylinder index lower limit (MBBCCHH)
	58-64	3A-40		Master index lower limit (MBBCCHH)
	65	41		Number of index levels
			0-3	Not used
			4	1= RPS type device (data)
			5	1= RPS type DTF
			6	1= Master index
			7	1= RPS type device (index)
&Filename.H	66-73	42-29		Address of last prime data record (MBBCCHHR)
	74-75	4A-4B		Logical record length
	76-77	4C-4D		Key length
	78-79	4E-4F		Block length (logical record length times number of records)
	80-81	50-51		Overflow record length (logical record length plus 10)
	82-83	52-53		Blocking factor (number of logical records)
	84-85	54-55		Index entry length (key length plus 10)
	86-87	56-57		Prime data record length (key length plus physical record length)
	88-89	58-59		Overflow record length with key (key length plus logical record length plus 10)
	90-91	5A-5B		Prime data record format length (key length plus physical record length plus 8)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	92-93	5C-5D		Overflow record format length (key length plus logical record length plus 18)
	94-95	5E-5F		Key location (in blocked records)
This is the end of the common DTF area. The format of the remainder of the table is variable and is generated according to the parameters specified in the DTFIS macro instruction				
&Filename.S	96-103	60-67		Seek/search address area (MBBCCHHR)
&Filename.P	104-105	68-69		Logical record counter (for blocking)
	106-107	6A-6B		Number of bytes for high level index
	108-111	6C-6F		Prime data record counter (logical records)
	112	70		Status indicators:
			0-1	Not used
			2	1= File closed
			3-5	Not used
			6	1= Last prime data track full
			7	1= Last block full
	113-117	71-75		Last track index normal entry address (CCHHR)
	118-122	76-7A		Last cylinder index entry address (CCHHR)
	123-127	7B-7F		Last master index entry address (CCHHR)
&Filename.B				CCW build area. See description of SETFL macro, phase 1 - $$BSETFL
	128-135	80-87		Seek CCW
	136-143	88-8F		Search ID Equal CCW
	144-151	90-97		TIC CCW
	152-159	98-9F		Read/Write CCW
	160-167	A0-A7		Search ID Equal CCW
	168-175	A8-AF		TIC CCW
	176-183	B0-B7		Verify CCW
&Filename.M	184-187	B8-BB		Address of IOREAL
	188-191	BC-BF		Address of data in WORKL. (FIXBLK= address of WORKL; FIXUNB= address of WORKL plus key).
	192-195	C0-C3		Address of key in WORKL. (FIXBLK= address of WORKL plus KEYLOC minus 1; FIXUNB= address of WORKL.)
	196-199	C4-C7		Block position indicator (address of logical record in IOAREAL)
	200	C8		Master index, extension indicator:
			0-2	Not used
			3	1= Extending file; 0= Creating file
			4-6	Not used
			7	1= Master index being used; 0= No master index being used

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	201-204	C9-CC		Cylinder index upper limit (CCHH)
	205-208	CD-D0		Master index upper limit (CCHH)
	209-215	D1-D7		Prime data upper limit (old upper limit if extension) (MBBCCHH)
	216-222	D8-DE		Prime data new upper limit (for extension) (MBBCCHH)
	223	DF		Last prime data track in cylinder minus 1
	224-225	E0-E1		Key length minus one
	226-227	E2-E3		Logical record length minus 1
	228-229	E4-E5		Address of track index dummy record (HR)
	230-231	E6-E7		Address of record before first prime data record in cylinder (HR)
	232	E8		Number of records on master index/cyl inder index track minus 1
	233-236	E9-EC		Master index/cyl inder index DASD address control field (CCHH): 2311 = X'00C70009' 2314/2319= X'00C70013' 3330 = X'01FF0012' 3340 = X'01FF000C'
	237-239	ED-EF		Prime data address control field (CCH): 2311 = X'00C700' 2314/2319= X'00C700' 3330 = X'01FF00' 3340 = X'01FF00'
	240-242	F0-F2		Prime data beginning of volume (CCH): 2311 = X'000100' 2314/2319= X'000100' 3330 = X'000100' 3340 = X'000100'
	243-245	F3-F5		Prime data end of volume (CCH): 2311 = X'00C700' 2314/2319= X'00C700' 3330 = X'019300' 3340 = X'015B00'(35MB), X'02B700'(70MB)
	246-247	F6-F7		Used for alignment
	248-251[1]	F8-FB		First entry in DSKXTN table (logical unit, cell number)
	256-259[2]	100-103		X'FFFFFFFF'= End of DSKXTN table
	260-263	104-107		Address of IOAREA2
	264-267	108-10B		Address used to relocate IOAREA2

1) Each entry in the DSKXTN table is 4 bytes long. The minimum number of entries is two. There is one entry per extent.
2) Location of the end-of-table indicator depends on length of DSKXTN table

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0 -15	C0-CF		CCB
	16	10	0	Not used
			1	Not used
			2	1= COBOL open; ignore option
			3	1= Track hold specified
			4	1= DTF table address constants relocated by OPENR
			5	Not used
			6	1= Data set security
			7	1= Wrong blocksize error during addition to file
	17-19	11-13		Logic module address
	20	14		File type for OPEN/CLOSE (X'25'= ADD)
	21	15		Option byte:
			0	Not used
			1	1= Prime data in core
			2	1= Cylinder overflow
			3	1= Cylinder index in core
			4	1= Blocked records
			5	1= Verify
			6-7	Not used
	22-28	16-1C		DTF file name
	29	1D		Prime data device type indicator: X'00'= 2311 X'01'= 2314/2319 X'04'=3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
&Filename.C	30	1E		Status byte:
			0	1= Uncorrectable DASD error (except WLR)
			1	1= WLR error
			2	1= EOF (sequential)
			3	1= No record found
			4	1= Illegal ID specified
			5	1= Duplicate record sensed
			6	1= Overflow area full
			7	1= Record retrieved from overflow area
	31	1F		Highest level index device type: X'00'= 2311 X'01'= 2314/2319 X'04'=3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
	32	20		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4
	33-35	21-23		First prime data record in cylinder (HHR)
	36-37	24-25		Last prime data track in cylinder (HH)
	38	26		High record number on master index/cylinder index track (R)

DTFIS (Add) - part 1 (....Cont'd)

DTF Assembly Label	Bytes		Bits	Function
	Dec	Hex		
	39	27		High record number on prime data track (R)
	40	28		High record number on overflow track (R)
	41	29		High record number on shared track (R)
	42	2A		High record number on track index (TI) track (R)
	43	2B		Retrieval byte :
			0	1= WORKR area specified
			1	1= WORKS area specified
			2	Overflow switch
			3	1= Read
			4	Not used
			5	1= Output
			6	1= Write key
			7	1= PUT macro issued
	44-50	2C-32		Prime data lower limit (MBBCCHH)
	51-57	33-39		Cylinder index lower limit (MBBCCHH)
	58-64	3A-40		Master index lower limit (MBBCCHH)
	65	41		Switches
			0	1= From WAITF routine
			1	1= WAITF seek check bit
			2-3	Not used
			4	1= RPS type device (data)
			5	1= RPS type DTF
			6	1= Master index
			7	1= RPS type device (index)
	66-73	42-49		Last prime data record address (MBBCCHHR)
	74-75	4A-4B		Logical record length (RECSIZE)
	76-77	4C-4D		Key length (KEYLEN)
	78-79	4E-4F		Block size (logical record length times number of records)
	80-81	50-51		Overflow record length (logical record length plus 10)
	82-83	52-53		Blocking factor (number of logical records in block (NRECDS)
	84-85	54-55		Index entry length (key length plus 10)
	86-87	56-57		Prime data record length (key length plus physical record length (block size)
	88-89	58-59		Overflow record length plus key (key length plus logical record length plus 10)
	90-91	5A-5B		Prime data record format length (key length plus blocksize plus 8)
	92-93	5C-5D		Overflow record format length (key length plus logical record length plus 18)
	94-95	5E-5F		Key location (KEYLOC) for blocked records
	96-97	60-61		Constant = 5
	98-99	62-63		Constant = 10

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	100-101	64-65		Displacement of part 2 of the DTFIS table from start of part 1
	102-103	66-67		Displacement of part 3 of the DTFIS table from start of part 1
&Filename.S	104-113	68-71		Seek/search address area (MBBCCHHRFP)
&Filename.W	114-123	72-7B		Random/sequential retrieval work area
&Filename.P	124-127	7C-7F		Prime data record count
	128	80		Status indicators:
			0-1	Not used
			2	1= File closed
			3-5	Not used
			6	1= Last prime data track full
			7	1= Block complete
	129-133	81-85		Last track index normal entry address (CCHHR)
	134-138	86-8A		Last cylinder index entry address (CCHHR)
	139-143	8B-8F		Last master index entry address (CCHHR)
	144-151	90-97		Last independent overflow record address (MBBCCHHR)
&Filename.I	152-153	98-99		Number of independent overflow tracks
&Filename.A	154-155	9A-9B		Number of full cylinder overflow areas
&Filename.O	156-157	9C-9D		Overflow record count
	158-164	9E-A4		Independent overflow area lower limit (MBBCCHH)
	165-171	A5-AB		Independent overflow area upper limit (MBBCCHH)
	172-175	AC-AF		A(&Filename.D) - Address of work area for cylinder overflow control record (COCR)
	176-179	B0-B3		A(&Filename.D+8) - Address of workarea for the current track index normal entry count field
	180-183	B4-B7		A(&Filename.D+16) - Address of work area for current track index overflow entry count field
	184-187	B8-BB		A(&Filename.D+24) - Address of workarea for current prime data record count field
	188-191	BC-BF		A(&Filename.D+32) - Address of work area for current overflow record count field
	192-195	C0-C3		A(&Filename.D+40) - Address of work area for track index normal entry data field
	196-199	C4-C7		A(&Filename.D+50) - Address of work area for current overflow record linkage field
	200-203	C8-CB		A(&IOREAL) - Address of IOREAL, the I/O area used for adding records to a file

DTFIS (Add) - part 1 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	204-207	CC-CF		A(&WORKL) - Address of WORKL, workarea containing user data records to be added to the file
	208-211	D0-D3		A(&Filename.K) - Address of the ADD key area
	212-215	D4-D7		A(&IOAREAL+8) - Address of key position in IOAREAL
	216-219	D8-DB		A(&IOAREAL+8+&KEYLEN) - Address of data position in IOAREAL

DTFIS (Add) - part 2

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.2	0 -3	00-03		A(&Filename.S+3) - Address of the seek/search address area plus 3
	4	04	0	1= Seek check indicated
			1-5	Not used
			6	1= Over/under seek has occurred
			7	1= An error has been found, but a seek check is indicated
	5 -7	05-07		A(&Filename.W) - Address of random/sequential retrieval work area
The following information is generated if the cylinder index in core option is specified				
	12-15	0C-0F		A(&INDAREA) -Starting address of main storage area specified for cylinder index
	16-17	10-11		AL2(&INDSIZE) - Number of bytes in main storage available for cylinder index
	18-25	12-19		Next cylinder index entry to be read (MBBCCHHR)
	26-30	1B-1E		Last cylinder index entry (CCHHR)
	31	1F		Core index byte:
			0	1= First time through B-transient, $$BINDEX
			1	1= End of cylinder index reached
			2	1= Index skip option specified
			3	1= Suppress in-core option and read cylinder index
			4-7	Not used
	32-35	1D-23		Pointer to key (stored by module)

DTFIS (Add) - part 2 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
The following information is generated if the prime data in core add function is specified This information is aligned on a double word boundary				
	36-37	24-25		Size of IOAREAL
	38-39	26-27		Maximum number of prime data records in main storage
	40-43	28-2B		Address of write CCW's
	44-47	2C-2F		Address of read CCW's
	48	30		Switch byte:
			0	1= EOF
			1-7	Not used
	49	31		Reserved.
IJHDCWRK	50-51	32-33		Work field for I/O Module.

DTFIS (Add) - part 3

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.B	0 -7	00-07		CCW X'07', &Filename.S+1, X'40', 6 - Long seek CCW with command chaining
	8 -127	08-7F		Channel program build area.
&Filename.D	128-135	80-87		Cylinder overflow control record (COCR)
	136-143	88-8F		Current track index normal entry count field address
	144-151	90-97		Current track index overflow entry count field address
	152-159	98-9F		Current prime data record count field address
	160-167	A0-A7		Current overflow record count field address
	168-177	A8-B1		Track index normal entry data field
	178-187	B2-BB		Current overflow record sequence link field
	188-197	BC-B5		Current track index overflow entry data field
	198	C6		X'01' - Add to EOF X'02' - Add to independent overflow area
	199-201	C7-C9		Overflow control bytes (CCH)
	202-203	CA-CB		High HR on overflow track
	204-211	CC-D3		Volume upper limit for prime data records (MBBCCHHR)
	212-217	D4-D9		CLC 0 (&KEYLEN,13),0 (6) - Unblocked CLC 0 (&KEYLEN,13), &KEYLOC-1 (6) - Blocked Utility CLC for key

DTF Assembly Label	Bytes		Bits	Function
	Dec	Hex		
	218-223	DA-DF		MVC 0 (&KEYLEN, 13),0 (12) - Unblocked MVC 0 (&KEYLEN, 13), &KEYLEN-1 (12) - Blocked Utility MVC for key
&Filename.E	224-227[1]	E0-E3		First entry in DSKXTN table (logical unit, cell number)
	232-235[2]	E8-EB		4X'FF' - End of DSKXTN table
&Filename.K	236 +	EC-end		Key area for ADD only. Number of bytes depends on key length, KEYLEN

1) Each entry in the DSKXTN table is four bytes long. The minimum number of entries is two. There is one entry per extent.

2) Location of the end-of-table indicator depends on length of DSKXTN table.

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0 -15	00-0F		Command Control Block (CCB)
	16	10	0	Not used
			1	1= GET issued
			2	1= COBOL open; ignore option
			3	1= HOLD option specified
			4	1= DTF table address constants relocated by OPENR
			5-6	Not used
			7	1= Different blocksize in format-1 label than in DTFIS.
	17-19	11-13		Address of logic module
	20	14		File type for OPEN/CLOSE (X'26'= RETRVE)
	21	15		Option byte:
			0	
			1	1= Prime data in core
			2	1= Cylinder overflow option
			3	1= Cylinder index in core option
			4	1= Blocked records
			5	1= Verify
			6-7	Not used
	22-28	16-1C		File name (DTF name)
	29	1D		Prime data device type: X'00'= 2311 X'01'= 2314/2319 X'04'= 3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
&Filename.C	30	1E		Status byte:
			0	1= Uncorrectable DASD error (except WLR error)
			1	1= WLR error
			2	1= EOF (sequential)
			3	1= No record found
			4	1= Illegal ID specified
			5	1= Duplicate record sensed
			6	1= Overflow area full
			7	1= Record retrieved from overflow area
	31	1F		High level index device type: X'00'= 2311 X'01'= 2314/2319 X'04'= 3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
	32	20		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4
	33-35	21-23		First prime data record in cylinder (HHR)
	36-37	24-25		Last prime data track in cylinder (HH)
	38	26		High record number on master index/cylinder index track (R)
	39	27		High record number on prime data track (R)

DTFIS (RETRVE, RANDOM) - part 1 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	40	28		High record number on overflow track (R)
	41	29		High record number on shared track (R)
	42	2A		High record number on track index track (R)
	43	2B		Retrieval byte :
			0	1= WORKR specified
			1	1= WORKS specified
			2	Overflow switch
			3	1= Read key
			4	Not used
			5	1= Output
			6	1= Write key
			7	1= PUT macro issued
	44-50	2C-32		Prime data lower limit (MBBCCHH)
	51-57	33-39		Cylinder index lower limit (MBBCCHH)
	58-64	3A-40		Master index lower limit (MBBCCHH)
	65	41		Switches
			0	1= From WAITF routine
			1	1= Seek check from WAITF
			2	1= Index track held
			3	1= Data track held
			4	1= RPS type device (data)
			5	1= RPS type DTF
			6	1= Master index
			7	1= RPS type device (index)
	66-73	42-49		Last prime data record address (MBBCCHHR)
	74-75	4A-4B		Logical record length
	76-77	4C-4D		Key length
	78-79	4E-4F		Block size (logical record length times number of records)
	80-81	50-51		Overflow record length (logical record length plus 10)
	82-83	52-53		Blocking factor
	84-85	54-55		Index entry length (key length plus 10)
	86-87	56-57		Prime data record length key (key length plus physical record length)
	88-89	58-59		Overflow record length with key (key length plus logical record length plus 10)
	90-91	5A-5B		Prime data record format length key (key length plus physical record length plus 8)
	92-93	5C-5D		Overflow record format length key (key length plus logical record length plus 18)
	94-95	5E-5F		Key location (blocked records)

DTFIS (RETRVE,RANDOM) - part 1 (....Cont'd)

DTF Assembly Label	Bytes		Bits	Function
	Dec	Hex		
	96-97	60-61		Constant= 5
	98-99	62-63		Constant= 10
	100-101	64-65		Displacement of part 2 of the DTFIS table from part 1
	102-103	66-67		Displacement of part 3 of the DTFIS table from part 1
&Filename.S	104-113	68-71		Seek/search address area (MBBCCHHRFP)
&Filename.W	114-123	72-7B		Random/sequential retrieval work area

DTFIS (RETRVE,RANDOM) - part 2

DTF Assembly Label	Bytes		Bits	Function
	Dec	Hex		
&Filename.2	0 -3	00-03		Address of seek/search address area plus 3
	4	04	0	1= Seek check indicated
			1-5	Not used
			6	1= Over/under seek has occurred
			7	1= An error has been found, but a seek check is indicated
	5 -7	05-07		Address of random/sequential retrieval work area
	8 -11	08-0B		Address of IOAREAS
	12-15	0C-0F		Address of IOAREAR
	16-19	10-13		Address of KEYARG
	20-23	14-17		Address of WORKR
	24-27	18-1B		Current sequential I/O area address
	28-31	1C-1F		4-byte NO-OP instruction, or L IOREG, * - 4 if IOREG specified.
	32	20		X'00'= No verify; X'40'= Verify
	33	21		X'08'= Unblocked; X'00'= Blocked
	34	22		R= First prime data record on shared track
	35-39	23-27		Upper limit for sequential retrieval (CCHHR)
	40-41	28-29		H'0'= Blocked records H'2'= Overflow record H'8'= Unblocked records
	42	2A		X'C7'= 2311, 2314 or 2319 X'FF'= 3330, 3340
	43-47	2B-2F		Initial values for sequential retrieval
&Filename.H	48-55	30-37		Current DASD address for sequential (MBBCCHHR)
	56-63	38-3F		Current overflow DASD address for sequential (MBBCCHHR)

DTFIS (RETRVE,RANDOM) - part 2 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	64-65	40-41		Sequential record counter
	66-67	42-43		Current track index entry for sequential (HR)
&Filename.T	68-69	44-45		Number of records tagged for deletion
	70-71	46-47		Load IOREG for random retrieval
&Filename.G	72-79	48-4F		DASD address save area (MBBCCHHR)
	80-83	50-53		Record pointer within I/O area for write operation
&Filename.R	84-87	54-57		Nonfirst overflow record count
The following information is generated when the cylinder index in core option is specified				
	92-95	5C-5F		A(&INDAREA) - Starting address of main storage area specified for cylinder index
	96-97	60-61		AL2(&INDSIZE) - Number of bytes in main storage available for cylinder index
	98-105	62-69		Next cylinder index entry to be read (MBBCCHHR) (Initialized by $$BINDEX to cylinder index starting address)
	106-110	6A-6E		Last cylinder index entry
	111	6F		Core index byte:
			0	1= First time through transient
			1	1= End of index reached
			2	1= Index skip option
			3-7	Not used
	112-115	70-73		Pointer to key (stored by the module)
	116-131	74-83		Reserved

DTFIS (RETRVE,RANDOM) - part 3

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.B	0 -7	00-07		X'07', &Filename.S+1, X'40', 6 - Long seek CCW with command chaining
	8 -63	08-3F		Area to build CCW-string
	64-67[1]	40-43		First entry in DSKXTN table (logical unit, cell number)
	72-75[2]	48-4B		4X'FF' End of DSKXTN table

1) The length of one entry is the four bytes shown here. The minimum number of entries is 2. There is one entry per extent.

2) The location of the end-of-table indicator depends on the length of DSKXTN table.

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0 -15	00-0F		Command Control Block (CCB)
	16	10	0	Used by ISAM Interface Program
			1	1= GET issued
			2	1= COBOL open; ignore option
			3	1= Track Hold specified
			4	1= DTF table address constants relocated by OPENR
			5	1= EOF on sequential retrieve
			6	1= Data set security
			7	1= Different blocksize in format 1 label than in DTFIS
	17-19	11-13		Address of logic module
	20	14		File type for OPEN/CLOSE (X'26'= RETRVE)
	21	15		Option byte:
			0	Not used
			1	1= Prime data in core
			2	1= Cylinder overflow option
			3	1= Cylinder index in core option
			4	1= Blocked records
			5	1= Verify
			6	1= IOAREAS just used; 0= IOAREA2 just used
			7	1= 2 I/O areas present
	22-28	16-1C		File name (DTF name)
	29	1D		Prime data device type: X'00'= 2311 X'01'= 2314/2319 X'04'= 3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
&Filename.C	30	1E		Status byte:
			0	1= Uncorrectable DASD error (except WLR error)
			1	1= WLR error
			2	1= EOF (sequential)
			3	1= No record found
			4	1= Illegal ID specified
			5	1= Duplicate record sensed
			6	1= Overflow area full
			7	1= Record retrieved from overflow area
	31	1F		High level index device type: X'00'= 2311 X'01'= 2314/2319 X'04'= 3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
	32	20		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4)
	33-35	21-23		First prime data record in cylinder (HHR)
	36-37	24-25		Last prime data track in cylinder (HH)

DTFIS (RETRVE, SEQNTL) - part 1 (...Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	38	26		High record number on master index/cylinder index track (R)
	39	27		High record number on prime data track (R)
	40	28		High record number on overflow track (R)
	41	29		High record number on shared track (R)
	42	2A		High record number on track index track (R)
	43	2B		Retrieval byte :
			0	1= WORKR specified
			1	1= WORKS specified
			2	Overflow switch
			3	1= Read key
			4	1= First record being processed (after issuing SETL macro)
			5	1= Output
			6	1= Write key
			7	1= PUT macro issued
	44-50	2C-32		Prime data lower limit (MBBCCHH)
	51-57	33-39		Cylinder index lower limit (MBBCCHH)
	58-64	3A-40		Master index lower limit (MBBCCHH)
	65	41		Index level number, WAITF indicator :
			0	1= From WAITF routine
			1	1= WAITF seek check bit
			2-3	Not used
			4	1= RPS type device (data)
			5	1= RPS type DTF
			6	1= Master index
			7	1= RPS type device (index)
	66-73	42-49		Last prime data record address (MBBCCHHR)
	74-75	4A-4B		Logical record length
	76-77	4C-4D		Key length
	78-79	4E-4F		Block size (logical record length times number of records)
	80-81	50-51		Overflow record length (logical record length plus 10)
	82-83	52-53		Blocking factor
	84-85	54-55		Index entry length (key length plus 10)
	86-87	56-57		Prime data record length (key length plus physical record length
	88-89	58-59		Overflow record length with key (key length plus logical record length plus 10)
	90-91	5A-5B		Prime data record format length (key length plus physical record length plus 8)
	92-93	5C-5D		Overflow record format length (key length plus logical record length plus 18)
	94-95	5E-5F		Key location (blocked records)

DTFIS (RETRVE,SEQNTL) - part 1 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	96-97	60-61		Constant= 5
	98-99	62-63		Constant= 10
	100-101	64-65		Displacement of part 2 of the DTFIS table from part 1
	102-103	66-67		Displacement of part 3 of the DTFIS table from part 1
&Filename.S	104-113	68-71		Seek/search address area (MBBCCHHRFP)
&Filename.W	114-123	72-7B		Random/sequential retrieval work area

DTFIS (RETRVE,SEQNTL) - part 2

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.2	0 -3	00-03		Address of seek/search address area plus 3
	4	04	0 1-5 6 7	1= Seek check indicated Not used 1= Over/under seek has occurred 1= An error has been found, but a seek check is indicated
	5 -7	05-07		Address of random/sequential retrieval work area
	8 -11	08-0B		Address of IOAREAS
	12-15	0C-0F		Address of IOAREA2
	16-19	10-13		Address of KEYARG
	20-23	14-17		Address of WORKR
	24-27	18-1B		Current sequential I/O area address
	28-31	1C-1F		L IOREG,*-4 - Load IOREG or a 4 byte NO-OP instruction
	32	20		X'00'= No verify; X'40'= Verify
	33	21		X'08'= Unblocked records; X'00'= Blocked rec'ds
	34	22		R= First prime data record on shared track
	35-39	23-27		Upper limit for sequential retrieval (CCHHR)
	40-41	28-29		H'0'= Blocked records H'2'= Overflow record H'8'= Unblocked records
	42	2A		X'C7'= 2311,2314 or 2319; X'09'= 2321; X'FF'= 3330, 3340
	43-47	2B-2F		Initial values for sequential (CCHHR)
&Filename.H	48-55	30-37		Current DASD address for sequential retrieval (MBBCCHHR)
	56-63	38-3F		Current overflow DASD address (MBBCCHHR)

DTFIS (RETRVE, SEQNTL) - part 2 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	64-65	40-41		Sequential record counter
	66-67	42-43		Current track index entry (HR)
&Filename.T	68-69	44-45		Number of records tagged for deletion.
	70-75	46-4B		For boundary alignment.
	76-91	4C-5E		Reserved.

DTFIS (RETRVE, SEQNTL) - part 3

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.B	0 -7	00-07		X'07', &Filename.S+1, X'40',6 - Long seek CCW with command chaining
	8 -63	08-3F		Area to build CCW-string
&Filename.E	64-67[1]	40-43		First entry in DSKXTN table (logical unit, cell number)
	72-75[2]	48-4B		4X'FF' - End of DSKXTN table

1) The length of one entry is the four bytes shown here. The minimum number of entries is 2. There is one entry per extent.

2) The location of the end-of-table indicator depends on the length of DSKXTN table.

DTFIS (ADDRTR) - part 1

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename	0 -15	00-0F		Command Control Block (CCB)
	16	10	0	Used by ISAM Interface Program
			1	1= GET issued
			2	COBOL open; ignore option
			3	1= Track hold option specified
			4	1= DTF table address constants relocated by OPENR
			5	EOF switch
			6	1= Data set security
			7	1= Wrong blocksize error during addition to file
	17-19	11-13		Logic module address
	20	14		File type for OPEN/CLOSE (X'27'= ADDRTR)
	21	15		Option byte:
			0	Not used
			1	1= Prime data in core
			2	1= Cylinder overflow
			3	1= Cylinder index in core
			4	1= Blocked records

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	21 (Cont'd)		5	1= Verify
			6	1= IOAREAS just used; 0= IOAREA2 just used
			7	1= 2 I/O areas present
	22-28	16-1C		DTF file name
	29	1D		Prime data device type indicator: X'00'= 2311 X'01'= 2314/2319 X'04'= 3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
&filename.C	30	1E		Status byte:
			0	1= Uncorrectable DASD error (except WLR error)
			1	1= WLR error
			2	1= EOF (sequential)
			3	1= No record found
			4	1= Illegal ID specified
			5	1= Duplicate record sensed
			6	1= Overflow area full
			7	1= Record retrieved from overflow area
	31	1F		Highest level index device type: X'00'= 2311 X'01'= 2314/2319 X'04'= 3330 X'08'= 3340 general X'09'= 3340 (35MB) X'0A'= 3340 (70MB)
	32	20		Relative position of the DSKXTN (logical unit, cell number) table (in words). This value is the length of the DTF table divided by 4.
	33-35	21-23		First prime data record in cylinder (HHR)
	36-37	24-25		Last prime data track in cylinder (HH)
	38	26		High record number on master index/cylinder index track (R)
	39	27		High record number on prime data track (R)
	40	28		High record number on overflow track (R)
	41	29		High record number on shared track (R)
	42	2A		High record number on track index (TI) track (R)
	43	2B		Retrieval byte:
			0	1= WORKR area specified
			1	1= WORKS area specified
			2	Overflow switch
			3	1= Read
			4	1= First record being processed (after issuing SETL macro)
			5	1= Output
			6	1= Write key
			7	1= PUT macro issued
	44-50	2C-32		Prime data lower limit (MBBCCHH)

DTFIS (ADDRTR) - part 1 (....Cont'd)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	51-57	33-39		Cylinder index lower limit (MBBCCHH)
	58-64	3A-40		Master index lower limit (MBBCCHH)
	65	41		Index level number, WAITF and track hold indicators :
			0	1= From WAITF routine
			1	1= Seek check from WAITF
			2	1= Index track held
			3	1= Data track held
			4	1= RPS type device (data)
			5	1= RPS type DTF
			6	1= Master index;
			7	1= RPS type device (index)
&Filename.H	66-73	42-49		Last prime data record address (MBBCCHHR)
	74-75	4A-4B		Logical record length (RECSIZE)
	76-77	4C-4D		Key length (KEYLEN)
	78-79	4E-4F		Block size (logical record length times number of records)
	80-81	50-51		Overflow record length (logical record length plus 10)
	82-83	52-53		Blocking factor (number of logical records in block (NRECDS))
	84-85	54-55		Index entry length (key length plus 10)
	86-87	56-57		Prime data record length (key length plus physical record length (block size)
	88-89	58-59		Overflow record length with key (key length plus logical record length plus 10)
	90-91	5A-5B		Prime data record format length (key length plus block size plus 8)
	92-93	5C-5D		Overflow record format length (key length plus logical record length plus 18)
	94-95	5E-5F		Key location (KEYLOC) for blocked records
	96-97	60-61		Constant = 5
	98-99	62-63		Constant = 10
	100-101	64-65		Displacement of part 2 of the DTFIS table from start of part 1
	102-103	66-67		Displacement of part 3 of the DTFIS table from start of part 1
&Filename.S	104-113	68-71		Seek/search address area
&Filename.W	114-123	72-7B		Random/sequential retrieval work area
&Filename.P	124-127	7C-7F		Prime data record count
	128	80		Status indicators :
			0-1	Not used
			2	1= File closed
			3-5	Not used
			6	1= Last prime data track full
			7	1= Block complete

DTF Assembly Label	Bytes		Bits	Function
	Dec	Hex		
	129-133	81-85		Last track index normal entry address (CCHHR)
	134-138	86-8A		Last cylinder index entry address (CCHHR)
	139-143	8B-8F		Last master index entry address (CCHHR)
	144-151	90-97		Last independent overflow record address (MBBCCHHR)
&Filename.I	152-153	98-99		Number of independent overflow tracks
&Filename.A	154-155	9A-9B		Number of full cylinder overflow areas
&Filename.O	156-157	9C-9D		Overflow record count
	158-164	9E-A4		Independent overflow area lower limit (MBBCCHH)
	165-171	A5-AB		Independent overflow area upper limit (MBBCCHH)
	172-175	AC-AF		A(&Filename.D)- Address of work area for cylinder overflow control record (COCR)
	176-179	B0-B3		A(&Filename.D+8) - Address of work area for the current track index normal entry count field
	180-183	B4-B7		A(&Filename.D+16) - Address of work area for current track index overflow entry count field
	184-187	B8-BB		A(&Filename.D+24) - Address of work area for current prime data record count field
	188-191	BC-BF		A(&Filename.D+32) - Address of work area for current overflow record count field
	192-195	C0-C3		A(&Filename.D+40) - Address of work area for track index normal entry data field
	196-199	C4-C7		A(&Filename.D+50) - Address of work area for current overflow record sequence-link field
	200-203	C8-CB		A(&IOAREAL) - Address of IOAREAL, the I/O area used for adding records to a file
	204-207	CC-CF		A(&WORKL) - Address of WORKL, work area containing user data records to be added to a file
	208-211	D0-D3		A(&Filename.K) - Address of the ADD key area
	212-215	D4-D7		A(&IOAREAL+8) - Address of key position in IOAREAL
	216-219	D8-DB		A(&IOAREAL+8+&KEYLEN) - Address of data position in IOAREAL

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.2	0 -3	00-03		A(&Filename.S+3) - Address of the seek/search address area plus 3
	4	04	0 1-5 6 7	1= Seek check indicated Not used 1= Over/under seek has occurred 1= An error has been found, but a seek check is indicated
	5 -7	05-07		A(&Filename.W) - Address of the random/sequential retrieval work area
	8 -11	08-0B		Address of IOAREAS, I/O area used for sequential retrieval
	12-15	0C-0F		Address of IOAREAR, I/O area used for random retrieval or address of IOAREA2 (if specified) for sequential retrieval
	16-19	10-13		Address of KEYARG, field containing user supplied key used for random READ/WRITE operations and sequential retrieval initiated by key
	20-23	14-17		Address of WORKR, work area used for random retrieval
	24-27	18-1B		Current sequential I/O area address
	28-31	1C-1F		1) L IOREG,*-4 - Load I/O register for sequential or 2) 4- byte NO-OP instruction for random
	32	20		X'00'= No verify; X'40'= Verify
	33	21		X'00'= Blocked; X'08'= Unblocked
	34	22		R= First prime data record on shared track
	35-39	23-27		Limits for sequential (CCHHR)
	40-41	28-29		H'0'= Blocked records H'2'= Overflow records H'8'= Unblocked records
	42	2A		X'C7'= 2311,2314 or 2319; X'FF'= 3330, 3340
	43-47	2B-2F		Initial values for sequential
&Filename.H	48-55	30-37		Current sequential DASD address (MBBCCHHR)
	56-63	38-3F		Current overflow DASD address (MBBCCHHR)
	64-65	40-41		Sequential record count
	66-67	42-43		Current track index entry for sequential (HR)
&Filename.T	68-69	44-45		Number of records tagged for deletion
	70-71	46-47		LR &IOREG,0 for random (or 2-byte NO-OP for sequential)
&Filename.G	72-79	48-4F		DASD address save area for random retrieval (MBBCCHHR)

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
	80-83	50-53		Record pointer within I/O area for write (for random retrieval)
&Filename.R	84-87	54-57		Non-first overflow record count
The following information is generated if the cylinder index in core option is specified. Bytes 88-91 (58-5B) are not used.				
	92-95	5C-5F		A(&INDAREA) - Starting address of main storage area specified for cylinder index
	96-97	60-61		AL2(&INDSIZE) - Number of bytes in main storage avialable for cylinder index
	98-105	62-69		Next cylinder index entry to be read (MBBCCHHR)
	106-110	6A-6E		Last cylinder index entry (CCHHR)
	111	6F		Core index byte:
			0	1= First time through B-transient, $$BINDEX
			1	1= End of cylinder index reached
			2	1= Index skip option specified
			3	1= Suppress index in-core option and read cylinder index
			4-7	Not used
	112-115	70-73		Pointer to key (stored by module)
The following information is generated if the prime data in core add function is specified. This information is aligned on a double word boundary. If both cylinder index in core and prime data in core add functions are specified, the following information is found in bytes (116-131) (74-83).				
	116-117	74-75		Size of IOAREAL
	118-119	76-77		Maximum number of prime data records in main storage
	120-123	78-7B		Address of write CCW's
	124-127	7C-7F		Address of read CCW's
	128	80		Switch byte:
			0	1= EOF
			1-7	Not used
	129	81		Reserved.
IJHDCWRK	130-131	82-83		Work field for I/O module.

DTF Assembly Label	Bytes Dec	Bytes Hex	Bits	Function
&Filename.B	0 -7	00-07		X'07', &Filename.S+1, X'40', 6 - Long seek CCW with command chaining
	8 -63	08-3F		Channel program build area
	64-127	40-7F		Channel progr.build area for add function only
&Filename.D	128-135	80-87		Cylinder overflow control record (COCR)
	136-143	88-8F		Current track index normal entry count field
	144-151	90-97		Current track index overflow entry count field
	152-159	98-9F		Current prime data record count field
	160-167	A0-A7		Current overflow record count field
	168-177	A8-B1		Track index normal entry data field
	178-187	B2-BB		Current overflow record sequence-link field
	188-197	BC-C5		Current track index overflow entry data field
	198	C6		X'01' - Add to EOF X'02' - Add to independent overflow area
	199-201	C7-C9		Overflow control bytes (CCH)
	202-203	CA-CB		High HR on overflow track
	204-211	CC-D3		Volume upper limit for prime data records (MBBCCHHR)
	212-217	D4-D9		CLC 0(&KEYLEN,13),0(6) - Unblocked CLC 0(&KEYLEN,13),&KEYLOC-1(6) - Blocked Utility CLC for key
	218-223	DA-DF		MVC 0(&KEYLEN,13),0(12) - Unblocked MVC 0(&KEYLEN,13),&KEYLOC-1(12) - Blocked Utility MVC for key
&Filename.E	224-227[1]	E0-E3		First entry in DSKXTN table (logical unit,cell number)
	232-235[2]	E8-EB		4X'FF' - End of DSKXTN table
&Filename.K	236+	EC-end		Key area for add only. Number of bytes depends on key length, KEYLEN

1) Each entry in the DSKXTN table is four bytes long. The minimum number of entries is 2. There is one entry per extent.

2) Location of the end-of-table indicator depends on length of DSKXTN table.

Bytes Dec	Bytes Hex	Bits	Contents	Function
0-15	0-F			Command Control Block (CCB).
16	10	0-3	B'0000'	Not used.
		4		1=DTF relocated by OPENR.
		5-7	B'000'	Not used.
17-19	11-13			Address of logic module.
20	14		X'1A'	DTF type for OPEN/CLOSE (X'1A'=diskette file).
21	15	0		1=Command chained file.
		1-2	B'00'	Not used.
		3		1=Work area specified.
		4	B'0'	Not used.
		5		1=Open; 0=Close.
		6		1=Input; 0=Output.
		7		Not used.
22-28	16-1C			Filename.
29	1D		X'06'	Device type code (X'06=3540).
30-35	1E-23		C'00CHR00'	Address of HDR1 label in VTOC.
36-37	24-25			Volume sequence number.
38	26			Open communications byte.
				Input File
		0		1=No more extents
		1-2	B'00'	Not used.
		3		1=Exit for user's EOF routine.
		4		1=Next extent on new volume.
		5-6		Not used.
		7		1=Extent switch.
				Output File
		0		1=No more extents.
		1		1=Extents needed at Close time.
		2-3	B'00'	Not used.
		4		1=Next extent on new volume.
		5		1=Extent entered via console.
		6-7		Not used
39	27	0		1=Extent bypassed before file opened (input).
		0-7		Sequence number of current extent opened (output).
40	28			Sequence number of last extent opened.
41-43	29-2B		X'000000'	Reserved.
44-47	2C-2F			Address of IOAREA1.
48-51	30-33			Address of last Read/Write CCW in chain.
52-53	34-35		X'0001'	Lower record limit.
54-57	36-39		X'00CC00RR'	End-of-data seek address (last record + 1)
58-59	3A-3B			Number of records in I/O area (used in short chain processing).
60-63	3C-3F		X'00FF0001'	Seek argument (0CHR).
64-67	40-43			End-of-file routine address (input); 4X'00' (output).
68-71	44-47		X'0049001A'	Seek argument control field.
72	48			Command chaining factor.

Bytes Dec	Bytes Hex	Bits	Contents	Function
73	49			Switch byte 1.
		0		1=Not first entry after open.
		1	B'0'	Not used.
		2		1=In close routine (output).
		3		1=Error chain to be skipped.
		4		1=End of extent.
		5-7	B'000'	Not used.
74-75	4A-4B			(record size multiplied by command chain factor)-1.
76-80	4C-50		X'FFFFFFFFFF'	Seek argument bucket.
81-83	51-53		X'000000'	Reserved.
84-87	54-57			Instruction to load user's I/O register (or NOP).
88-91	58-5B			Address of current I/O area.
92-95	5C-5F			Logical record size.
96-99	60-63			Address of last byte of the I/O area.
100	64			Logical indicators.
		0		1: ERROPT=address.
		1		1: ERROPT=IGNORE.
		2		1: ERROPT=SKIP.
		3		Not used.
		4		1=Two I/O areas.
		5-7		Not used.
101-103	65-67			Address of user's error handling routine.
104	68			CCW count (write command only).
105	69			Allowed operations
		0		1=Allow read commands.
		1		1=Allow write commands.
		2		1=Suppress unit check on C4/C6.
		3-7	B'00000'	Not used.
106	6A		X'00'	Sector factor (X'00'=128).
107	6B		X'00'	Reserved.
108	6C	0		1=Write protect.
		1		1=No feed at EOF.
		2		1=Check multivolume sequence.
		3		1=Multivolume file.
		4		1=Verify requested.
		5		1=c6s written (update ERMAP).
		6		1=Read/Write security.
		7	B'0'	Not used.
109-111	6D-6F		X'000000'	Not used.
112-119	70-77			Feed CCW.
120-127	78-7F			Define ops CCW (output); 8X'00' (input).
128-135	80-87			Seek CCW.
136-143	88-8F			TIC CCW.
144-X	90-Y		X=143+8*(# of CCWs)	Read/Write data CCWs, 1, 2, 13, or 26.
			Y=8F+8*(# of CCWs)	Read/Write CCWs.
X+1	Y+1			NOP CCW (output only).

Bytes		Bits	Contents	Function
Dec	Hex			
0 -15	00-0F			CCB
16	10	0-1		Not used
		2		COBOL open; ignore option
		3		Not used
		4		DTF Table address; constants relocated by OPENR
		5		Not used
		6		1= ASCII 0= EBCDIC
		7		Not used
17-19	11-13		3X'00'	
20	14		X'12' X'14'	Standard labeled, output Standard labeled, input, forward
21	15	0-3		Not used
		4		1= input; 0= output
		5-7		Not used
22-29	16-1D			Symbolic filename
30	1E			Not used
31	1F	0-4	B'01100'	Used as displacement by OPEN
		5		
		6-7		Reserved
32	20	0		1= Standard labels
		1-2		Not used
		3		1= No rewind
		4		Not used
		5		User label address; 1= yes, 0= no
		6-7		Not used
33-35	21-23			User label routine address
36	24	0		1= DTFPH table
		1		Not used
		2		File switch: 1= input, 0= output
		3		Not used
		4		1= EOF switch
		5-7		Not used
37-39	25-27			User label exit
40-43	28-2B		DC F'0'	Reserved for OPEN
44-87	2C-57			EOV routine
88-89	58-59		DC 2X'00'	Reserved for OPEN
90-95	5A-5F		DC 6X'00'	File serial number
96-99	60-63		DC 4X'00'	Volume sequence number
100-103	64-67		DC 4X'00'	File sequence number

Bytes Dec	Bytes Hex	Bits	Function
0 -15	00-0F		CCB
16	10	0	1= Dequeue old volume extents
		1	Not used
		2	1= File assigned 'IGN' (COBOL)
		3	Not used
		4	1= DTF relocated by OPENR
		5-7	Not used
17-19	11-13		3X'00'
20	14		DTF type (X'21')
21	15		Open/Close indicators
		0	Not used
		1	1= Blocked files
		2	1= Work file
		3	1= Work area
		4	1= Not version 1 table type
		5	1= Open; 0= Closed
		6	1= Input; 0= Output
		7	1= User labels specified
22-28	16-1C		Filename (See byte 29)
29	1D		Device type code: X'05'= 3330-11 X'00'= 2311 X'07'= 3350 X'01'= 2314,2319 X'08'= 3340 general X'09'= 3340 35MB X'04'= 3330-1, -2 X'0A'= 3340 70MB
30	1E		C'F'= EOF indicator for DTFPH
30-35	1E-23		(BCCHHR) Address of F1 label in VTOC (output) (BCCHHR) Address of next DLBL-EXTENT record (input)
36-37	24-25		Volume sequence number
38	26		Open communication byte: Output
		0	1= No more EXTENTS
		1	1= EXTENTS for LIOCS at close
		2	1= Process trailer labels
		3	1= Process header labels
		4	1= New volume on next extent
		5	1= EXTENTS entered via console
		6	1= Process trailer labels at close
		7	1= Check EXTENT for minimum of 2 tracks
			Input
		0	1= No more EXTENTS
		1	Not used
		2	1= No F1 label, process EXTENTS only
		3	Not used
		4	1= New volume on next EXTENT
		5	Not used

Bytes Dec	Bytes Hex	Bits	Function
38 (Cont'd)	26		Open communications byte (Input) (Cont'd)
		6	1= Process header labels
		7	Not used
39	27		Sequence number of current EXTENT being opened
40	28		Sequence number of last EXTENT opened (not a console EXTENT entry)
41-43	29-2B		Address of user's label routine
44	2C	2	1= Version 3 DTF
45-47	2D-2F		Not used
48-51	30-33		CCHH address of user's label track. Initially X'80000000'
52-53	34-35		Lower head limit (HH) X'0000' if type 1; X'00nn' if type 128 (n= head limit)
54-57	36-39		EXTENT upper limit (CCHH)
58-59	3A-3B		BB seek address: X'0000' if disk device
60-63	3C-3F		EXTENT lower limit (CCHH)
64	40		Record number: 1= Input; 0= Output
65-67	41-43		Not used
68-71	44-47		CCHH control bucket CCHH= X'00C80009' if 2311 - type 1 CCHH= X'00C80013' if 2314 or 2319 - type 1 CCHH= X'01940012' if 3330 - type 1 CCHH= X'03280012' if 3330-11 - type 1 CCHH= X'015C000B' if 3340 35MB CCHH= X'02B8000B' if 3340 70MB CCHH= X'022B001D' if 3350 - type 1
72	48		Record number
73	49		Not used
74-75	4A-4B		Not used
76-80	4C-50		CCHHR bucket= extent lower limit and record number
81-83	51-53		Not used

Note : where nn = current upper head number

Bytes		Bits	Function
Dec	Hex		
0-15	0-F		CCB
16	10		X'08' indicates DTF relocated by OPENR.
17-19	11-13		3X'00'.
20	14		DTF type (X'23').
21	15		Option codes.
		0	1=Output, 0=Input.
		1	Not used.
		2	Not used.
		3	Not used.
		4	Not used.
		5	Not used.
		6	Not used.
		7	Not used.
22-28	16-1C		Filename
29	1D		Device type code :
			X'00' = 2311
			X'01' = 2314, 2319
			X'04' = 3330-1, -2
			X'05' = 3330-11
			X'07' = 3350
			X'08' = 3340 general
			X'09' = 3340 35MB
			X'0A' = 3340 70MB.
30-31	1E-1F		Logical unit address of first volume containing the file.
32	20	0	Not used.
		1	1= Device supports RPS.
		2	1= Version 3 DTF
		3-7	Reserved for future use
33-35	21-23		Address of user label routine.
36-39	24-27		Address of user routine to process EXTENT information.

DTFPH (DISKETTE)

Bytes		Bits	Function
Dec	Hex		
0-15	0-F		CCB.
16	10	0	1=Dequeue old volume extents.
		1-3	Not used.
		4	1=DTF relocated by OPENR.
		5-7	Not used.
17-19	11-13		3X'00'.
20	14		DTF type (X'21').
21			Open/close indicators.
		0-2	Not used.
		3	1=Work area.
		4	1=Not version 1 DTF table type.
		5	1=Open; 0=Closed.
		6	1=Input ; 0=Output.
		7	Not used.
22-28	16-1C		Filename (see byte 29).
29	1D		Device type code (3540=X'06').
30	1E		C'F'=EOF indicator for DTFPH.
30-35	1E-23		(0CHR00) Address of HDR1 label in VTOC (output).
36-37	24-25		Volume sequence number.
38	26		Open communications byte.
			Input
		0	1=No more extents.
		1-3	Not used.
		4	1=New volume or new extent.
		5-7	Not used.
			Output
		0	1=No more extents.
		1	1=Extents for LIOCS at close.
		2-3	Not used.
		4	1=New volume on next extent.
		5	1=Extents entered via console.
		6	Not used.
		7	1=Check extent for minimum of 2 tracks.
39	27		Sequence number of current extent being opened.
40	28		Sequence number of last extent opened (not a console extent entry).
41-43	29-2B		Not used.
44-47	2C-2F		Address of IOAREA1.
48-51	30-33		Not used.
52-53	34-35		X'0000'.
54-57	36-39		Extent upper limit (0CHR).
58-59	3A-3B		Not used.
60-63	3C-3F		Extent lower limit (0CHR).
64	40		Record number. 1=Input, 0=Output.
65-67	41-43		Not used.
68-71	44-47		0CHR control bucket. 0CHR= X'0049001A' for 3540 (output only).
72	48		Record number.
73	49		X'10' - multivolume file (input) X'40' - last volume on multivolume file (input).

Bytes		Bits	Function
Dec	Hex		
74	4A		Record size (maximum of 128)
75	4B		Not used.
76-80	4C-50		0CHR bucket = extent lower limit and record number (output).
81-83	51-53		Not used.

Bytes Dec	Bytes Hex	Bits	Function
0-15	00-0F		CCB. If RPS is supported, the CCW address in bytes 9-11 (09-0B) is changed by OPEN to point to an RPS CCW string in the user virtual area. CLOSE restores it.
16	10	0-1	Not used.
		2	COBOL open; ignore option.
		3	Not used.
		4	DTF table address constants relocated by OPENR.
		5-7	Not used.
17-19	11-13		Address of logic module. If RPS is supported, OPEN changes this address to point to an RPS version of the logic module in the system virtual area. CLOSE restores it.
20	14		DTF Type =X'33'.
21	15		Open/Close indicators - X'02'=input, X'00'=output.
22-28	16-1C		Symbolic filename.
29	1D		DASD or diskette device indicators X'00'=2311 X'01'=2314, 2319 X'04'= 3330-1, -2 X'05'= 3330-11 X'07'= 3350 X'08'=3340 general X'09'=3340 35MB X'0A'=3340 70MB.
30-35	1E-23		DASD address of format-1 label.
36-37	24-25		DASD or diskette volume sequence number.
38	26		Open communications switch.
		0	1=No more extents --diskettes.
		1-3	Not used.
		4	Always 1.
		5-7	Not used.
39	27		Sequence number of current extent.
40	28		Sequence number of last extent, or X'80' for 1442 reader punch.
41	29		Open indicator = X'20'.
42	2A		Device type indicators :
		0	Unused.
		1	1=DTF has been extended into the partition GETVIS area.
		2	1=DASD
		3	1=tape
		4	1=printer
		5	1=punch
		6	1=reader
		7	1=RPS supported.
43	2B		Logic module device indicators : X'F3' = DASD or diskette device. X'F1' = reader or tape device. X'F0' = other type devices.

Bytes		Bits	Function
Dec	Hex		
44	2C		Logic module option switches
		0	1=input, 0=output.
		1	1=eject for RDR-PCH; 0=no eject.
		2	1=not first pass; 0=first pass.
		3	1=two I/O areas; 0=one I/O area.
		4	1=2540 Punch.
		5	1=SYSLST/SYSPCH.
		6	1=Tape SYSLST/SYSPCH.
		7	1=ASCII ; 0=EBCDIC code.
45-47	2D-2F		Alternate I/O area address.
48	30	0-1	Reserved for future use.
		2	1= Version 3 DTF.
		3-7	Reserved for future use.
49-51	31-33		Reserved for future use.
52-53	34-35		Extent lower head limit.
54-57	36-39		Extent upper head limit.
58-64	3A-40		DASD seek address.
			Diskette seek address at byte 60 (3C).
65-67	41-43		Users EOF address.
68-72	44-48		Control bucket CCHHR.
			Byte 72 (48) always X'01' for diskettes.
73	49		Logic module switches
			X'01'=input
			X'00'=output
			X'00'=both input and output on diskettes.
74-75	4A-4B		Logic module constants
			X'0020' DASD output
			X'0018' DASD input
			X'0008' Diskette devices
			X'0000' Non-DASD devices.
76-80	4C-50		Count field CCHHR (0CHR0 for diskettes).
81	51		Key length.
82-83	52-53		Data length.
84-87	54-57		Instruction to load IOREG with correct I/O area address.
88-103	58-67		Seek, Search CCWs.
			Seek, Read/Write CCW for diskette files.
104-111	68-6F		TIC CCW.
			NOP CCW for diskette output files; unused for diskette input files.
112-119	70-77		Input/output CCW.
120-127	78-7F		Second output CCW.
128-151	80-97		Verify CCWs for output.
152-159	98-9F		Error CCW1.
160-167	A0-A7		Error CCW2.
168-231	A8-E7		Save area (64 bytes).
232-235	E8-EB		DC A(WLRERR) if WLRERR=Address.
			B 28(15) if ERROPT=omitted.
			B 25(15) if ERROPT=SKIP.
			B 28(15) if ERROPT=IGNORE.

Bytes Dec	Bytes Hex	Bits	Function
236-239	EC-EF		DC A(ERROPT) if ERROPT=Address. B 0(15) if ERROPT=omitted. B 24(15) if ERROPT=SKIP. B 28(15) if ERROPT=IGNORE.
0-15	00-0F		CCB. If RPS is supported, the CCW address in bytes 9-11 (09-0B) is changed by OPEN to point to an RPS CCW string in the user virtual save area. CLOSE restores it.
16	10	0	Not used
		1	Set by Maint; indicates that LIOCS must retrieve extents from the VTOC instead of the label cylinder.
		2	COBOL open; ignore option.
		3	X'10' indicates an unlabeled FORTRAN tape.
		4	DTF table address constants relocated by OPENR.
		5	Used by FORTRAN (Sequential Disk Backspace and Rewind).
		6	1 = ASCII, 0 = EBCDIC.
		7	FORTRAN is calling DTFCP.
17-19	11-13		Logic module address. If RPS is supported, OPEN changes this address to point to an RPS version of the logic module in the system virtual save area. CLOSE restores it.
20	14		DTF type X'32' except in the case of disk assigned to units SYS000 to SYSnnn. In this case, a DTFCP open phase changes it to X'20'.
21	15		Open indicators: X'02' input, X'00' output, except for tapes assigned to SYS000 to SYSnnn when X'00' = input and X'08' is output. X'08' DISK=YES indicator.
		0	1 = no rewind, 0 = rewind.
22-28	16-1C		Filename (see byte 29).
29	1D		Device type code : X'00' = 2311 X'01' = 2314, 2319 X'04' =3330-1, -2 X'05'= 3330-11 X'07'= 3350 X'08' = 3340 general X'09' = 3340 35MB X'0A'= 3340 70MB.
30-35	1E-23		File address for disk; block count if bit 7 of byte 16 is on.
36-37	24-25		Volume sequence number or work area.
38	26		Open switch
39	27		Sequence number of current extent.
40	28		Sequence number of last extent, or X'80' if 1442 punch.
41	29		X'80' indicates request for standard label tape OPEN.
42	2A		X'80' device is a 2560. X'40' DTF has been extended into the user virtual save area

Bytes Dec	Bytes Hex	Bits	Function
42	2A		X'20' device is a DASD X'10' device is a tape X'08' device is a printer X'04' device is a punch X'02' device is a reader X'01' RPS is supported.
43	2B		X'F3' device is a DASD X'F1' device is a reader X'F0' device is other type.
44	2C	0	1=input, 0=output.
		1	1=eject needed for a reader punch; 0=no eject.
		2	0=first pass, 1=not first pass.
		3	1=two I/O areas, 0=one I/O area.
		4	1=2540 punch.
		5	1=SYSLST or SYSPCH.
		6	1=SYSLST or SYSPCH on output tape.
		7	1=TLBL is present and tape is labeled.
45-47	2D-2F		IOAREA2 address.
48	30	0	1= Always on.
		1	Reserved for future use.
		2	1= Version 3 DTF
		3–7	Reserved for future use.
49–51	31–33		Reserved for future use.
52-53	34-35		Lower head limit.
54-57	36-39		Extent upper limit.
58-64	3A-40		BBCCHHR seek address.
65-67	41-43		EOF address.
68-71	44-47		Control bucket CCHH.
72	48		Number of record per track for output, number of record per track +1 for input.
73	49		X'00' for output, X'01' for input.
74-75	4A-4B		X'0020' for output, X'0018' for input for DASD X'0008' for 2560 and 5424/5425 output. X'0000' for nondisk device.
76-80	4C-50		CCHHR for count field.
81	51		Key length.
82-83	52-53		Data length.
84-87	54-57		Instruction to load user I/O area address to I/O register.
88-111	58-6F		Seek, search, TIC CCWs.
112-119	70-76		CCW for DASD input and first CCW for DASD output. This CCW can be used for other device if unit is not a DASD
End-of-table if DTF is defined for an input file			
120-127	77-7F		Second CCW for output.
128-151	80-97		Verify CCWs for output.

Bytes		Bits	Function
Dec	Hex		
End-of-table if DTF is defined for output file and DEVADDR does not equal SYSPCH.			
152-159 160-167 168-231	98-9F A0-A7 A8-E7		2540 punch error recovery CCW 1. 2540 punch error recovery CCW 2. Reserved.
When the CP open initializes the table and determines that the device is a 2540 punch, the following bytes in the table are changed:			
30 32-35 48-55 56-63 64-71 72-151 152-231	1F 20-23 30-37 38-3F 40-47 48-97 98-E7		X'FF' indicator to DTFCP open phases and logic module. Instruction to load user I/O area to I/O register. CCW. 2540 punch error recovery CCW 1. 2540 punch error recovery CCW 2. 80-byte card image, savearea 1.. 80-byte card image, savearea 2.
When the CP open initializes the table and determines that the device is a 2560 or 5424/5425, the following bytes in the table are changed:			
32-35 48-55 56-63 64 65	20-23 30-37 38-3F 40 41		Instruction to load user I/O area to I/O register. First output CCW. Second output CCW. Stacker select character V for ASCII. Stacker select character W for EBCDIC.

DTFCP (DISK=NO)

Bytes		Bits	Function
Dec	Hex		
0-15	00-0F		CCB.
16	10	0-1	Not used.
		2	COBOL open; ignore option.
		3	Not used.
		4	DTF table address constants relocated by OPENR.
		5	Not used.
		6	1=ASCII (used only if DISK=YES), 0=EBCDIC (used only if DISK=YES).
		7	FORTRAN is calling DTFCP.
17-19	11-13		Logic module address.
20	14		DTF type X'32' except in the case of tape assigned to units SYS000 to SYSnnn. In this case, a DTFCP open phase changes it to X'10'.
21	15		Open indicators X'02' input, X'00' output (except for tapes assigned to SYS000 to SYSnnn when it is X'00' input, X'08' output).
22-28	16-1C		Filename (see byte 29).
29	1D		Reserved for future use.
30	1E		Indicator to DTFCP open phase and logic module. X'FF' for input files. X'00' for output files.
31	1F		Reserved for future use.
32-35	20-23		Instruction to load user's I/O area address into I/O register.
36-37	24-25		Volume sequence number or work area.
38	26		Open switch.
39	27		Sequence number of current extent.
40	28		Sequence number of last extent, or X'80' if 1442 punch.
41	29		X'20'.
42	2A		X'80' device is a 2560. X'40' device is a 6424/5425. X'10' device is a tape. X'08' device is a printer. X'04' device is a punch. X'02' device is a reader.
43	2B		X'F1' device is a reader or tape. X'F0' device is other type.
44	2C	0	1=input, 0=output.
		1	1=eject needed for a reader-punch, 0= no eject.
		2	1=not first pass, 0=first pass.
		3	1=two I/O areas, 0=one I/O area.
		4	1=2540 punch.
		5	1=SYSLST or SYSPCH.
		6	1=SYSLST or SYSPCH on output tape.
		7	Reserved for future use.
45-47	2D-2F		IOAREA2 address.
48-55	30-37		CCW.
End-of-table if DTF is defined as output file and DEVADDR is not equal to SYSPCH.			

Bytes		Bits	Function
Dec	Hex		
56-63	38-3F		2540 punch error recovery CCW 1.
64-71	40-47		2540 punch error recovery CCW 2.
65-67	41-43		EOF address, input only.
End-of-table if DTF is defined as input file			
72-151	48-97		80-byte card image, save area 1.
152-231	98-E7		80-byte card image, save area 2.
If the device is a 2560 or 5424/5425, bytes 56 onward contain the following information.			
56-63	38-3F		Second output CCW.
64	40		Stacker select character V for ASCII.
65	41		Stacker select character W for EBCDIC.
66-75	42-4B		Reserved for future use.
76-235	4C-EB		First I/O area.
236-237	EC-ED		Reserved.
238-317	EF-13D		Second I/O area.
318-319	13E-13F		Reserved.

DTFCP (DISK=PARAMETER OMITTED)

Bytes Dec	Bytes Hex	Bits	Function
0-15	00-0F		CCB
16	10	0-1	Not used.
		2	COBOL open; ignore option.
		3	Not used.
		4	DTF table address constants relocated by OPENR.
		5	Not used.
		6	1=ASCII (used only if DISK=YES), 0=EBCDIC (used only if DISK=YES).
		7	Used by FORTRAN
17-19	11-13		Logic module address.
20	14		DTF type X'31' except in the case of tape assigned to units SYS000 to SYSnnn. In this case DTFCP open phase changes it to X'10'.
21	15		Open indicators X'02' input, X'00' output (except for tapes assigned to SYS000 to SYSnnn when it is X'00' input, X'08' output.
22-28	16-1C		Filename.
29	1D		Reserved for future use.
30	1E		X'00' indicator to DTFCP open phases and logic module.
31	1F	0	1=input, 0=output.
		1	1=eject needed for a read punch, 0=no eject.
		2	1=not first pass, 0=first pass.
		3	1=two I/O areas, 0=one I/O area.
		4	1=2540 punch.
		5	1=SYSLST or SYSPCH.
		6	1=SYSLST or SYSPCH on output tape.
		7	1=TLBL specified and tape is labeled.
32	20		Open indicators.
33-35	21-23		IOAREA2 address.
36-39	24-27		Instruction to load user's I/O area address into I/O register.
40-47	28-2F		CCW.
End of table if DTF is defined as output file and DEVADDR is not equal to SYSPCH.			
48-55	30-37		2540 punch error recovery CCW 1.
56-63	38-3F		2540 punch error recovery CCW 2.
57-59	39-3B		EOF address, input only.
End of table if DTF is defined as input file.			
64-143	40-8F		80-byte card image, save area 1.
144-223	90-13F		80-byte card image, save area 2.

DTFCP (DISK=PARAMETER OMITTED)

Bytes		Bits	Function
Dec	Hex		
For 2560 and 5424/5425, bytes 48 onwards contain the following information:			
48-207	30-CF		IOAREA1.
208-209	D0-D1		Reserved
210-369	D2-171		IOAREA2.
370-371	172-173		Reserved
372-451	174-1C3		Compare area.

DTF - Table Types

DTF Type Code (Byte 20) of DTF Table	DTF	Description
X'00'	DTFCD	Combined files
X'01'	DTFPT	Paper tape files
X'02'	DTFCD	Reader and 3881 Optical Mark Reader files
X'03'	DTFCN	Console
X'04'	DTFCD	Punch files
X'05'	DTFCD	Reader files on 2560, 5425
X'07'	DTFPR	Printer files on 2560
X'08'	DTFPR	Printer files
X'09'	DTFOR	Optical Reader files except 3881 and 3886 files
X'0A'	DTFOR	Optical Reader files (HEADER=YES)
X'0B'	DTFMR	Magnetic Ink Character Recognition (MICR) and Optical Reader/Sorter files
X'0C'	DTFDR	3886 Optical Character Reader files
X'10'	DTFMT	Magnetic tape workfiles
	DTFCP	Magnetic tape workfiles (compiler). (Note 1)
X'11'	DTFMT	Nonstandard or unlabeld tape files
X'12'	DTFMT	Standard labeled, output tape files
	DTFPH	Standard labeled, output tape files (physical IOCS)
X'13'	DTFMT	Standard labeled, input tape files (read backward)
X'14'	DTFMT	Standard labeled, input tape files (read forward)
X'1A'	DTFDU	Diskette Input/Output Unit files
X'20'	DTFSD	Sequential DASD workfiles and data files
	DTFCP	DASD workfiles (compiler)
X'21'	DTFPH	Sequential DASD files, MOUNTED=SINGLE (physical IOCS)
X'22'	DTFDA	Direct access files
X'23'	DTFPH	Direct access files, MOUNTED=ALL (physical IOCS)
X'24'	DTFIS	Indexed sequential, LOAD file
X'25'	DTFIS	Indexed sequential, ADD file
X'26'	DTFIS	Indexed sequential, RETRVE file
X'27'	DTFIS	Indexed sequential, ADDRTR file
X'28'	ACB	Access Method Control Block for VSAM
X'30'	DTFCP	Compiler file for DOS Version 1 (Note 1)
X'31'	DTFCP	Compiler file for DOS Versions 2 onward
X'32'	DTFCP	Compiler file for DOS Versions 2 onward (Note 2)
X'33'	DTFDI	Device independent system unit files
X'40'	DTFBT	Basic Telecommunication Access Method (BTAM) file (Note 3)
X'50'	DTFQT	Queued Telecommunication Access Method (QTAM) file (Note 3)
X'60' - X'67'		

Notes

1. DTF type is X'30' except for tape or DASD assigned to units SYS000 to SYSnnn. In this case, the DTFCP open phases change the DTF type to X'10' for tape workfiles, or X'20' for DASD workfiles.

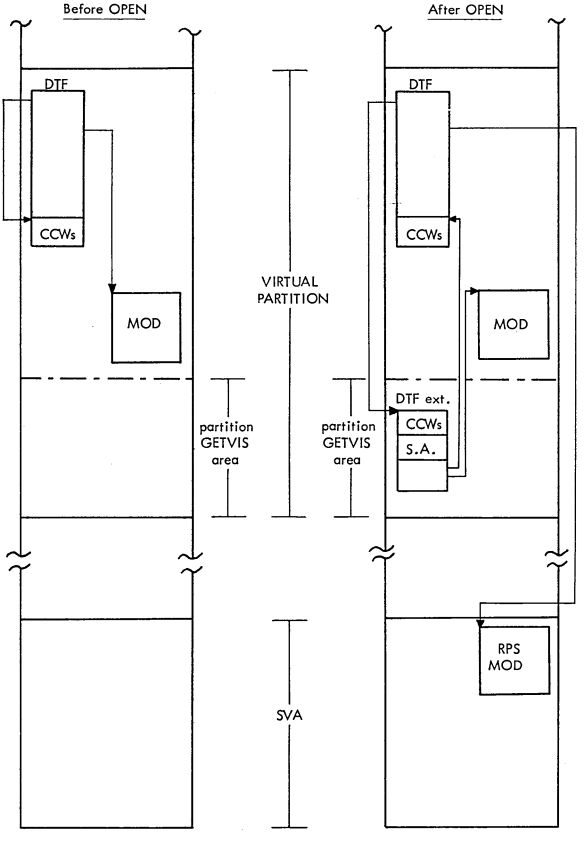
Notes (continued)

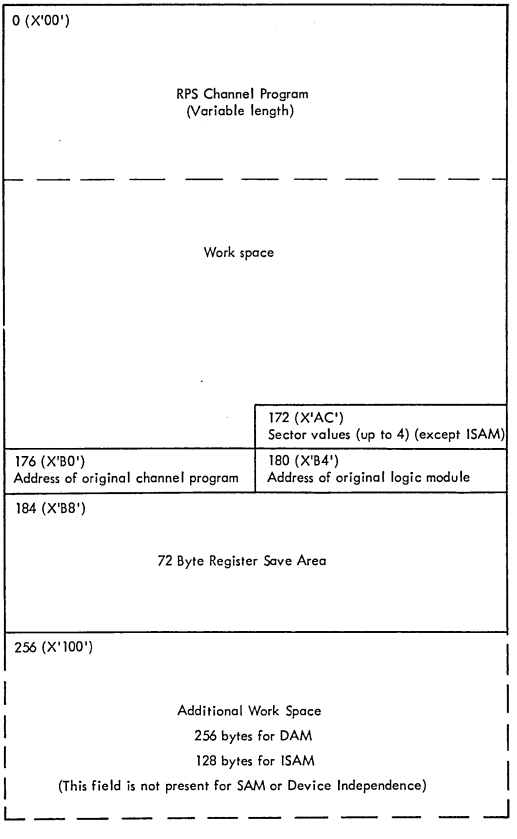
2. DTF type is X'32' except for DASD assigned to units SYS000 to SYSnnn. In this case, the DTFCP open phases change the DTF type to X'20' for DASD workfiles.

3. The following control unit codes are ORed into the low-order 4 bits of the DTF type code.

Control Unit	Code
7770	1
2848	3
2701	4
2702	5
2703	6

RPS DTF/MODULE RELATIONSHIP




0 (X'00')
RPS Channel Program
(Variable length)
Work space
172 (X'AC')
Sector values (up to 4) (except ISAM)
176 (X'B0')
Address of original channel program
180 (X'B4')
Address of original logic module
184 (X'B8')
72 Byte Register Save Area
256 (X'100')
Additional Work Space
256 bytes for DAM
128 bytes for ISAM
(This field is not present for SAM or Device Independence)

CHAPTER IV
DOS/VSE SUPERVISOR CONTROL BLOCKS AND AREAS

SUPERVISOR STORAGE ALLOCATION

Generation Macro	Macros Called	Generated Code	Base Registers Used
IOTAB	SGLOWC	Hardware/Software Interface (PSW's, Logout Areas, etc.).	R0, R11
	SGNUC	Interrupt Handler, Job Accounting In-Line Routine.	R0, R11
	SGSVC	Various SVC Routines.	R0, R11
		Various Constants and Tables must be below 8 K. CRTGEN, PIB Tables, Exit Tables, I/O Tables, Foreground Communication Regions etc., having Y-Type Address Pointers in Low Storage, must be below 32 K.	R0, R11
	SMICR	External Interrupt Handler.	R14
		C-Transient, B-Transient, and A-Transient Area	–
	SGEFCH	FTTAB and SSLD Initialization.	R9
	DISP	Task Selection.	R6
	SGAFCH	Fetch Data Section (CCWs, Control Blocks).	R11
	SGDFCH	Fetch Overall Logic and Directory Search.	R9
	SGCCWT	CCW Translation for 370 Mode.	R8, R9
	SGCCWF	CCW Analysis and Fixing Routine for ECPS:VSE Mode.	R8, R9
	SGPCK	Program Check Handler.	R14
	SGPMR	Page Manager. (SGPLLEV) Load Leveller. (SGPFIX) Fixing Routines. (SGPOPT) Page in SVCs. (SGPDATA) Data for Page Manager.	R9 R15 R9 R9 R8
	SGSVCX	Various SVC Routines.	R7, R14, R15
	MCRAS	Machine/Channel Check Handler, RTA.	R15
	SGSCVRT	RPS Convert Routine.	R9
	SGIOS	SVCO (EXCP) and SVC 15 (SYSIO) Routines. (SGSCHED) Channel Scheduler Routine. (IOINTER) I/O Interrupt Handler (SGDSK) Disk Error Recovery Routine. (SGSERI) Service Task Interface and Data.	R13 R13 R9 R13 R12
	SGCFCH	Fetch SVC Routines.	R13
	SGERP	Interface to ERP Transients.	R13
	SGAP	Asynchronous Processing SVC Routines.	R13
	SGRM	Resource Management SVC Routines.	R13
	SGLOCK	USE, RELEASE Routines.	R13
	SGAM	CDLOAD, GETVIS, and FREEVIS Routines.	R14
	SGBFCH	IDRA Area and Program Fetch.	R9
	SGSM	Allocate and Setlimit SVC Routines.	R13
	SGPREAL	Get/Free Real Storage for 370	R9
	SGSER	AVR Task and SVC Routines.	R13
	SGXECB	Cross Partition Common SVC Routines.	R12
	SGACCT	GETJA SVC Routine.	R13
	SGINF	Logical SV/PP Common SVC Routines.	R12
	SGATAB	Tables having A-Type Address Pointers in low Storage (CRTSAV, SDAGDT, ISTAVT).	–
SEND	SGEND	IPL Initialization Routine. CCW Translation Copy Buffers. DSECT Macros etc.	R7, R9

Note:
Other generation macros like PIOCS, FOPT, etc. only set globals but do not generate code.

SVC		Macro supported	Function
Dec	Hex		
0	0	EXCP	Execute Channel Program.
1	1	FETCH	Fetch any phase.
2	2		Fetch a logical transient (B-transient).
3	3		Quiesce I/O
4	4	LOAD	Load any phase.
5	5	MVCOM	Modify supervisor communication region (if issued by MVCOM macro). Fetch another physical transient (if issued by a physical transient).
6	6	CANCEL	Cancel a problem program or task.
7	7	WAIT	Wait for a CCB (or IORB) or TECB.
8	8		Transfer control to the problem program from a logical transient (B-transient).
9	9	LBRET	Return to a logical transient (B-transient) from the problem program after an SVC 8.
10	A	SETIME	Set timer interval.
11	B		Return from a logical transient (B-transient).
12	C		Reset PCIL being condensed bit (displ. 59 in COMREG).
13	D		Set PCIL being condensed bit (displ. 59 in COMREG).
14	E	EOJ	Cancel job and go to job control for end of job step.
15	F	SYSIO	Headqueue I/O request and execute channel program.
16	10	STXIT(PC)	Provide supervisor with linkage to user's PC routine for program check interrupts.
17	11	EXIT(PC)	Return from user's PC routine.
18	12	STXIT(IT)	Provide supervisor with linkage to user's IT routine for interval timer interrupts.
19	13	EXIT(IT)	Return from user's IT routine.
20	14	STXIT(OC)	Provide supervisor with linkage to user's OC routine. for external or attention interrupts (operator comm.).
21	15	EXIT(OC)	Return from user's OC routine.
22	16		Seize/Release system; Enable/disable external and I/O interrupts; Set key in user's PSW.
23	17		Store the load address of a phase at a defined user address.
24	18	SETIME	Set timer interval and provide supervisor with linkage to user's TECB, if any.
25	19		Issue HALT I/O on a teleprocessing device, or HALT I/O on any device if issued by OLTEP. Dequeued an unstarted OLTEP I/O request to a shared device.
26	1A		Validate address limits.
27	1B		Issue an HIO for a telecommunication device without dequeueing the CCB.
28	1C	EXIT(MR)	Return from user's stacker select routine (MICR type devices only).
29	1D	WAITM	Provide support from multiple wait macro WAITM.
33	21	COMRG	• Force task select for system tasks.
34	22	GETIME	Provides Time-of-Day and updates the DATE field.
35	23		Hold a track for use by the requesting task only.
36	24	FREE	Free a track held by the task issuing the FREE.

SVC		Macro supported	Function
Dec	Hex		
37	25	STXIT(AB)	Provide supervisor with linkage to user's AB routine for abnormal termination of a task.
38	26	ATTACH	Initialize a subtask and establish its priority.
39	27	DETACH	Perform normal termination of a subtask. It includes calling the FREE routine to free any tracks held by the subtask.
40	28	POST	Inform the system of the termination of an event and ready any waiting tasks.
41	29	DEQ	Inform the system that a previously enqueued resource is now available.
42	2A	ENQ	Prevent tasks from simultaneous manipulation of a shared data area (resource).
44	2C		Support the creation of unit check records from outside the A- or R-transient area.
45	2D		Provide emulator interface.
46	2E		Provide OLTEP with the facility to operate in supervisory state.
47	2F	WAITF	Provide support for multiple wait macro WAITF for MICR type devices.
48	30		Fetch a CRT transient.
49	31		Used by VTAM to init. exec. of channel program
51	33	HIPROG	Make directory entry information for a phase available to the requesting task. Calculate the highest address of an overstructure of phase and store it in the COMPREG.
52	34	TTIMER	Return the remaining time interval, or cancel a time interval.
53	35		Used by VTAM/ACF to schedule user exit in applic. program.
54	36		Replease page frames to selection pool. Applies only to 370 mode of operation.
55	37		Allow SDAID to acquire processor storage needed for program initialization (applies only 370 mode of operation).
56	38		Support the POWER/VS-CP interface when DOS/VSE operates under VM/370 (applies only 370 mode of operation).
57	39	GETPRTY SETPRTY	Return partition priorities to the requesting task. Change partition priorities as specified.
58	3A	INVPART	Initialize partition.
59	3B	INVPAGE	Initialize Tables or invalidate pages.
60	3C	GETDADR	Provide virtual address of location within I/O areas for ERP and CRT routines.
61	3D	GETVIS	Request allocation of storage within the same partition or within the SVA.
62	3E	FREEVIS	Free storage requested through a GETVIS macro.
63	3F	USE	Use a resource.
64	40	RELEASE	Release a resource.
65	41	CDLOAD	Load a phase in the requesting partition's GETVIS area unless that phase is already in the SVA.
66	42	RUNMODE	Return mode which program is running.
67	43	PFIX	Fix page(s) in real storage.
68	44	PFREE	Free page(s) in real storage.
69	45	REALAD	Return real address corresponding to a given virtual address.

SVC		Macro supported	Function
Dec	Hex		
70	46	VIRTAD	Return virtual address corresponding to a given real address.
71	47	SETPFA	Establish or terminate the linkage between the supervisor and an user page-fault appendage routine.
72	48	GETCBUF/FEECBUF	Get or free copy buffer for IDAL or tape ERP.
73	49	SETAPP	Allow linkage to channel and appendage routines.
74	4A		Fix page(s) in real storage for restart.
75	4B	SECTVAL	Calculate a sector value (disk dev. with RPS feature)
76	4C		Initiate recording on SYSREC file.
77	4D	TRANSCSW	Return the virtual address of an ERP CCW address copied from the pertinent CSW.
78	4E	CHAP	Change subtask priority (supported if AP=YES).
79	4F	SYNCH	Give control to synchronous exit.
80	50	SETT	Set task time interval.
81	51	TESTT	Return the remaining task time interval or cancel a time interval.
82	52		Reserved.
83	53	ALLOCATE	Allocate real or virtual partitions.
84	54	SETLIMIT	Set partition sizes.
85	55	RELPAG	Release contents of one or more pages.
86	56	FCEPGOUT	Force a page-out of one or more pages.
87	57	PAGEIN	Page-in one or more pages.
88	58	TPIN	Start TP Balancing.
89	59	TPOUT	Stop TP Balancing.
90	5A	PUTACCT	Provide interface with POWER/VS for additional account information (by user).
91	5B		Provide interface with POWER/VS for standard account information.
92	5C	XECBTAB	Define, delete, or check an entry in the cross partition ECB table.
93	5D	XPOST	Set the traffic bit in a cross-partition ECB and ready any waiting tasks.
94	5E	XWAIT	Wait for a cross-partition ECB to be posted.
95	5F	EXIT AB	Return from an user's abnormal termination routine.
96	60	EXIT(TT)	Return from user's task timer exit.
97	61	STXIT(TT)	Provide supervisor with linkage to user's task timer exit routine for task time interval end.
98	62	EXTRACT MODCTB	Extract system control information. Modify a PUB 2 table entry.
99	63	GETVCE	Return a specific volume characteristics table entry.
100	64	PFIX PFREE	Fix or free a page in the system GETVIS area.
101	65	MODVCE	Update the volume characteristics table.
102	66	GETJA	Update the fields in the requesting partition's job accounting table.
103	67		Execute I/O operations for SYSFIL on a FBA device.
104	68	EXTENT	Build, return, or delete DASD extent information.
105	69	SUBSID	Accept, return, and delete subsystem identification information.

COMMAND CONTROL BLOCK (CCB)

Count	Trans-mission Informa-tion	CSW Status Bits	Type Code and logical Unit	Reserved for logical IOCS or 3895 PIOCS	CCW Address	Reserved for physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Bytes		Description	
0–1	Used for residual Count.	Used by BTAM to indicate the numer of copy blocks needed at channel end appendage time (370 mode only).	
2–3	Transmitting information between Physical IOCS & Problem Program	Byte 2	Set on by:
		Bit 0: Traffic Bit (Wait) (Note 5)	PIOCS *
		Bit 1: End of File (/* or /&) PRT1 UCSB Parity Check (Line Complete) (Note 2)	PIOCS
		Bit 2: Unrecoverable I/O error	PIOCS
		Bit 3: Accept unrecoverable I/O error	Pr. Pr.**
		Bit 4: Return DASD Data Checks Diskette Data Checks, 2671 errors, or 1017/1018 errors to the user; indicate actiontype messages for DOC; Return 5424/5425 not ready.	Pr. Pr.
		Bit 5: Post at Device End (Note 5)	Pr. Pr.
		Bit 6: Return Tape Read Data Check; 1018 or 2560 Data Check; 2520, 2540, 2560, 3881 or 5424/5425 Equipment Check; Accept 3504, 3505 or 3525 Perm. Error; DASD Data Chechs on Read or Verfiy Command on 3203, PRT1, or 5203 Pass-back Requested. (Notes 3, 6, 8 and 10)	Pr. Pr.
		Bit 7: User Error Routine (Note 9)	Pr. Pr.
		Byte 3	Set on by:
		Bit 0: DASD Data Check in Count Area; Permanent Error for 3330, 3340 or 3350 MICR-SCU Not Operational; 1287/1288 Data Check; 3203, PRT1, or 5203 Print Check/Equipment Check; 3540 Special Record Transferred.	PIOCS
		Bit 1: DASD Track Overrun; MICR Intervention required; 1287-Keyboard Correction in Journal Tape Mode; 1017-Broken Tape PRT1-Print Quality/Equipment Check.	PIOCS
		Bit 2: DASD End of Cylinder; MICR- (Note 4) 1287/1288-Hopper Empty in Document Mode. PRT1/2245 Line Position Error. (Note 7)	
		Bit 3: 2520, 2540, 3881-Equipment Check; 2560 3203, 5203, 5424/5425 Data Check/ Equipment Check; Tape Read Data Check; DASD-Any Data Check; 1287-Equipment Check; 1017/1018 Data Check; PRT1-Print Check/Data Check; 3504, 3505, 3525 Perm. Error (Note 8); Diskette Data Check.	PIOCS
		Bit 4: Non-Recovery Questionable Condition: Card-Unusual Command Sequence DASD-No Record Found 1287/1288-Document Jam or Torn Tape; PRT1-UCSB Parity Check (Command retry); 5424/5425 Not Ready.	PIOCS
		Bit 5: No Record Found Condition (Retry on Disk Devices).	Pr. Pr.

* Physical IOCS
** Problem Program

COMMAND CONTROL BLOCK (CCB) (. . . Cont'd)

Count	Trans-mission Informa-tion	CSW Status Bits	Type Code and logical Unit	Reserved for logical IOCS or 3895 PIOCS	CCW Address	Reserved for physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 7	8	9 11	12	13 15	16 23

Byte(s)		Description	
2–3	(. . . Cont'd)	Byte 3 (Cont'd)	
		Bit 6: Carriage Channel 9 Overflow or Verify Error for DASD; 1287-Document Mode-Late, Stacker Select; 1288-End of Page.	PIOCS
		Bit 7: Command Chaining, Retry from the next CCW to be executed.	Pr. Pr.
4–5	CSW Status Bits	Byte 4 (Note 1) Bit 0: Attention 1: Status Modifier 2: Control Unit End 3: Busy 4: Channel End 5: Device End 6: Unit Check 7: Unit Exeption	Byte 5 Bit 0: Program Controlled Interruption 1: Incorrect Length 2: Program Check 3: Protection Check 4: Channel Data Check 5: Channel Control Check 6: Interf. Control Check 7: Chaining Check
6–7	Type Code and Logical Unit	Byte 6 X'0u' Original CCB X'08' Physical addressing is requested. Byte 7 contain the PUB index. X'2u' Translated CCB X'4u' BTAM request original CCB X'6u' BTAM request translated CCB X'8u' User-translated CCB in virtual partition u: 0 = The address in byte 7 refers to a System Logical Unit. 1 = The address in byte 7 refers to a Programmer Logical Unit.	
		Byte 7 Hexadecimal representation of SYSnnn: SYSRDR = 00 SYSREC = 0A SYSIPT = 01 SYSCLB = 0B SYSPCH = 02 SYSVIS = 0C SYSLST = 03 SYSCAT = 0D SYSLOG = 04 SYS000 = 00 SYSLINK = 05 SYS001 = 01 SYSRES = 06 SYS002 = 02 SYSSLB = 07 ⋮ SYSRLB = 08 ⋮ SYSUSE = 09 SYS240 = F0	
8	Reserved for Logical IOCS or 3895 PIOCS (Note 10)	Buffer Offset: ASCII Input Tapes X'00'–X'63' ASCII Output Tapes Fixed X'00' Variable X'00' or X'04' Undefined X'00' 2501 Double CCW-Support X'80' (2501 Double-CCW Support is active) SNS Task I/O Request X'80' (I/O Error on Alternate Channel)	
9–11	CCW Address	Virtual or real address of CCW associated with this CCB depending on byte 6: Real address if byte 6 = X'2u', X'6u', or X'8u'; Virtual address if byte 6 = X'0u', or X'4u'.	

Count	Trans-mission Informa-tion	CSW Status Bits	Type Code and logical	Reserved for logical IOCS or 3895	CCW Address	Reserved for physical IOCS	CCW Address in CSW	Optional Sense CCW
0 1	2 3	4 5	6 Unit 7	PIOCS 8	9 11	12	13 15	16 23

Byte(s)		Description
12	Reserved for Physical IOCS	X'80' CCB being used by ERP X'40' Channel Appendage Routine present for TP Device, VSAM or POWER/VS X'20' Sense Information desired (Note 9) X'10' Message writer X'08' EU Tape Error X'04' OLTEP Appendage available X'02' Tape ERP Read Opposite Recovery X'01' Reserved
13–15	CCW Address in CSW	Virtual Address of CCW pointed to by CSW at Channel End (if byte 6 = X'8u', it ist the real address) or address of the Channel End Appendage Routine for TP devices, VSAM or POWER/VS.
16–23	Optional Sense CCW	8 bytes appended to the CCB when Sense Information is desired.

Note 1: Bytes 4 and 5 contain the status bytes of the Channel Status Word (Bits 32–47). If byte 2, bit 5 is on and device end results as a separate interrupt, device end will be OR-ed into CCB byte 4.

Note 2: Indicates /* or /& statement on SYSRDR or SYSIPT. Byte 4, bit 7 (unit exception) is also on.

Note 3: DASD data checks on count not returned.

Note 4: For 1255/1259/1270/1275/1419, disengage. For 1275/1419D, I/O Error is external interrupt routine (Channel data check or bus-out check).

Note 5: The traffic bit (Byte 2, bit 0) is normally set on at channel end to signify that the I/O was completed. If byte 2, bit 5 has been set on, the traffic bit and bits 2 and 6 in byte 3 will be set on at device end. Also see Note 1.

Note 6: 1018 ERP does not support the Error Correction Function.

Note 7: This error occurs an equipment check, data check or FCB parity check. For 2245, this error occurs as a data check or FCB parity check.

Note 8: For 3504, 3505, 3525 input or output files using ERROPT, byte 3, bit 3 is set on if a permanent error occurs. Byte 2, bit 6 is set on to allow you to accept permanent errors.

Note 9: If user error routine is specified and the user needs the sense information to further process the error, byte 12, bit 2 must alos be set. Otherwise, the supervisor error routine will clear off the status on return and the sense information ist not available.

Note 10: 3895 error codes are returned in CCB Byte 8. Refer to 3895 document reader/inscriber maschine and programming description for information on these error codes.

INPUT/OUTPUT REQUEST BLOCK (IORB)

Count	Trans-mission Informa-tion	CSW Status Bits	Type Code and Logical Unit	Reserved for Logical IOCS	CCW Address	Reserved for Physical IOCS
0	2	4	6	8	9	12

CCW Address in CSW	Fix Flag	Address of Fixlist	Version ID	Special Process-ing Flags	Optional Extension	
					ID	Address
13	16	17	20	22	24	25

Input/Output Request Block (IORB)

Byte(s)	Description
0–1	Used for residual count
2–3	Byte 2: For transmitting information between physical IOCS and problem program. Set by the physical IOCS: Bit 0: Traffic bit, wait (Note 3) Bit 1: End-of-File /* or /& (Note 2) Bit 2: Irrecoverable I/O error Set by the Problem Program: Bit 3: Accept irrecoverable I/O error Bit 4: Reserved Bit 5: Post at device end (Note 3). Bit 6: Reserved Bit 7: Skip system error recovery Byte 3: Reserved für ERP return information.
4–5	Byte 4 (Note 1): Bit 0 (32): Attention Bit 1 (33): Status modifier Bit 2 (34): Control unit end Bit 3 (35): Busy Bit 4 (36): Channel end Bit 5 (37): Device end Bit 6 (38): Unit check Bit 7 (39): Unit exception Byte 5: Bit 0 (40): Program controlled interruption Bit 1 (41): Incorrect length Bit 2 (42): Program check Bit 3 (43): Protection check Bit 4 (44): Channel data check Bit 5 (45): Channel control check Bit 6 (46): Interface control check Bit 7 (47): Chaining check
6–7	Byte 6 (Class byte): Bit 0: Reserved Bit 1: Reserved Bit 2: Copied IORB (370 mode only) Bit 3: Reserved Bit 4: Physical addressing Bit 5: IORB ID Bit 6: Reserved Bit 7: Programmer logical unit Byte 7 (Addressing byte) Hexadecimal representation of SYSnnn: SYSRDR = 00 SYSREC = 0A SYSIPT = 01 SYSCLB = 0B SYSPCH = 02 Reserved = 0C SYSLIST = 03 SYSCAT = 0D SYSLOG = 04 SYS000 = 00 SYSLINK = 05 SYS001 = 01 SYSRES = 06 SYS002 = 02 SYSSLB = 07 SYSRLB = 08 SYSUSE = 09 SYS240 = XX

Byte(s)	Description
8	Reserved for Logical IOCS
9–11	Address for the CCW associated with this IORB. The address is virtual, except if in byte 6, bit 2 is on, then the address is real.
12	Reserved for physical IOCS: Bit 0: IORB is used by ERP Bit 1: Reserved Bit 2: IORB has an extension Bit 3: Reserved Bit 4: EU tape error Bit 5: Reserved Bit 6: Tape ERP read opposite recovery Bit 7: Reserved
13–15	Virtual address of CCW pointed to by CSW at Channel End.
16	Fix Flag (ignored in 370 mode): Bit 0 = 1 Compressed: The system needs not to compress the fixlist. Each page to the fixed for the channel program is covered only once by the fixlist. = 0 Not compressed: The system must compress the fixlist. Bit 1 = 1 Fixed: No fixlist is provided by the user. The user has fixed all areas. = 0 Not fixed: The user has provided a fixlist. Bit 2–7 Reserved
17–19	Address of the fixlist (ignored in 370 mode): Each fixlist area is contiguous and consists of one or more 8-Byte fixlist entries. Each entry contains a begin and an end address describing a storage area that has to be fixed for the I/O request (an area containig the channel program or an input/output area).
20–21	Version identification code.
22–23	Special processing flags (set by LIOCS): Bit 0: SYSFIL request for FBA device. Bits 1–15: Reserved

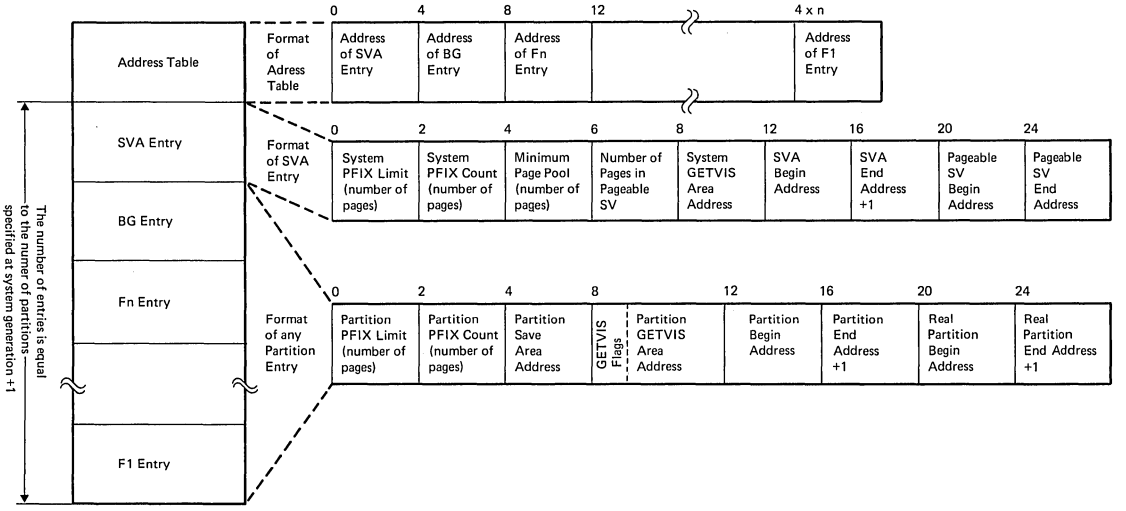
Begin of optional parameters (set by the problem program).

24	Optional parameter ID: Bit 0: Last parameter Bits 1–7: ID code (00 = ECB)
25–27	Address portion of optional parameter

Note 1: Bytes 4 and 5 contain the status bytes of CSW (Bits 32–47). If byte 2, bit 5, is on, the accumulated interrupt information will be stored in byte 4 and 5 of the IORB.

Note 2: Indicates /* or /& statement on SYSRDR or SYSIPT. Byte 4, bit 7, (unit exception) is forced on.

Note 3: The traffic bit (byte 2, bit 0) is normally set on at channel end to signify that the data transfer is completed. If byte 2, bit 5, has been set on, the traffic bit is set on at device end. See also Note 1.



Bytes 220–223 (X'DC'–X'DF') of the system communication region (SYSCOM) contain the address of the storage management control block (SMCB). Label SMCB identifies the first byte of the tabel.

SYSCOM

Displacement Hex	Displacement Dec						
0	0	4	8	0C	10		18
		4	8	12	16		24
Address of Error Block		Address of Attention Exit	Address of Operator Option Cancel Exit	Address of SYSRES PUB	Reserved		Address of Ext. Interr. Routine
xxxx		xxxx	xxxx	xxxx	xxxxxxxx		xxxx

1C	20	24	25	28	2A	2C	2E	30
28	32	36	37	40	42	44	46	48
Address of Logical Transient Area	Address of 1st byte of Problem Program Area	Free List Pointer	Address of Channel Queue	Number of Channel Queue Entries	Length of One Error Queue Entry	Number of Partitions	Flags and Switches (see expansion)	Reserved
xxxx	xxxx	x	xxx	xx	xx	xx	xx	xxxx

34	35	38	40	44	46	48	4C
52	53	56	64	68	70	72	76
Configuration Byte (see expansion)	Address of CRT Table	Reserved	Flags and Switches (see expansion)*	System Task Selection Control Field*	Address of Task Selection	Reserved	TH Free List Pointer
x	xxx	xxxxxxxx	xxxx	xx	xx	xxxx	x

* See end of tables for further explanation.

Note: The address of SYSCOM can be found at fixed location X'80'–X'83'.

4D 77	50 80	54 84	58 88	5A 90	5C 92	60 96	64 100	68 104
Address of TH Table	Address of Timer Request Table	Address of AB Table	Key of Task owing LTA (LIK)	Key of Task running (TIK)	Address of POWER/VS Table	Address of VTAM Address Vector Table	Address of RF Table	Address of EU ECB Table
xxx	xxxx	xxxx	xx	xx	xxxx	xxxx	xxxx	xxxx

6C 108	70 112	74 116	78 120	7C 124	80 128	88 136
Address of OLTEP bucket	Address of RAS Linkage Area	Address of ASCII Translate Table	Address of PUB Ownership Table	Address of Job Accounting Common Table	Reserved	Addres of SDAID Comm. Area
xxxx	xxxx	xxxx	xxxx	xxxx	xxxxxxxx	xxxx

8C 140	90 144	94 148	98 152	9C 156	A0 160	A1 161	A2 162	A3 163
Address of Line Mode Table	Reserved	Address of PTA	Address of first System Task Block	Address of Task Block of Active System Task	1 byte for Alignment	Pointer to SENSE Task Block	Pointer to Disk ERP Task Block	Pointer to RAS Task Block
xxxx	xxxx	xxxx	xxxx	xxxx	x	x	x	x

A4 164	A5 165	A6 166	A7 167	A8 168	A9 169	AA 170	AB 171	AC 172	AF 175	B0 176
Pointer to PMGR Task Block	Pointer to PGT Task Block	Pointer to PAGEIN Task Block	Pointer to SUPV (FETCH) Task Block	Reserved	Pointer to CRT Task Block	Reserved	Pointer to ERP Task Block	Reserved	Pointer to SVF Task Block	Address of Task Timer Table
x	x	x	x	x	x	x	x	xxx	x	xxxx

B4 180	B8 184	BC 188	BE 190	C0 192	CB 203	CC 204	CE 206	D0 208
Reserved	TRTMSK pointer	TP Balanc-ing Parameter	Key of partition owning Task Times	Repositioning Information for 2560/5424/5425 ERP	Number of Error Queue Entries	Length of PUB Table in bytes	Number of Active Partitions	Address of Segment Table (370 mode)
xxxx	xxxx	xx	xx	xxxxxxxxxx	x	xx	xx	xxxx

D4 212	DC 220	E0 224	E4 228	E6 230	E8 232	EC 236
Reserved	Address of Storage Management Control Block	Address of DPD Table	Address of System Operator Console	Reserved	Reserved	Address of End of Real Storage in 370 mode
xxxxxxxx	xxxx	xxxx	xx	xx	xxxx	xxxx

F0 240	F4 244	F5 245	F8 248	FC 252	100 256	104 260	108 264
Address of Fetch Table	SVA Flags (see expansion)	Address of SVA	Address of System GETVIS area	Address of RPS Local Directory List	Address of RPS Sector Calculation Routine	Address of System Code	Reserved
xxxx	x	xxx	xxxx	xxxx	xxxx	xxxx	xxxx

10C 268	110 272	114 276	118 280	11C 284	120 288	124 292	128 296
Pointer for Symbolic Label Access	Address of Supv.-IPL Communication Region	Address of SVA Phase Area	Address of Operator Option Cancel Exit	Reserved	Save area if Progr. Check in SUPVR	Reserved	Table of System Units
xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx	xxxx

* See end of tables for further explanation.

Note: The address of SYSCOM can be found at fixed Location X'80'—X'83'.

Expansion of SYSCOM Flag Bytes

Byte Dec	Byte Hex	Description
46	2E	Bit 0 1 = VSE/Advanced Functions package 0 = SCP package Bit 1 1 = DOS/VSE and follow-on releases Bit 2 1 = 1 TPBAL not active Bit 3 Reserved Bit 4 1 = CBF supported Bits 5–7 Reserved
47	2F	Bit 0 1 = 4300 support generated 0 = /370 support generated Bits 1–4 Reserved Bit 5 1 = AT least on CKD disk supported Bit 6 1 = FBA support generated Bit 7 1 = 3800 support generated
52	34	Bit 0 1 = CRT support initialized Bits 1–5 Reserved Bits 6, 7 00 = Typewriter support generated 01 = 125D support generated 11 = 3277 support generated
64	40	Reserved for RMS X'80' RMSR supported X'40' Full RMS support (MCAR/CCH and RMSR) X'20' Reserved, must be 0
65	41	X'80' Initial selection of ERP X'40' Reserved X'20' Timer interrupt pending X'10' MICR Stacker-select active X'08' Invalid address during fetch X'04' SIO routine entered after interrupt X'02' Reserved X'01' IPL in progress
66	42	X'80' Initial RAS request X'40' RAS WAIT request outstanding X'20' RAS IPL in progress X'10' Reserved X'08' POWER/VS supported, always on X'04' POWER/VS initialized X'02' GETREAL for SDAID in progress X'01' Reserved
67	43	X'80' System GETVIS area initialized X'40' ECPREAL supported, always on X'20' VSAM supported, always on X'10' Reserved X'08' XECB support generated X'04' Reserved X'02' Batch deactivated by TPIN X'01' Reserved
68	44	Always zero
69	45	SELECT byte: X'00' No system task active X'01' SNS active X'02' DSK active X'03' RAS active X'04' PMGR active X'05' PGT active X'06' PGIN active X'07' SUPV active X'09' CRT active X'0B' ERP active X'0F' SVT active

Expansion of SYSCOM Flag Bytes

Byte		Description
Dec	Hex	
244	F4	SVA Flag X'80' Reserved X'40' SDL active X'20' Reserved X'10' Build of SDL in progress X'08' SDL overflow X'04' Reserved X'02' Reserved X'01' Reserved
252	FG	X'00000000' RPS not initialized X'00XXXXXX' Pointer to RPS LDL in SVA
256	100	X'00000000' No RPS support X'00XXXXXX' Pointer to Sector Calculation Routine

nnCOMREG

Displacement Hex	Displacement Dec	Date	Reserved	User area	UP SI Byte	Job Name	Highest Storage Address of the Partition	End Address of Last Phase Fetched or loaded	Address of uppermost Byte of Phase with highest Ending Address	Label Area Length
0	0									
8	8									
0C	12									
17	23									
18	24									
20	32									
24	36									
28	40									
2C	44									
		xxxxxxxx	xxxx	xxxxxxxxxxxx	x	xxxxxxxx	xxxx	xxxx	xxxx	xx

Displacement Hex	Displacement Dec	PIK	End of Virtual Storage Address	Machine Configur. Byte	System Configur. Byte	Standard Language Translator I/O Options	Dump, Log, RELLDR and ASCII Options	Job Control Byte	Linkage Control Byte	Language Translator Control Byte	Job Duration Indicator Byte	Reserved	Address of FOCL
2E	46												
30	48												
34	52												
35	53												
36	54												
37	55												
38	56												
39	57												
3A	58												
3B	59												
3C	60												
3E	62												
		xx	xxxx	x	x	x	x	x	x	x	x	xx	xx

Job Control Switches

Displacement Hex	Displacement Dec	Address of PUBTAB	Address of FAVP	Address of JIBTAB	Address of BTAB	Address of FICL	Address of NICL	Address of LUBTAB	Line Count for SYSLST	System Date	LIOCS Comm. Bytes	Address of PIB Table	ID Number of last Checkpoint or DASDFP Indicator
40	64												
42	66												
44	68												
46	70												
48	72												
4A	74												
4C	76												
4E	78												
4F	79												
58	88												
5A	90												
5C	92												
		xx	xx	xx	xx	xx	xx	xx	x	xxxxxxxx	xx	xx	xx

Note:
A communication region exists for each partition supported by the system.
The address of the communication region of the active partition is in fixed loc.X'14'—X'17'.

5E 94	60 96	62 98	63 99	64 100	66 102	68 104	6A 106	6C 108	6E 110
Job Zone in Minutes	Address of Disk Information Block (DIB)	Device Flag for Automatic Close	Reserved	Address of PC Option Table less 8 bytes	Address of IT Option Table less 8 bytes	Address of OC Option Table less 8 bytes	Key of Partition	Reserved	Logical Transient Key (LTK)
xx	xx	x	x	xx	xx	xx	xx	xx	xx

70 112	74 116	78 120	7C 124	7E 126	80 128	84 132	86 134	87 135
Address of SYSPARM	Address of J. A. Partition Table	Address of TOD clock Common Area	Address of PIB Table Extension	Address of MICR DTF Table (PDTABB)	Slot for Pointer to Lable Area	Address of BG Comm. Region	Option Indicator	System Configuration Byte 2 and RMSR Open Flag Byte
xxxx	xxxx	xxxx	xx	xx	xxxx	xx	x	x

88 136	8C 140	8D 141	8E 142	8F 143	97 151	98 152	9F 159
Reserved for compatability Reasons	Standard Job Control Options 1	Temporary Job Control Options 1	Disk Configuration	Catalog Procedure Name	Switch for Catalog Procedure	JCL Statement Name	81 bytes SYSIN Indicator
xxxx	x	x	x	xxxxxxxx	x	xxxxxxx	x

Note:
A communication region exists from each partition supported by the system.
The address of the communication region of the active partition is in fixed loc. X'14'—X'17'.

A0 160	A4 164	A5 165	A6 166	A8 168	AC 172	AD 173	AE 174	AF 175	B0 176	B4 180
Address of POWER/VS Partition Control Block	POWER/VS Flag Byte 1	POWER/VS Flag Byte 2	Reserved	Address of LUB Table Extension	Job Control Switch 5	Job Control Switch 6	Standard Job Control Options 2	Temporary Job Control Options 2	Reserved	Reserved
xxxx	x	x	xx	xxxx	x	x	x	x	xxxx	xxxx

Note:
A communication region exits for each partition supported by the system.
The address of the communication region of the active partition is in fixed loc. X'14'–X'17'.

Key to Communication Region Displacement

0 MM/DD/YY or DD/MM/YY either set permanently by the job control date statement, or updated every time a GETIME macro is issued when time-of-day support is provided. Format controlled by BGCOMREG + 53.
(System Configuration Byte, date convention bit 0).

8 Reserved

12 User area.

23 User program switch indicator.

24 Job name set by the job control program form information found in the job statement.

32 Address of the uppermost byte available to the problem program.

36 Address of the uppermost byte of the last phase of the problem program fetched of loaded. Not filled in when phase is in SVA.

40 Address of the uppermost byte of the phase with the highest ending address for this partition, starting with the same 4 characters as the root phase (operand on the exec statement) and residing in the same core image library as the root phase. If the root phase is in the SVA, the partition start address plus 2 K will be used.

44 Length of the problem program label area.

46 Partition identification key (PIK) of the partition owning this communication region. However, the communication region of the BG partition always contains the PIK of the currently active partition. That is, if contains the PIK of the BG partition only, if the BG partition is active.

48 End address of virtual storage.

Expansion of COMREG Flag Bytes

Byte Dec	Byte Hex	Description
52	34	Machine Configuration Byte (Values set as supervisor generation time) X'80' Always set to indicate standard storage protect X'40' Decimal feature (always set) X'20' Floating point feature X'10' Physical transient overlap option (always set) X'08' Always set to indicate standard timer feature X'04' Channel switching support X'02' Burst mode on multiplex channel support (always set) X'01' Reserved
53	35	System Configuration Byte X'80' DDMMYY (Date convention bit set by STDOPT statement) X'40' Two or more partitions, always on X'20' DASD file-protect support X'10' DASD SYSIN – SYSOUT X'08' Teleprocessing, always on X'04' Two or more partitions, always on X'02' Asynchronous processing, always on X'01' Track hold/Block hold
54	36	This byte contains the standard language translator I/O options after generation, defaults are valid: 1100 110 X'80' DECK option output objekt modules on SYSPCH X'40' LIST option output source module listings and diagnostics on SYSLST X'20' LISTX option output hexadecimal object module listings on SYSLST (compilers only) X'10' SYM option output symbol tables on SYSLST/SYSPCH X'08' XREF option output symbolic cross-reference list on SYSLST X'04' ERRS option output diagnostics on SYSLST (compilers only) X'02' CHARSET option input on SYSIPT is 48 or 60 character set X'01' Reserved
55	37	This byte contains the standard supervisor options for abnormal EOJ. Relocating Loader and Control statement display and the indicator for the presence of the ASCII-EBCDIC and EBCDIC-ASCII translation tables. X'80' Always on X'40' DUMP option (DUMP=YES OR DUMP=PART) X'20' Partition is in wait state, because a volume is to be mounted X'10' LOG option Yes, list all control statements on SYSLST X'08' Dummy device search in progress; do not enter ERP X'04' Reserved X'02' Relocating Load option yes, Relocating Loader supported (always set) X'01' ASCII option yes, ASCII supported
56	38	Job Control Byte X'80' Job accounting Interface (JA) not supported X'40' Return to caller on LIOCS disk open failure X,20' Job control input from SYSRDR X'10' Job control output on SYSLOG X'08' Cancel job X'04' Pause at end-of-job step X'02' SYSLOG is a console printer-keyboard or DOC (always on) X'01' SYSLOG is assigned to the same device as SYSLST
57	39	Linkage Control Byte X'80' SYSLNK open for output X'40' Update of Second Level Directory and RAS loadlist in progress (interface between $MAINDIR and Supervisor) X'20' Allow EXEC X'10' Catalog linkage editor output X'08' Supervisor has been updated X'04' PCIL open in progress X'02' Update of System Core Image Library in progress (interface between $MAINDIR and Supervisor) X'01' Reserved

Expansion of COMREG Flag Bytes

Byte		Description
Dec	Hex	
58	3A	Language processor control byte. This is a set of switches used to specify nonstandard language translator options. The switches within the byte are controlled by job control OPTION statements and when set to 1, override standard options. The format of this byte is identical to the standard option byte (displacement 54) with one exception: Bit 7 in this byte is used to indicate to LIOCS that the rewind and unload option has been specified.
59	3B	Job Duration Indicator Byte X'80' Job in progress X'40' Dump or partition dump on an abnormal EOJ condition (see byte 141) X'20' Pause at EOJ step. Set by attention routine for job control X'10' Job control output on SYSLST X'08' Job is being run out of sequence with a temporary assignment for SYSRDR X'04' PCIL is being condensed X'02' //DATE statement processed for current job X'01' Batch command just issued
60	3C	Reserved
62 . . 76	3E 4G	Addresses of I/O tables as illustrated
78	4E	Set to the value nn specified in the LINES=nn parameter of the STDOPT statement. Initial value = 56.
79	4F	The format of the system date contained within this field is determined by the IPL program from information supplied in the date convention bit (displacement 55). Bytes 85–87 contain the day count.
88	58	Bytes reserved for use by LIOCS. Transient dump programs insert a key to indicate to the LIOCS End-of-Volume routine, $$BCMT07, that it was called by B-transient.
90	5A	Address of the first part of the Program Information Block (PIB) table.
92	5C	ID number of the last checkpoint (hexadecimal).
93	5D	Bytes 92 and 93 are also the temporary indicator of file protected DASD. Bits 0–15 correspond to channels 0–15. A bit ON means DASDFP for that channel.
94	5E	Job zone for Time-of-Day. If ZONE=EAST, value is positive, if ZONE=WEST, value is negative.
96	60	Address of disk I/O position data. This is the starting address of the Disk Information Block (DIB) table for the partition.
98	62	Device flag for automatic close. Bit 0: One or more 3800 printer extended buffering DTFs is/are open.
99	63	Flag byte X'80' = BTAM in the partition.
100	64	PC table
102	66	IT table
104	68	OC table
106	6A	PIK of partition
108	6C	Reserved
110	6E	Logical Transient Key (LTK) contains the same value as the PIK (Displacement 46) when the logical transient is requested. When the transient area is not in use, LTK is equal to zero. The SVC 2 routine sets the LTK. The SVC 11 routine resets the LTK. (Only significant in BG communication region.)
112	70	Address of SYSPARM field.
116	74	Address of Job Accounting partition table.
120	78	Address of Time-of-Day clock common area.
124	7C	Address of second part of Program Information Block (PIB) table.
126	7E	Address of PDTABB, table of DTF addresses for MICR support.
128	80	Slot for pointer to label area.

Expansion of COMREG Flag Bytes

Byte Dec	Byte Hex	Description
132	84	Address of background communications region.
134	86	Option Indicator Byte X'80' Reserved X'40' EU interface active X'20' Teleprocessing request X'10' Supervisor support for tape, always on X'08' Reserved X'04' Reserved X'02' Reserved X'01' GETVIS function has been initiated
135	87	System Configuration Byte 2 and RMSR Open Flag Byte X'80' PCIL supported, always on X'40' TOD clock supported, always on X'20' PFIX macro supported, always on X'10' Fetch $$BOPEN by $JOBCTLJ X'08' Fetch $$BOPEN by $JOBCTLD X'04' Fetch $$BOPEN by $JOBBCTLJ for WTM X'02' Reserved X'01' RPS supported
136	88	Reserved for compatibility reasons.
140	8C	Standard Job control option byte Initial value: X'80' EDECK Standard Option 0 X'40' ALIGN Standard Option 1 X'20' PARTDUMP Standard Option 0 X'10' RLD Standard Option 0 X'08' SXREF STD Option 0 X'04' TERM STD Option 0 X'02' Reserved – X'01' ACANCEL Standard Option 0
141	8D	Temporary Job control option byte X'80' EDECK Temporary Option X'40' ALIGN Temporary Option X'20' PARTDUMP Temporary Option X'10' RLD Temporary Option X'08' SXREF Temporary option X'04' TERM Temporary option X'02' SUBLIB=DF Temporary Option X'01' 1=ACANCEL Temporary Option
142	8E	Disk Configuration Byte 0–3 Reserved X'08' 3350 supported X'04' 3340 supported X'02' 3330 supported X'01' 2311 and 2314/2319 supported.
143	8F	Cataloged Procedure Name
151	97	Interface Byte for Cataloged Procedures X'80' Procedure being executed X'40' Overwrite processing X'20' Procedure with data X'10' Overwrite request for Job Control X'08' Insert request for Job Control X'04' Procedure end X'02' SYSLOG procedure X'01' Overwrite request for Supervisor
152	98	ICL statement name for Cataloged Procedure

Expansion of COMREG Flag Bytes

Byte Dec	Byte Hex	Description
159	9F	SYSIN 81 Bytes Indicator X'80' Permanent 81 bytes on SYSRDR X'40' Permanent 81 bytes on SYSIPT X'20' Temporary 81 bytes on SYSRDR X'10' Temporary 81 bytes on SYSIPT Reserved X'01' Allow/& for MAINT CATALS
160	A0	Pointer to POWER/VS partition control block.
164	A4	POWER/VS Flag Byte 1 X'80' POWER/VS Accounting Supported X'40' Partition under control of POWER/VS X'20' POWER/VS Partition X'10' Reserved X'08' Partition is waiting for work Reserved
165	A5	POWER/VS Flag Byte 2 Reserved
166	A6	Reserved
168	A8	Pointer to LUB Table Extension
172	AC	Job Control Switch 5 Bit 0: 1= EXEC LNKEDT statement is to be generated 1: Reserved 2: Reserved 3: 1=NEWVOL ignored 4: Reserved 5: Reserved 6: Reserved 7: 1=Job Control active
173	AD	Job Control Switch 6 0–6 Reserved 7: 1=Fetch search sequence changed
174	AE	Standard Job Control Option 2 (Reserved)
175	AF	Temporary Job Control Option 2 (Reserved)

PARTITION IDENTIFICATION KEY (PIK)

Task or Partition	PIK value in BGCOMREG			
	Number of partitions supported			
	5	4	3	2
Attention	X'00'	X'00'	X'00'	X'00'
BG	X'10'	X'10'	X'10'	X'10'
F4	X'20'			
F3	X'30'	X'20'		
F2	X'40'	X'30'	X'20'	
F1	X'50'	X'40'	X'30'	X'20'

TASK INTERRUPT KEY (TIK)

Task	TIK value			
	Number of partitions supported			
	5	4	3	2
Attention	X'00'	X'00'	X'00'	X'00'
BG	X'10'	X'10'	X'10'	X'10'
F4	X'20'			
F3	X'30'	X'20'		
F2	X'40'	X'30'	X'20'	
F1	X'50'	X'40'	X'30'	X'20'
Subtask*	X'60'–X'F0'	X'50'–X'F0'	X'40'–X'F0'	X'30'–X'F0'

* If AP=YES

LOGICAL TRANSIENT OWNER IDENTIFIKATION KEY (LIK)

The halfword LIK at displacement 88 in SYSCOM contains the same value as the TIK when the Logical Transient Area (LTA) is in use and therefore identifies the owner of the LTA. When LTA is free, the halfword LIK contains zeros. The SVC2 routine sets the LIK, and the SVC11 routine resets it to zero.

LOGICAL TRANSIENT KEY (LTK)

The halfword LTK at displacement 110 in each partition communication region has a zero value in the high-order byte and a key value in the low-order byte. In a foreground communication region, the key value in the LTK is not significant. The LTK in the background communication on region (BGCOMREG) has the same value as the PIK of partition of the task that owns the LTA, or contains zeros when the LTA is free. The SVC2 routine sets the LTK, and the SVC11 routine resets it to zero.

RASLINK

Hex / Dec / Displacement	CPU ID Field	Damaged Channel Byte	RAS Flag Byte	Machine Check Flags	Reserved	RAS Table (RASTAB) Address	Base Address for RAS Monitor	Internal Model Number	Length of I/O Extended Logout Area	Address of extended mach. check LOGOUT area
0 0		8 8	9 9	0A 10	0B 11	0C 12	10 16	14 20	16 22	18 24
	xxxxxxxx	x	x	x	x	xxxx	xxxx	xx	xx	xxxx

Key to RAS Linkage Area displacement

0 CPU ID Field

8 Address of damaged channel, or X'FF' if no channel damaged

9 RAS Flag Byte

bit	flag	description
0	X'80'	RAS active
1	X'40'	RAS SIO flag
2	X'20'	RTA in control
3	X'10'	RAS I/O delayed
4	X'08'	Channel check on error SIO
5	X'04'	Reserved
6	X'02'	Channel check on SIO
7	X'01'	I/O active for SIO

10 Machine Check Flags

bit	flag	description
0–4		Reserved
5	X'04'	Hard machine check
6	X'02'	All machine records built
7	X'01'	All channel check records built

11 Reserved

12 Address of RAS Monitor Table (RASTAB)

16 Address for base register in RAS Monitor Program

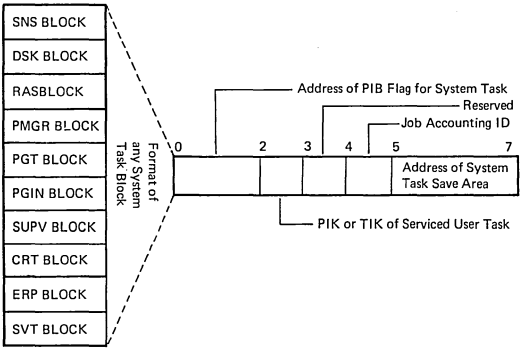
20 Internal Model Number

22 Length of I/O extended logout area

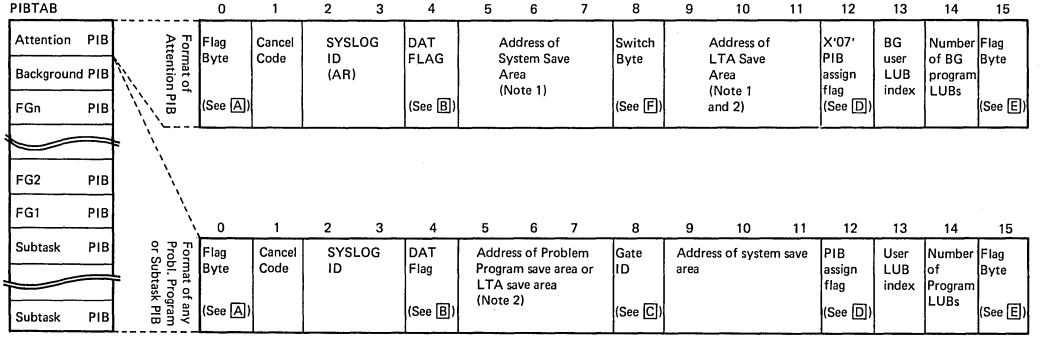
24 Address of machine check extended LOGOUT area (if byte 0 = X'80', address not yet valid)

Bytes 112–115 (X'70'–X'73') of the System Communication Region (SYSCOM) contain the address of the area. Label RASLINK identifies the first byte of the area.

SYSTEM TASK BLOCKS



Notes:
Bytes 152–155 (X'98'–X'9B') of the System Communication Region (SYSCOM) contain the address of the first System Task Block.



Note 1: When the LTA is active for logical attention the contents of bytes 5–7 and bytes 9–11 of the attention PIB are exchanged.

Note 2: When the LTA is active for problem programs the contents of byte 9–11 of the attention PIB and byte 5–7 of the problem program PIB are exchanged.

Byte 90–91 (X'5A'–X'5B') of the partition communication regions contain the address of the first part of the PIB table. Label PIBTAB identifies the first byte of the table.

A Flag Byte (First Byte in PIB)
Flags which are always used:

X'65' Task is waiting for free page frames
X'67' Task is waiting for IDRA
X'69' Task is waiting for ALLOCATE routine
X'6B' Task is waiting for SETLIMIT routine
X'6D' Task is waiting for GETVIS routine
X'71' Program is waiting for GETREAL
X'73' Program is waiting because system is seized
X'75' Program is waiting for copy block
X'77' Program is waiting for TFREE
X'79' Program is waiting for channel queue entry
X'7B' Program is waiting for CCW translation
X'7D' Program is waiting for AVR information
X'7F' Program is waiting for XECB table
X'80' Program is not active
X'81' Program is SVC2 bound (waiting for the LTA to be released)
X'82' Program is SVC7 bound (waiting for I/O interruption)
X'83' Program is ready to run
X'85' Program is SVC5 bound (waiting for the PTA to be released)
X'86' Initial selection of RAS (used only for RAS PIB flag)
X'87' Program is set to common bound condition
X'89' Program is terminator bound (waiting for terminator to be released)
X'8B' Task is waiting for USE/RELEASE routine
X'8D' Task is waiting for subsystem identifier routine
X'8F' Extent bound/task is waiting for ext. routine
X'91' CIL bound
X'93' Task is waiting for SDL update to be completed

Flags used only under certain conditions:

Flags with partition-dependent values:

The table below shows the various possible values of these flags and the partition to which a given value refers, depending on the number of partitions. The meanings of the flags types A, B, C, and D are explained below the table.

Flag				Partition referenced			
				NPARTS =			
A	B	C	D	2	3	4	5
X'27'	X'35'	X'43'	X'55'	BG	BG	BG	BG
X'29'	X'37'	X'45'	X'57'	F1	F2	F3	F4
X'2B'	X'39'	X'47'	X'59'		F1	F2	F3
X'2D'	X'3B'	X'49'	X'5B'			F1	F2
X'2F'	X'3D'	X'4B'	X'5D'				F1

Flag A The codes are used by the PFIX routine to set a partition PFIX bound

Flag B The codes are used to gate the CDLOAD routine for tasks running in one partition.

Flag C The codes are used to gate the GETVIS routine for tasks running in one partition.

Flag D The codes are used by the load leveler to deactivate a partition .

Flags with partition-independent values:

X'62' Program is SVC35-bound
X'63' Program is SVC35-bound } only if TRKHLD=n
X'6D' Program is waiting for the next freed page frame.
X'51' Program is SVC38-bound
X'53' Program is SVC41/42-bound

The following flag is used only if CBF=n:
X'6F' Program is waiting for free console buffer table entry

The following flag is used only if DASD File Protect is supported:
X'8F' Task is waiting for extent routine

B PIB DAT FLAG

X'01' Return to reentrant supervisor routine
X'02' Return to gated supervisor routine
X'04' Move CCB/IORB at dispatching time
X'08' Service delayed external interrupt
X'10' Deactivation for this task is being delayed
X'20' Operator communications to be serviced
X'40' Task has seized the system
X'80' Program is running in virtual mode

C Gate Identifier

X'71' Gating of SVC 58 required
X'53' Gating of SVC41/42 required

The flags are only used if PIB DAT Flags is X'03', that is, Bits 6 and 7 are on. (See B).

D PIB Assign Flag

X'80' SYSRES DASD file protect inhibited (allow write operation on SYSRES)
X'40' Channel appendage exit allowed (BTAM, POWER/VS)
X'20' Cancel occured in LTA
X'10' Cancel control (set on a foreground cancel)
X'08' Hold foreground assignments
X'04' Do not ready task for cancel
X'02' Terminator finished (fetch end-of-job monitor)
X'01' Privileged status indicator (either the LTA or the SVA resident terminator is active for this task)

E Problem Program PIB Flag (Last Byte in PIB)

Bit 0: 1 = Return to terminator
Bit 1: 1 = Cancel in LTA and device not assigned
Bit 2: 1 = /& on SYSIN
Bit 3: 1 = Partition in stopped state
Bit 4: 1 = Initialize Terminator
Bit 5: 1 = Task is terminating
Bit 6: 1 = Maintask: Subtask attached
Subtask: Cancel all requester
Bit 7: 1 = Task in AB routine is active

F Attention PIB Switch Byte

Bit 0: Reserved
Bit 1: 1 = Fetch Physical Attention Transient $$ABERRZ
Bit 2: Reserverd
Bit 3: 1 = Emergency cancel request
Bit 4: Reserved
Bit 5: 1 = Command available (DOC)
Bit 6: 1 = Fetch Logical Attention Routine ($$BATTNA)
Bit 7: 1= External Interrupt request

PIB2TAB

Attention PIB
Background PIB
FGn PIB
FG2 PIB
FG1 PIB
Subtask PIB
Subtask PIB

Format of any PIB Extension entry

0 1	2 3	4 5 6 7	8 9 10 11	12 13	14	15
Address of communication region (Note 1)	System LUB index	Interrupt Information (See table Ⓐ below)	Address of termination ECB if any, otherwise zeros	Program Interrupt Key (PIK)	unused	Flag Byte Ⓑ

Note 1: Always BG communication region in Attention- and Background PIB extension. Appropriate communication region in other PIB extensions. To place this address in a register instruction ICM should be used.

For each PIB Table entry (first part) an entry exists in the PIB Table Extension (second part).

Bytes 124–125 (X'7C'–X'7D') of the partition communication region(s) contain the address of the PIB Table extension. Label PIB2TAB identifies the first byte of the PIB Table extension.

Ⓐ

Type of interruption	C Contents of PIB Extension Bytes			
	4	5	6	7
SVC	00	ICL *	Interruption Code	
PC	00	ICL *	Interruption Code	
I/O	00	00	I/O Address	

* ICL (Instruction Length Code) is in bits 5 and 6, other bits are zeros.

Ⓑ

Byte 15	
Bits 0–1: Reserved	
Bit 2:	1 = Task owns CRT
Bit 3:	VSAM automatic close in progress
Bit 4:	Not used
Bit 5:	1 = SVC screening
Bit 6:	Reserved
Bit 7:	XECB: Task issued SVC 92, 93, or 94

SAVE AREAS

Layout of LTA and Problem Program Save Area

Note 1		PSW		Reg 9	Reg A
Reg B	Reg C	Reg D	Reg E	Reg F	Reg 0
Reg 1	Reg 2	Reg 3	Reg 4	Reg 5	Reg 6
Reg 7	Reg 8	Note 2		FLTPT R0	
FLTPT R2		FLTPT R4		FLTPT R6	

Note 1: Problem Program Save Area: Program Name
LTA Save Area: Transient Name

Note 2: Problem Program Save Area: Byte 0, 1: Reserved
Bytes 2–7: Job Start Time in TOD Format

LTA Save Area: Reserved

Layout of User Save Area (for AB, PC, OC, IT, and TT Routines)

Interrupt Status Information *		Reg 0	Reg 1	Reg 2	Reg 3
Reg 4	Reg 5	Reg 6	Reg 7	Reg 8	Reg 9
Reg A	Reg B	Reg C	Reg D	Reg E	Reg F

* Interrupt Status Information:

Byte 0: Reserved

Byte 1: Protection key and mask (from PSW byte 1)

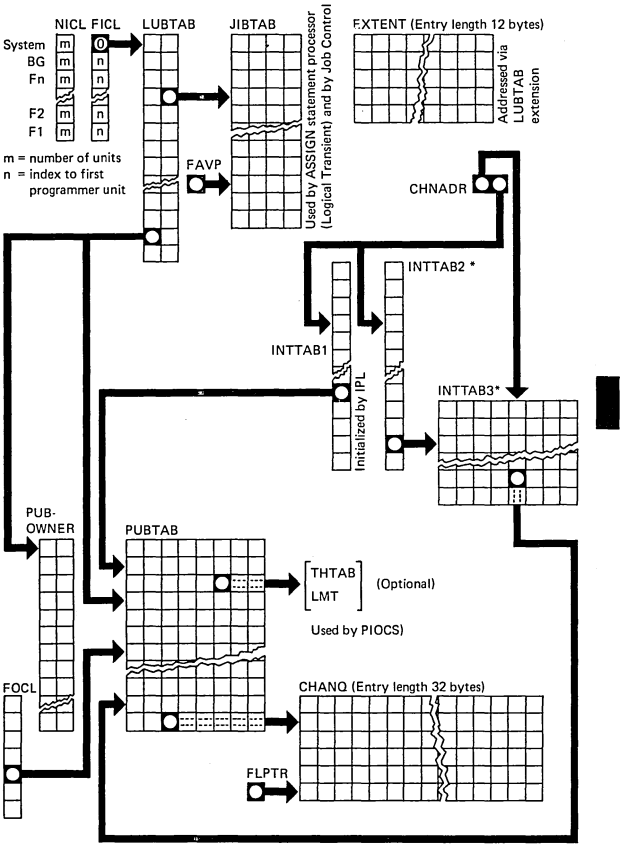
Bytes 2, 3: Interruption Code

Byte 4: Bits 0–1 Instruction Length Code
Bits 2–3 Condition Code
Bits 4–7 Program Mask

Bytes 5–7: Instruction Address

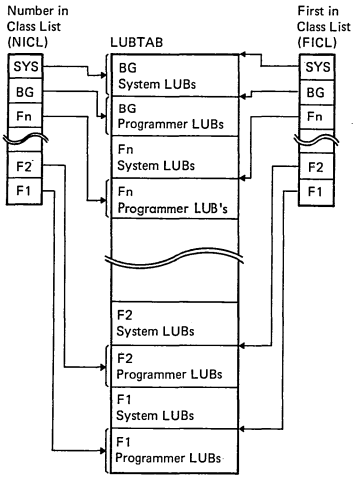
Byte 8: General Register 0–15

I/O TABLE INTERRELATIONSHIP



* Optionally allocated and initialized by IPL.

LOGICAL UNIT BLOCK (LUB) TABLE



0 0 0 0 0 0 0 0 — Points to first PUB and PUBOWNER
0 0 0 0 0 0 0 1 — Points to second PUB and PUBOWNER
0 0 0 0 0 0 1 0 — Points to third PUB and PUBOWNER
.
.
.
1 1 1 1 1 1 1 0 — Ignore, assigned ignore
1 1 1 1 1 1 1 1 — Null Pointer, the LUB is unassigned

When a logical unit is assigned, the system inserts a pointer to the PUB for the phyisical device specified.

LUB Table for any Partition

- SYSRDR
- SYSIPT
- SYSPCH
- SYSLST
- SYSLOG
- SYSLNK
- SYSRES
- SYSSLB
- SYSRLB
- 1) SYSUSE
- SYSREC
- SYSCLB
- Reserved
- SYSCAT
- SYS000
- SYS001
- SYS002
- SYS003
- SYS004
- 2) SYSnn

Format of any LUB

Byte 0	Byte 1
PUB Pointer	JIB Pointer

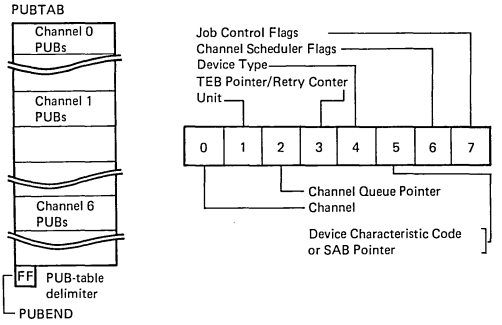
JIB Index (Multiply by 4 = Displacement into JIB Table) or X'FF' = Null Pointer, no JIB for this LUB.

A LUB has a JIB pointer when:
1. The logical unit is temporarily assigned.
2. The logical unit assignment is alternate (ALT).
3. A DASD file (exept a system I/O file on disk) is opened (DASD file protect only).

Bytes 76 and 77 (X'4C'–'4D') of the partition communication region contain the address of the LUB table. Label LUBTAB identifies the first byte of the table.

Note 1: SYSUSE may be called SYSCTL in error recovery messages.

PHYSICAL UNIT BLOCK (PUB) TABLE



Byte 0: Channel number (Hex 0–6, FF = NULL)

Byte 1: I/O device unit number

Byte 2: Hex 0, 1, 2, ... points to the first channel queue entry for this device

Byte 3: This byte is an ERP retry counter

Byte 4: Device type code

Byte 5: SS of the MODE = parameter in the DVCGEN macro for tape unit. (See section 2) For an ICA line Model 115 or 125, this byte contains the displacement index of the entry in the Line Mode Table (LMT). The address of the LMT is in SYSCOM.

For DASD with track hold this byte contains a pointer to the track hold table or X'FF'.

For MICR type devices, this byte indicates which external interrupt line is in use.

For a 3704/3705 Communications Controller, this byte contains the type number of the Channel Adapter.

For 2560 or 5424/5425

Bit 0 1 = Repositioning required
1 0 = SYSPCH temporarily assigned to hopper 1
1 = SYSPCH temporarily assigned to hopper 2
2 0 = SYSIPT temporarily assigned to hopper 1
1 = SYSIPT temporarily assigned to hopper 2
3 0 = SYSRDR temporarily assigned to hopper 1
1 = SYSRDR temporarily assigned to hopper 2
5 0 = SYSPCH permanently assigned to hopper 1
1 = SYSPCH permanently assigned to hopper 2
6 0 = SYSIPT permanently assigned to hopper 1
1 = SYSIPT permanently assigned to hopper 2
7 0 = SYSRDR permanently assigned to hopper 1
1 = SYSRDR permanently assigned to hopper 2

For 3800
Bits 0 and 1
00 = 3800
01 = 3800 B
10 = 3800 C
11 = 3800 BC

PHYSICAL UNIT BLOCK TABLE (. . . Cont'd)

Byte 6: Channel Scheduler Flags

Bit	0	1 = Device busy
	1	1 = Switchable device
	2	1 = Unit check is pending on device
	3	1 = I/O error queued for recovery
	4	1 = Operator intervention outstanding
	5	1 = I/O interrupt outstanding
	6	1 = Burst or overrunable device
	7	1 = 7-track tape unit

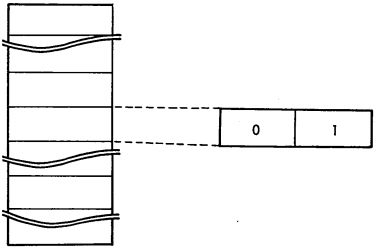
Byte 7: Job Control Flags

Bit	0–4	Standard MODE assignment for 7-track tape (all ones if not tape, all zeros if device is down)
Bit	5	1 = Device supports RPS
	6	Reserved
	7	Unit check pending on alternate channel

Notes: A null entry is generated at supervisor generation time for each device to be supported by the supervisor. Then standard physical unit assignments are made to the PUB table. Physical unit assignments can also be made during IPL. PUBs are ordered by channel and priority within a channel. A PUB entry must be generated or added during IPL for any device of the installation.

An entry in the PUB ownership table is associated with each entry in the PUB table.

PHYSICAL UNIT BLOCK OWNERSHIP TABLE



Byte 0: Bit 0 1 = Volume is owned by VTAM
1 1 = Waiting for volume to be mounted
Bit 2–7 Reserved

Byte 1: Identifies the partition that owns the PUB according to follwing table:

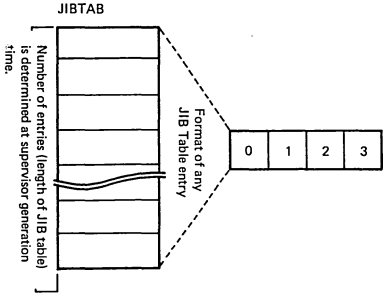
Flag	Partition owning PUB if number of partitions is:			
	2	3	4	5
X'00'	UA *			
X'01'	BG	BG	BG	BG
X'02'	F1	F2	F3	F4
X'04'		F1	F2	F3
X'08'			F1	F2
X'10'				F1

* unassigned

Note: The number of entries in the PUB Ownership table is equal to the number of entries in the PUB table. Associated with each PUB entry is an entry in the PUB Ownership table.

Bytes 120–123 (X'78'–'7B') of the system communications region (SYSCOM) contain the address of the PUB Ownership table. Label PUBOWNER identifies the first byte of the table.

JOB INFORMATION BLOCK (JIB) TABLE

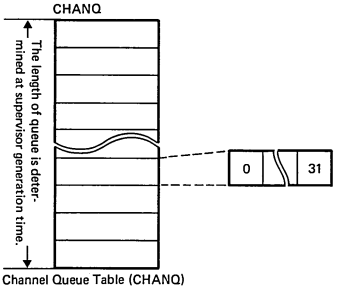


Byte(s)	Description	
0–1 (Contents depends on the bit setting in byte 2)	Bit setting Byte 2	Contents
	Bit 0 = 1 Stored standard assignment	LUB entry of stored standard assignment. (PUB and JIB pointer)
	Bit 1 = 1 Alternate assignment	Byte 0: PUB Pointer Byte 1: X'00'
2	Meaning if bit = 1 Bit 0: Stored standard assignment Bit 1: Alternate assignment Bit 4: The alternate assignment indicated in bit 1 is permanent. Bit 5: Cataloged procedure processing Bit 6/7: Reserved	
3	Chain byte Contains the displacement index of the next JIB. X'FF' defines the end of the chain.	

Bytes 68–69 (X'44'–'45') of the partition communication region contain the address of the JIB table entry. Label JIBTAB identifies the first byte of the table.

CHANNEL QUEUE TABLE (CHANQ)

Bytes 37–39 (X'25'–X'27') of the system Communication Region (SYSCOM) contain the address of the Channel Queue table. Label CHANQ identifies the first byte of the table.



Channel Queue Table (CHANQ)

The layout of an entry is as follows:

Bytes	Contents
0	Contains the displacement within the channel queue of the next entry in the free list or in the list for an specific device, or X'FF' when it is the last entry in the free list or a device list.
1–3	Contains the CCB/IORB address for the specified device.
4	Contains the storage protection key for the I/O request, in the form X'nk', where n = user storage protection key. (Attention or system task (or if the request is from the LTA) = 0, BG task = 1, FG task = 2–5, depending on the number of partitions). K = 0 for Attention and all user tasks. System tasks: X'01' SNS X'02' DSK X'03' RAS X'04' PMGR X'05' PGT X'06' PGIN X'07' SUPV X'09' CRT X'0B' ERP X'0F' SVT
5	Contains information on special processing that is required for this I/O request: X'80' Interrupt not yet processed X'40' Dequeue unconditional X'20' Reserved X'10' Console buffering request X'08' Reserved X'04' DASD file protect needed X'02' SYSFIL on CKD device X'01' SYSFIL on FBA device
6	Contains pointer (displacement index) to the LUB table, identifying the logical unit making the I/O request. This is doubled to get the actual displacement into the LUB table.
7	Contains the displacement within the PIB table of the PIB of the task requesting I/O.
8	Contains information on the conditions encountered at SIO-time. X'F0' Physical I/O information X'80' Device running X'40' Alternate channel I/O X'30' Reserved X'0F' SIO condition code X'08' Condition code is 0 X'04' Condition code is 1 X'02' Condition code is 2 X'01' Condition code is 3

CHANNEL QUEUE TABLE (. . . Cont'd)

9–11 Contains transmission information as passed from the user CCB/IORB (bytes 2, 3 and 12).

12 Reserved for page fixing routine.

13–15 Address of internal fix list as returned from page-fix routine.

16 Error retry count

17–19 Contains the address of the PUB2 entry.

20 Contains head queue related processing information:

X'80'	Reserved
X'40'	Head queue request
X'20'	Device busy status from PUB
X'10'	Queued-in-error from PUB

21 Contains information on the requestor issuing the I/O request:

X'04'	RAS retry request
X'FB'	Reserved

22 Contains information on the group of devices the device belongs to:

X'80'	CKD device or diskette
X'40'	FBA device
X'20'	Tape device
X'10'	Teleprocessing device
X'08'	2260 or 3277 device
X'04'	Unit record device
X'03'	Reserved

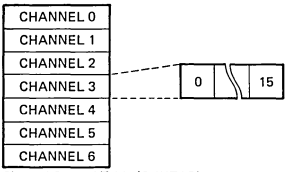
23 Contains information on where to continue I/O interrupt processing:

X'00'	Dispatcher (DISP)
X'04'	I/O initiator (INITRG)
X'08'	I/O interrupt handler (INTRIN)
X'0C'	I/O error IGNORE routine (IGNORE)
X'10'	CANCEL (ERR1A)
X'14'	CANCEL (ERR31)

24–31 Contain the accumulated I/O interrupt information.

CHANNEL CONTROL TABLE

CHANTAB



Channel Control Table (CHNTAB)

Bytes

0

X'10'	Byte Multiplexer Channel
X'11'	Byte Multiplexer Channel with burst mode support
X'13'	Byte Multiplexer Channel running in burst mode
X'20'	Block Multiplexer Channel
X'00'	Selector Channel
X'80'	Channel not operational or not present to the system

1 Number of unit checks pending on this channel

2 Flag Byte

X'80'	Channel must be restarted
X'7F'	Reserved

3 Reserved

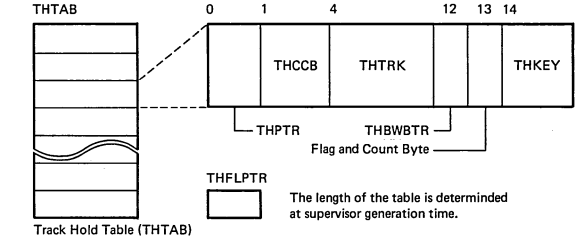
4–7 Address of first PUB on channel

8–11 Address of last PUB started on channel

12–15 Address of PUB that needs channel exclusively

Bytes 60–63 (X'3C'–X'3F') of the System Communication Region (SYSCOM) contain the address of the Channel Control Table. Label CHNTAB identifies the first byte of this table.

TRACK HOLD (THTAB) TABLE



Track Hold Table (THTAB)

Bytes

0 Pointer to next entry in chain (forward pointer). All table entries belonging to one PUB are chained together. The chain delimiter is X'FF'.

1–3 CCB/IORB address

4–11 For CKD devices: Address of the track held, in the form BBCCHHOO.
For FBA devices: Physical block number of first and last block of the range held.

12 Pointer to previous entry (backward pointer). The first entry of the chain contains the PUB index.

13 Flag and count byte:

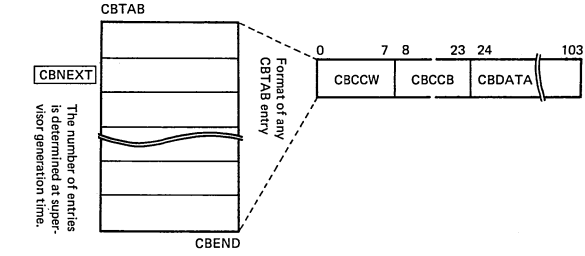
Bit 0: A task is waiting for this track or range of blocks.
Bit 1: First entry of queue. Byte 12 contains PUB index.
Bit 2, 3: Reserved
Bit 4–7: Hold count: the number of holds is one more than this value.

14, 15 Key of task owning this entry.

Bytes 77–79 (X'4D'–X'4F') of the System Communication Region (SYSCOM) contain the address of the Track Hold table. Label THTAB identifies the first byte of the table.

Byte 76 (X'4C') of the System Communication Region (SYSCOM) contains the address of the Track Hold Free List Pointer. Label THFLPTR identifies the location of the pointer.

CONSOLE BUFFERING TABLE



CBNEXT points to next buffer entry

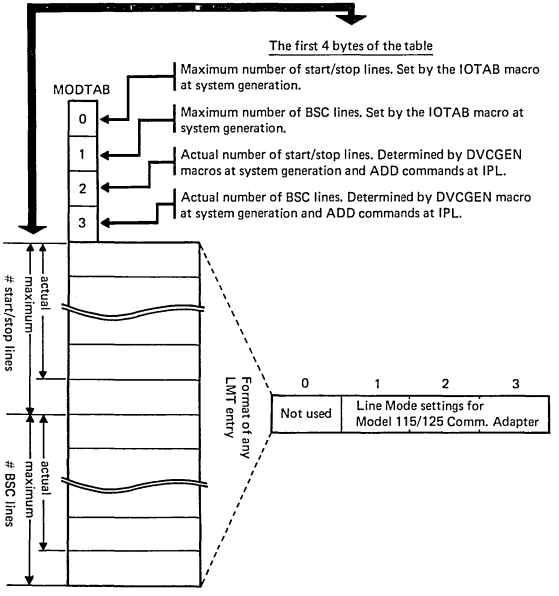
Bytes 0–7 CCW:
Command code, chain byte, and count field copied from the user's CCW. The data address is always the address of the data portion of the buffer entry.

Bytes 8–23 CCB/IORB:
The CCB/IORB used by the console buffering routine. The CCW address is always the address of the CCW in the buffer entry.

Bytes 24–103 Contains the data moved from the requestor's output area.

Label CBTAB identifies the first byte of the Console Buffer.

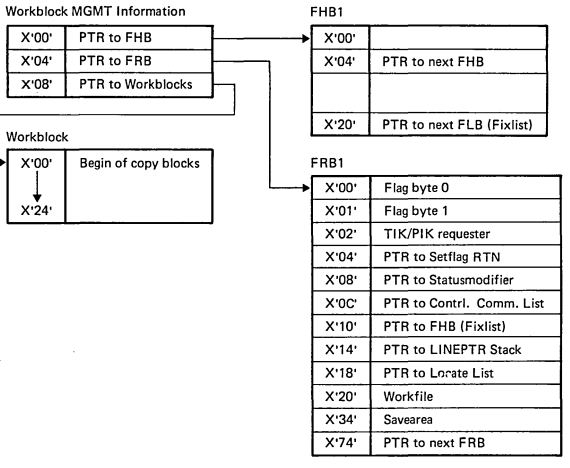
LINE MODE TABLE



Bytes 140–143 (X'8C'–X'8F') of the System Communication Region (SYSCOM) contain the address of the table. Label MODTAB identifies the first byte of the table.

This table is required by the models 115/125 only and is used to save the TP mode at IPL time.

RELATIONSHIP OF CONTROL AND WORKBLOCKS FOR CHANNEL PROGRAM FIXING



LAYOUT OF FIXLIST HEADER BLOCK

FHB1

X'00'	Flag Byte 1 (General Fixing Function)	Flag Byte 2 (Reserved)	TIK/PIK of Requestor
X'04'	Pointer to next active FHB		
X'08'	BA1		EA1
X'0C'	BA2		EA2
X'10'	BA3		EA3
X'14'	BA4		EA4
X'18'	BA5		EA5
X'1C'	BA6		EA6
X'20'	Pointer to next fixlist block		

Note:
Bytes 0–3 (X'00'–X'03') of the Workblock Management Information contain the address of the FHB1.

Layout of Fixlist Header Block (FHB) for General Fixing Function

Flag Byte 1 (General Fixing Function):

Bit 0 = 1 Fixing function request complete.
Bit 1 = 1 At least one page is fixed for this task or the fixing request is pending.
Bit 2 = 1 Fixing of pages required.
Bit 3–7 Reserved

FHBn

X'00'	Flag Byte 1 (General Fixing Function)	Flag Byte 2 (Fast Fixing Support)	TIK/PIK of Requestor
X'04'	Saved queue forward pointer		
X'08'	Saved queue backward pointer		
X'0C'	Pointer to replica or zero		
X'10'	Pointer to next active FHB		
X'14'	BA1		EA1
X'18'	BA2		EA2
X'1C'	BA3		EA3
X'20'	Pointer to next fixlist block		

Note:
Bytes 16–19 (X'14'–X'17') of the Fix Request Block (FRB) contain the address of the Fixlist Header Block (FHB).

Layout of Fixlist Header Block (FHB) for Fast Fixing Support

Flag Byte 2 (Fast Fixing Support):

Bit 0 = 1 Fast fixing in progress.
Bit 1 = 1 FHB belongs to saved FHB queue.
Bit 2–7 Reserved

FIX REQUEST BLOCK

FRB1

X'00'	Flag byte 0
X'01'	Flag byte 1
X'02'	TIK/PIK requester
X'04'	PTR to Setflag RTN
X'08'	PTR to Statusmodifier
X'0C'	PTR to Contrl. Comm. List
X'10'	PTR to FHB (Fixlist)
X'14'	PTR to LINEPTR Stack
X'18'	PTR to Locate List
X'20'	Workfile
X'34'	Savearea
X'74'	PTR to next FRB

Note:
Bytes 4–7 (X'04'–X'07') of Work-block MGMT Information contain the address of Fix Request Block (FRB).

Fix request Block Example

Flag Byte 0:

Bit 0 Data chaining specified
Bit 1 READ/SENSE command
Bit 2 READ BACKWARD command
Bit 3 Status modifier command (STM) and data chaining
Bit 4 Status modifier command (STM)
Bit 5 Status modifier handling in process
Bit 6 Reserved
Bit 7 DOIO request (Routine CCWDOIO)

Flag byte 1 (Fast Fixing Support):

Bit 0 = 1 Replica creation required
Bits 1–7 Reserved

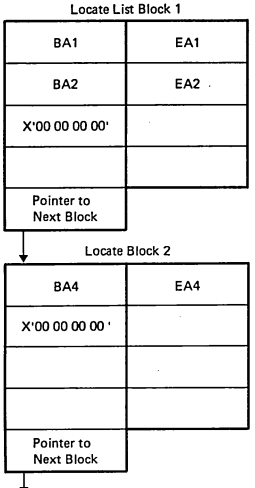
LAYOUT OF FIXLIST BLOCK (FLB)

X'00'	BA1	EA1
X'04'	BA2	EA2
X'08'	BA3	EA3
X'0C'	BA4	EA4
X'10'	BA5	EA5
X'14'	BA6	EA6
X'18'	BA7	EA7
X'1C'	BA8	EA8
X'20'	Pointer to next fixlist block or zero	

Note:
Bytes 4–8 (X'04'–X'08') of the Fixlist Header Block (FHB) contain the address of the Fix-list Block (FLB).

Layout of Fixlist Block (FLB)

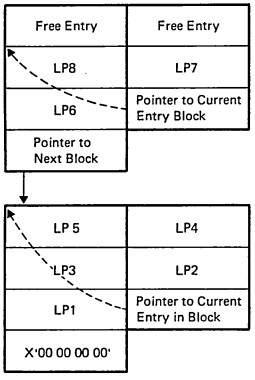
LAYOUT OF LOCATE LIST BLOCK



Note:
Bytes 24–27 (X'18'–X'1F') of FRB contain the address of Locate List Block 1.

Locate List Block Exampel

LAYOUT OF LINE POINTER BLOCKS

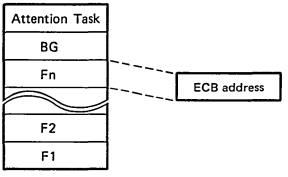


Note:
Bytes 20–23 (X'14'–X'17') of FRB contain the address of Line Pointer Block.

Line Pointer List Example

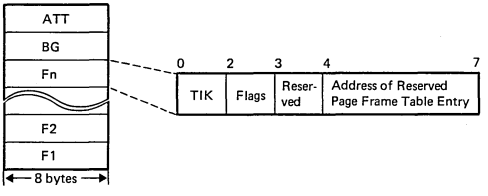
EMULATOR ECB TABLE (EUECBTAB)

EUECBTAB



Bytes 104–107 (X'68'–X'6B') of the System Communications Region (SYSCOM) contain the address of the Emulator ECB Table. Tabel EUECBTAB identifies the first byte of the table.

FIXWTAB



Bytes

0, 1: TIK of task issuing the PFIX request; inserted when the PFIX request is started.

2: Flag byte. Set when the PFIX request is started.

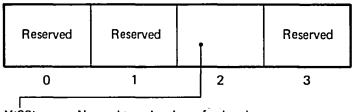
X'00' = PFIX request
X'40' = PFIX restart request
X'80' = GETREAL request

DENSITY DATA

Density (Bytes per inch)	Parity	Convert Feature	Translate	SS Code *
200	odd	on	off	10
200	odd	off	off	30
200	odd	off	on	38
200	even	off	off	20
200	even	off	on	28
556	odd	on	off	50
556	odd	off	off	70
556	odd	off	on	78
556	even	off	off	60
556	even	off	on	68
800	odd	on	of	90
800	odd	off	off	B0
800	odd	off	on	B8
800	even	off	off	A0
800	even	off	on	A8
800	dual density nine-track----------			C8
1600	dual density nine-track----------			C0
6250	dual density nine-track----------			D0
800	single density nine-track----------			C0
1600	single density nine-track----------			C0
6250	single density nine-track----------			

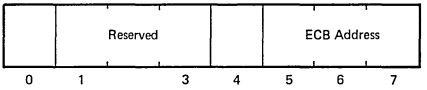
* Refer to PUB Table, byte 5

EVENT CONTROL BLOCK (ECB)



X'80' Normal termination of subtask
X'C0' Abnormal termination of subtask

RESOURCE CONTROL BLOCK (RCB)



Byte 0 X'FF' if resource is in use, X'00' if resource is not in use
Bytes 1–3 Reserved
Byte 4 Bit 0 = 1 Another task waiting for the resource
Bit 1 = 1 No other task waiting for the resource
Bytes 5–7 ECB address of current resource owner

CROSS PARTITION ECB (XECB) TABLE

XECB name	Access code	XECB address	Task ID of owner	Task ID of waiter/ poster

0 7 8 9 11 12 13 14 15

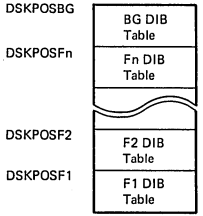
Byte 8

Parameterlist for TYPE = DEFINE

Bytes 0–7 XECB name
Byte 8 X'0n', bits 4–7 contain the access code

Paramterlist for TYPE = DELETE or DELETALL or RESET or CHECK

Bytes 0–7 XECB name
Bytes 8–9 X'1000' TYPE = RESET
X'4000' TYPE = DELETE
X'4800' TYPE = DELETALL
X'8000' TYPE = CHECK



The number of DIB tables depends on the number of partitions specified at supervisor generation.

Format of any DIB table if SYSFIL = YES

	0						6	7		9	10						16	17	18	19	20	21	22	23
	Current Address							K	D	D	End Address							UL	LL	M	RC			
SYSLNK	C	C	H	H	00	00	00	P	P	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
SYSIN	B	B	C	C	H	H	R	00	00	50	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00
SYSPCH	B	B	C	C	H	H	R	00	00	51	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00
SYSLST	B	B	C	C	H	H	R	00	00	78	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00
PRCDIB	B	B	C	C	H	H	R	00	00	50	B	B	C	C	H	H	R	H	H	XX	XX	XX	00	00

Format of any DIB Table if SYSFIL = NO

	0						6	7		9	10						16	17	18	19	20	21	22	23
	Current Address							K	D	D	End Address							UL	LL	M	RC			
SYSLNK	C	C	H	H	00	00	00	P	P	00														
PRCDIB	B	B	C	C	H	H	R	00	00	50	B	B	C	C	H	H	R	H	H	Xx	XX	XX	00	00

Format of 3540 Diskette DIB Table

	0						6	7		9	10						16	17	18	19	20	21	22	23
	Current Address							K	D	D	End adress									M	RC			
SYSIN	00	00	00	00	C	H	R	00	00	00	FLG	EXT #	HDR #	00	C	H	R	00	00	XX	00	00	00	00
SYSPCH	00	00	00	00	C	H	R	00	00	00	FLG	EXT #	HDR #	00	C	H	R	00	00	XX	00	00	00	00
SYSLST	00	00	00	00	C	H	R	00	00	00	FLG	EXT #	HDR #	00	C	H	R	00	00	XX	00	00	00	00

Bytes 0–6 Current address
Bytes 0–5 Current address of key: the next address to be used (both for input and output).
Byte 6 Record number of current address.

Bytes 7–9 KDD
Key and data length of the symbolic device.
PP: starting cylinder of private core image library if PCIL is assigned; otherwise zero.

Bytes 10–16 End address
Bytes 10–15 End address of key: the last address within the limits of the extent.
16 Record number of end address.

Byte 17 UL Upper head limit

Byte 18 LL Lower head limit

Byte 19 M Maximum number of records per track.

Byte 20–21 RC Record count: residual capacity for beginning of operator notification. Set a system generation time with SYSFIL parameter, or after IPL with SET statement (RCLST and/or RCPCH operands).
A warning message is issued by job control after end-of-job step when the minimum number of remaining records has been reached or exceeded during the previous job (not supported for 3504).

Byte 22 X'40' Indicates RPS support

Byte 23 Reserved

Label DSKPOSBG identifies the first byte of the BG DIB table.

The address of the DIB tables are contained in bytes 96 and 97 (X'60'–X'61') of the appropriate partition communication region.

DIB Table for FBA devices

Byte(s)	Label	Description
0–3	ULPBN	End addres of extent. Upper limit of physical block number
4–7	CRPBN	Current address. Current physical block number
8, 9	CIOFF	Offset of current record within control interval
10, 11	LNGCI	Length of control intervals in bytes
12	PBPERCI	Number of physical blocks per control interval
13–15	PBUFFER	Pointer to data buffer
16	DIBFLAGS	X'01' Buffer-in-use flag X'02' End of extent reached X'04' Force write out X'10' Source begin readjustment required X'40' Task waiting for DIB X'80' DIB gage flag
17–19	PDIBX	Pointer to DIB extension (DIBX).
20, 21	DIBRSCNT	Residual count for JCL message
22, 23		Reserved

A FBA device also requires a DIB extension (DIBX) table. It has the following format:

Byte(s)	Description
0–23	IORB
24–31	Fixlist first area
32–39	Fixlist second area
40–47	DEFINE EXTENT CCW
48–55	LOCATE CCW
56–63	READ/WRITE CCW
64–79	Parameter field for DEFINE EXTENT CCW
80–87	Parameter field for LOCATE CCW

Disk Information Block Extension Table (DIBX) for FBA Devices

TABLES FOR MICR DTF ADDRESSES AND POINTERS

The table of DTF addresses (PDTABB) contains six 8-byte entries; one for each line of the direct control feature on the system.

PDTABB

Byte	0	1	2	3	4	5	6	7
	AND instruction				Owner ship Flags	DTF address for MICR		
0	NI	PDSTAT+1		X'FE'		Device on line 7		
8	NI	PDSTAT+1		X'FD'		Device on line 6		
16	NI	PDSTAT+1		X'FB'		Device on line 5		
24	NI	PDSTAT+1		X'F7'		Device on line 4		
32	NI	PDSTAT+1		X'EF'		Device on line 3		
40	NI	PDSTAT+1		X'DF'		Device on line 2		

Bytes 0–3 The NI instruction is executed in main line coding to turn off the external line status after its detection.

PDSTAT+1 will contain one more of the following interrupt codes:

External interrupt code bit	Interrupt Code (byte X'87')	External interrupt cause
15	nnnnnnn1	External signal 7
14	nnnnnn1n	External signal 6
13	nnnnn1nn	External signal 5
12	nnnn1nnn	External signal 4
11	nnn1nnnn	External signal 3
10	nn1nnnnn	External singal 2

n = other external interrupt conditions

Byte 4 Contains the flag of the partition containing the DTF
Background = 10
Foreground = 10–50, depending on the number of partitions

Bytes 5–7 Contain the address of the DTF table

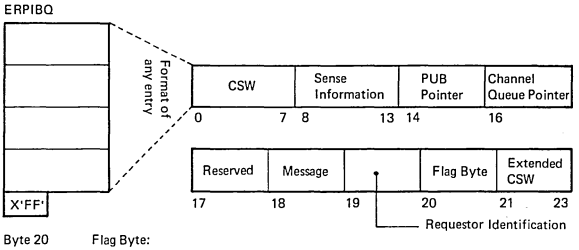
Table of pointers (PDTABA) to DTF addresses with the external interrupt line. The table is set up to handle the status in descending order from bit 15 to bit 10 of the external interrupt code.

PDTABA

Byte	0	1	2	3	4	5	6	7
0	00	08	00	10	00	08	00	18
8	00	08	00	10	00	08	00	20
16	00	08	00	10	00	08	00	18
24	00	08	00	10	00	08	00	28
32	00	08	00	10	00	08	00	18
40	00	08	00	10	00	08	00	20
48	00	08	00	10	00	08	00	18
56	00	08	00	10	00	08	00	

Bytes 126 and 127 (X'7E'–X'7F') of the partitions communication region contain the address of these tables. Label PDTABB identifies the first byte of the first table. The tables are also used for optical reader/sorters.

ERROR RECOVERY PROCEDURE INFORMATION BLOCK (ERPIB)



Byte 20 Flag Byte:

Bit	0	1 =	Channel Check on SIO
Bit	1–4		Reserved
	5	1 =	Sense data stored
	6	1 =	DASD ERPIB active
	7		Reserved

Note:
A free entry is indicated by X'FF' in byte zero, end of queue is indicated by 'FF' in byte zero. Label ERPIBQ identifies the first byte of queue.

ERBLOC

0 – 7	8 – 11	12 – 15	16 – 19	20 – 23
Name of phase to be fetched	Branch address to retry the failing operation	Branch address to ignore the error	Branch address to cancel the task	Return address for A-transient

The A-transient loads one of these addresses in register 14 before branching to the return address (see bytes 20–23).

ERRQ1

First error queue entry		Last error queue entry

Error queue entries (44 bytes each)

The number of entries is determined at supervisor generation.

Note:
The address of the ERBLOC can be found in SYSCOM at Displacement 0 (X'00').

Layout of an error queue entry for I/O error or BTAM

Bytes 0–7 CSW

Bytes 8–9 Address of PUB for device in error

Byte 10 Flag byte:

Bit 0 1 = No record found on DASD
1 1 = Intervention required
2 1 = Passback (set by device ERP)
3 1 = Allow ignore
4 Force automatic retry
6 Sense command outstanding
7 1 = Active entry

Byte 11 Message code: may refer to a device error recovery message generated by physical IOCS (See Cancel Codes and Messages) or:

This location may contain one of the following:
X'E2' The error is recoverable
X'AE' A record is to be recorded on the system recorder file and a physical transient is to be fetched (last two characters of phase name are in bytes 20–21).

CKD devices:

Bytes 12–15 Disk seek address. For an alternate entry, this field contains the address of the PUB of the originating routine.

FBA devices:

Byte 12 OS FBA device type code

Bytes 13–15 X'00'

Byte 16 PUB channel queue pointer

Bytes 17–19 Address of CCB/IORB

Bytes 20–43 Sense data or alternate entry name: If byte 11 contains X'AE', bytes 20–21 contain the last two characters of the phase name of the physical transient to be fetched.

CANCEL CODES AND MESSAGES

Cancel Code (hex)	Message Code	Description part of Message or Condition	Label
00	---------	In all cases default value exept those listed	
0A	0S21I	Processing error in access control	ERR0A
0B	0S20I	Access control violation	ERR0B
0C	0S19I	Execution failure in ICCF pseudo partition	ERR0C
0F	OP80I	Invalid 'read from/or write' to system file on FBA device	ERR0F
10	---------	Normal EOJ	ERR10
11	0V07I	No channel program translation for unsupported device	ERR11
12	0V06I	Insufficient buffer space for channel program translation	ERR12
13	0V05I	CCW with count greater than 32 K	ERR13
14	0V04I	Page pool too small	ERR14
15	0V02I	Page fault in disabled program	ERR15
17	0S02I	Program request (Same as 23 but causes dump because subtasks were attached when maintask issued CANCEL macro.)	ERR17
18	---------	Elimates cancel message when maintask issues DUMP macro with subtasks attached	
19	0P74I	I/O operator option	ERR19
1A	0P73I	I/O error	ERR1A
1B	0P82I	Channel failure	ERR1B
1C	0S14I	CANCEL ALL macro	
1D	0S12I	Main task termination	
1E	0S13I	Unknown ENQ requestor	
1F	0P81I	CPU failure	ERR1F
20	0S03I	Program check	ERR20
21	0S04I	Illegal SVC	ERR21
22	0S05I	Phase not found	ERR22
23	0S02I	Program request	ERR23
24	0S01I	Operator intervention	ERR24
25	0P77I	Invalid address	ERR25
26 *	0P71I	SYSxxx not assigned (unassigned LUB Code)	ERR26
27	0P70I	Undefined logical unit	ERR27
2A	0V10I	I/O error on page data set	
2B	0P84I	I/O error during fetch from PCIL	ERR2B
2C	0V09I	Illegal parameter passed by PHO routine	ERR2C
2D	0P88I	Failing strorage block (program cannot be executed)	ERR2D
2E	0S16I	Invalid resource request (possible deadlock)	ERR2E
2F	0V03I	More than 255 PFIX requests for 1 page	ERR2F
30	0P72I	Reading past/&statement (on SYSRDR or SYSIPT)	ERR30
31	oP75I	I/O error queue overflow (error queue overflow)	ERR31
32	0P76I	Invalid DASD address	ERR32
33	0P791	Invalid first CCW	
34		Reserved	
35	0P85I	Job control open failure	ERR35
36	0V08I	Program check or page fault in I/O appendage routine	ERR36
37		Reserved	
38	0V11I	Wrong privately translated CCW	ERR38
39		Reserved	

CANCEL CODES AND MESSAGES (. . . Cont'd)

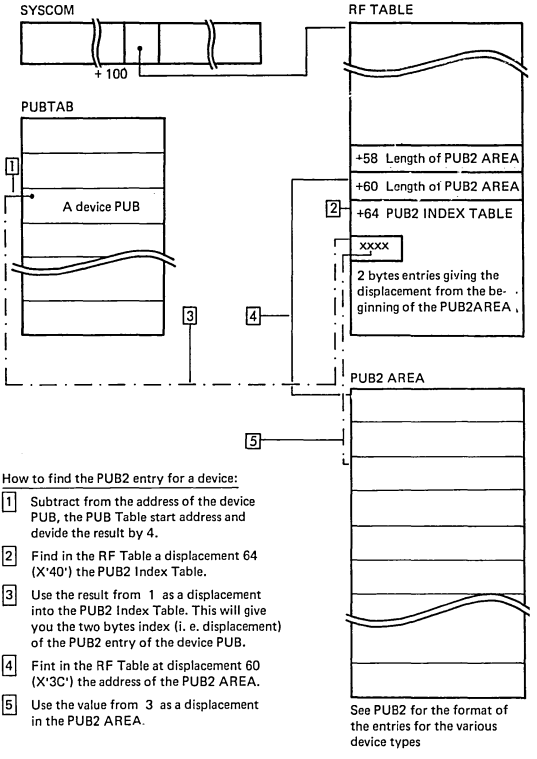
Cancel Code (hex)	Message Code	Description part of Message or Condition	Label
40	5J97I	ACF/VTAM error (invalid condition code)	
41	5J97I	ACF/VTAM error (invalid condition code)	
42	0P86I	Violated DASD file Protection	
XX	0P78I	Unrecognized Cancel Code	
	0P83A**	Supervisor catalog failure	
	0P87A**	IPL Failure	

* If the CCB/IORB is not available, the logical unit is SYSxxx.

** The cancel code is not significant in case of a supervisor catalog or IPL failure, because the system is placed in the wait state without any further processing by the terminator.

XX Any other DIGITS.

PUB2 ENTRY ADDRESSING



How to find the PUB2 entry for a device:

1. Subtract from the address of the device PUB, the PUB Table start address and devide the result by 4.
2. Find in the RF Table a displacement 64 (X'40') the PUB2 Index Table.
3. Use the result from 1 as a displacement into the PUB2 Index Table. This will give you the two bytes index (i. e. displacement) of the PUB2 entry of the device PUB.
4. Fint in the RF Table at displacement 60 (X'3C') the address of the PUB2 AREA.
5. Use the value from 3 as a displacement in the PUB2 AREA.

See PUB2 for the format of the entries for the various device types

PUB 2 TABLE

PUB 2 Table Entry Format for Unit Record and Unsupported Devices

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte: Bit 0: 1 = Device in intensive mode Bit 1: 1 = Device in diagnostic mode Bit 2: 1 = No recording mode Bit 3: 1 = Call statistics transient 2 Bit 4: 1 = Use PUB 2 name completion field Bit 5: 1 = Volume opened on this device Bit 6, 7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	6	Statistical data counters
Total length 12		

PUB 2 Table Entry Format for DASD

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag byte: Bit 0: 1 = Device in intensive mode Bit 1: 1 = Device in diagnostic mode Bit 2: 1 = No recording mode Bit 3: 1 = Call statistics transient 2 Bit 4: 1 = Use PUB 2 name completion field Bit 5: 1 = Volume opened on this device Bit 6, 7 Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	1	Flag byte: Bit 0: 1 = Soft DASD error is queued Bit 1: 1 = ERP requests logging of error Bit 2–7: Reserved
7	2	Reserved
9	1	Physical module identifier
10	6	Volume serial number
End 3340 Total length 16		
16	8	Statistical data counters
End 2311 and 2314/2319 Total length 24		
24	8	Additional statistical data counters (3330, 3350, and FBA)
End all other DASD Total length 32		

PUB 2 Table Entry Formats for Tapes

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag Byte: Bit 0: 1 = Device in intensive mode Bit 1: 1 = Device in diagnostic mode Bit 2: 1 = No recording mode Bit 3: 1 = Call statistics transient 2 Bit 4: 1 = Use PUB2 name completion field Bit 5: 1 = Volume opened on this device Bit 6, 7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	2	Name of ERP that wants control

PUB 2 Table Entry Formats for Tapes

Decimal Displacement	Byte Length	Description
8	1	Flag byte 1: Bit 0: 1 = Unsolicited interrupt Bit 2: 1 = ERP is in control Bit 3: 1 = ERP requests repositioning Bit 4: 1 = Use original TIE byte * 0 = Use opposite TIE byte * Bit 5: 1 = Intercept next SIO request * Bit 6: 1 = ERP read opposite request * Bit 7: 1 = Restart user's CCW chain *
9	1	Flag byte 2: Bit 0: 1 = Last ERP operation was ERG * Bit 1: 1 = Last ERP operation was reposition * Bit 2: 1 = Cleaner action in progress * Bit 3: 1 = Read Opposite Recovery in progress * Bit 4: 1 = Message stored in P2ORGTIE Bit 5: 1 = Error on attempt to recover by repositioning * Bit 6: 1 = Data check after ERP in control * Bit 7: Reserved
10	1	Flag byte3*: Bit 0: 1 = Failing CCW is Write or Control command Bit 1: 1 = User reading backwards Bit 2: 1 = Read Opposite Recovery (ROR) Bit 3: 1 = Maximum ROR retries Bit 4: 1 = Command chaining ROR Bit 5: 1 = ROR suppressed incorrect length Bit 6: 1 = User used SILI (Suppress Incorrect Length Indicator) Bit 7: 1 = Reserved
11	1 8	Temporary read count Save Area for 1st 8 sense bytes (8809)
12	1	Temporary write count
13	1	Noise record count
14	2	Erase gap count
16	2	Cleaner action count
18	1	Permanent read errors count
19	1	Permanent write errors count *
20	1 1	TIE original direction SAVE AREA for message number (8809)
21	1	TIE opposite direction *
22	1	ERP counter 0 *
23	1	ERP counter 1 *
24	8	ERP work area
32	6	Tape serial number
38	2	Block Length
40	4 8	User ROR command address from CSW User CCW area during rewind/unload (8809)
44	2	User ROR residual count from CSW
46	2	Reserved
48	16 24	Work area for rewind-unload error action SAVE AREA for last 24 sense bytes (8809), end of 8809 entry: total length 72 bytes
64	10	2400-series statistical data counter area
74	2	Reserved
End 2400-series: Total length 76		
64	20	3410/3420 statistical data counter area
End 3410/3420: Total length 84		

* Fields not used for 3402 tape entry

PUB 2 TABLE (. . . Cont'd)

PUB 2 Table Entry Format for 3886 Optical Character Reader

Decimal Displacement	Byte Length	Description
0	3	Usage count (number of non-ERP SIOs)
3	1	Flag byte: Bit 0: 1 = Device in intensive mode Bit 1: 1 = Device in diagnostic mode Bit 2: 1 = No recording mode Bit 3: 1 = Call statistics transient 2 Bit 4: 1 = Use PUB2 name completion field Bit 5: 1 = Volume opened on this device Bit 6, 7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit mask
6	20	Statistical data counters
End 3886	Total length 26	

PUB 2 Table Entry Format for 3540 Diskette

Decimal Displacement	Byte Length	Description
0	3	Usage Count (number of non-ERP SIOs)
3	1	Flag byte: Bit 0: 1 = Device in intensive mode Bit 1: 1 = Device in diagnostic mode Bit 2: 1 = No recording mode Bit 3: 1 = Call statistics transient 2 Bit 4: 1 = Use PUB2 name completion field Bit 5: 1 = Volume opened on this device Bit 6, 7: Reserved
4	1	CE mode limit byte
5	1	CE mode byte/bit maske
6	8	Statistical data counters
14	1	Bit 0–5: Reserved Bit 6: 1 = System file opened by Job Control Bit 7: 1 = System file opened by Problem Program
15	1	Reserved
End 3540	Total length 16	

RECORDER FILE TABLE (RF TABLE)

Displacement Dec	Displacement Hex	Label	Byte Length	Description
		RFTABLE		Label of Starting Address
0	0	RFFLAGS1	1	Bit 0: 1 = File full Bit 1: 1 = RDE option included Bit 2: 1 = Initial IPL Bit 3: Reserved Bit 4: 1 = File is to be created Bit 5: 1 = File has been created Bit 6: 1 = File is on FBA device Bit 7: 1 = File open and ready
1	1	RFFLAGS2	1	Bit 0: 1 = File full message request Bit 1: 1 = Threshold message request Bit 2: 1 = I/O error message request Bit 3: 1 = Data lost message request Bit 4: 1 = EVA message request Bit 5: 1 = File owned by RTA recorder Bit 6: 1 = File owned by PTA recorder Bit 7: 1 = File being accessed by EREP
2	2	RFLAGS3	1	Bit 0: 1 = Last track message issued once Bit 1: 1 = Error is to be recorded Bit 2: 1 = Short form records request Bit 3: 1 = Individual records for unlabeled tapes Bit 4: 1 = Reserved Bit 5: 1 = Reserved Bit 6: 1 = Exit to $$BOMT05 indicator for $$BOPEN Bit 7: 1 = Exit to $$BOMT01 indicator for $$BOPEN
3	3	RFFLAGS4	1	Work area switches for various transients including BTAM Bit 0: 1 = Multiple records required (PRT1, 3895, recording) Bit 1: 1 = PLB record required (PRT1 recording) OBR record required (3895 recording) Bit 2: 1 = FCB record required (PRT1 recording) MDR records required (3895 recording) Bit 3: 1 = UCB record required (PRT1 recording) Bit 4: 1 = Ignore exit requested (PRT1 recording) Bit 5: Reserved Bit 6: 1 = Retry exit requested Bit 7: 1 = Record not written
4	4	RFFLAGS5	1	Reserved
5	5	RFNOFN	1	N of N records (low order 4 bits contain the number of records to be recorded and high order 4 bits contain the record number being processed
6	6	RFRECTYP	1	Record type code
7	7	RFREL	1	DOS/VS ID-Release Level code
8	8	RFRDSW1	1	Record dependent switch 1
9	9	RFRDSW2	1	Record dependent switch 2
10	A	RFBUFLG	2	Length of data buffer (FBA)
12	C	RFMCONST RFBUFAD	2 4	Multiplier for track balance (CKD) Address of data buffer (FBA)
14	E	RFDCONST	2	Divisor for track balance (CKD)
16	10	RFOCONST RFNAVR	2 (2)	Overhead track for balance (CKD) Displacement of next available RDF in buffer (FBA)
18	12	RFRECLEN	2	Length of record
20	14	RFTIMEA	4	Address of RMSR time entry
24	18	RFRECADR	4	Address of record
28	1c	RFSEEK RFCUBL	7 (4) (3)	Work area for seek address (BBCCHHR) (CKD) Work area for block number (FBA) Reserved

RECORDER FILE TABLE (RF TABLE) (. . . Cont'd)

35	23	RFEREPK	1	Key of EREP partition
36	24	RFHDRCH RFHDRBL	4 (4)	SYSREC cylinder/head (CKD) SYSREC block number (FBA
40	28	RECHMAP	2	Map of supported channels
42	2A	RFCHIDC	8	Channel ID codes
50	32		2	Reserved
52	34	RFEXIT	4	Exit phase name or exit address
56	38	RFEVARTH	1	EVA read threshold
57	39	RFEVAWTH	1	EVA write threshold
58	3A	RFP2ENTL	2	Length of PUB2 Entry Area
60	3C	RFP2ENT	4	Address of PUB2 Entry Area
64	40	RFP2ITAB	*	PUB2 Index Table

* Two bytes are generated for each PUB2 entry in the system. (See PUB2 Entry Addressing for using the PUB2 Index Table to access the PUB2 entries.

Bytes 100–103 (X'64'–X'67') of the system communication region (SYSCOM) contain the address of the Recorder File Table. Label RFTABLE identifies the first byte of the table.

CRT CONSTANT TABLE (CRTTAB)

0 7	8 9 11	12 13 15	16 17 19	20 27
Name of CRT Routine	Dummy Sense Byte	Address of CRT Save Area	Flag Byte 1	Address of CRT Trans. Area	Flag Byte 2	Address of System Task Save Area	CRT error Information *

* limited channel logout is saved here

Byte 8 — Sense byte:

Bit 0 1 = Command reject
Bit 1 1 = Intervention required (only if Console Printer is attached)
Bit 4 1 = Equipment check (only if Console Printer is attached)
Bit 7 1 = Operation check

or: this byte is used for saving Name Indicator

Byte 12 — Flag byte 1:

X'80' ERP message
X'40' Unit check for CRT
X'20' CRT Fetch bound
X'10' Device End simulated
X'08' Validation error
X'04' Redisplay mode
X'02' CRT error

Byte 16 — Flag Byte 2:

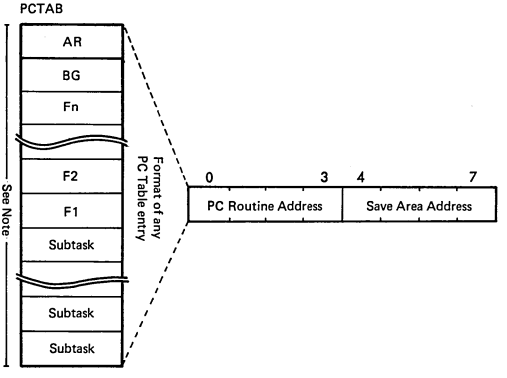
X'80' CRT busy
X'40' Sense byte prepared
X'20' End of CRT routine
X'10' Data already read
X'08' Attention pending
X'04' Request pending
X'02' Attention request
X'01' EOJ on CRT

Bytes 52–55 (X'34'–X'37') of the System Communication Region (SYSCOM) contain the address of the CRT CONSTANT Table. Label CRTTAB identifies the first byte of the table.

The first byte of this address, byte 52, inidcates the type of the generated CRT support. The bits have the following meaning:

Bits 0–5 Reserved
Bits 6–7 Indicate the support for the operator console that has been generated in the supervisor:
00 = Typewriter
01 = 125 D
11 = 3277/3278

PROGRAM CHECK OPTION TABLE



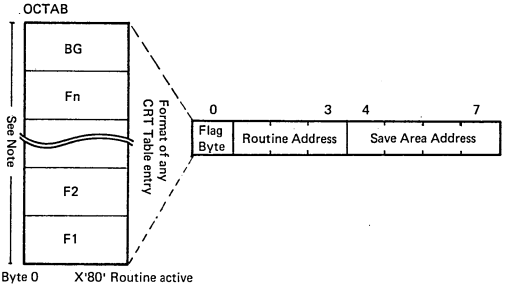
Bytes 0–3 No STXIT issued: zero
SIXT issued: Address of the user Program Check Routine
SIXT issued and the user routine is already in use: Complement of user Program Check Routine address

Bytes 4–7 No STXIT issued: zero
STXIT issued: Address of the user Save Area

Note:
Each generated table always comprises 16 entries; the subtask entries occupy the higher address locations in the table.

Bytes 100–101 (X'64'–X'65') of the partition communication region contain the address of the PC Option Table. Label PCTAB identifies the first byte of the table.

OPERATOR COMMUNICATION TABLE



Byte 0 X'80' Routine active

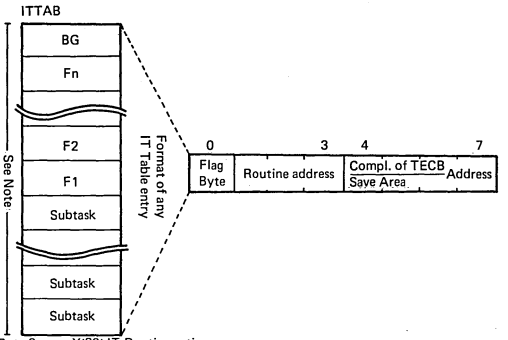
Bytes 1–3 No STXIT issued: zero
STXIT issued: Address of the user Operator Communication Routine
STXIT issued and the user routine is already in use: Complement of the user Operator Communication Routine address

Bytes 4–7 No STXIT issued: zero
STXIT issued: Address of the user Save Area

Note:
There is one entry for each partition supported.

Bytes 104–105 (X'68'–X'69') of the partition communication region contain the address of the OC Table. Label OCTAB identifies the first byte of the table.

INTERVAL TIMER TABLE



Byte 0 X'80' IT Routine active

Bytes 1–3

No TECB or STXIT issued:	zero
TECB issued:	Address of the timer event control block
STXIT issued:	Address of user interval timer routine
STXIT issued and user routine is already in use:	Complement of the user interval timer routine

Note:
Bit 1 of the byte 0 is used as a flag bit:
Bit 1 = If timer interrupt occurred and timer exit routine could not be entered. The processing of the timer interrupt is delayed until exit IT or Exit AB is issued.

Bytes 4–7

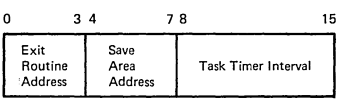
No TECB or STXIT issued:	zero
TECB issued:	Complement of the TECB address
STXIT issued:	Address of the user save area

Note:
One table entry is built for each partition.

The table always comprises 15 entries; the subtask entries occupy the higher address locations in the table.

Bytes 102–103 (X'66'–X'67') of the partition communication region contain the address of the IT Table. Label ITTAB identifies the first of the table.

TASK TIMER OPTION TABLE



Bytes 0–3

No STXIT issued:	zero
STXIT issued:	Address of entry point of user's task timer routine address
STXIT issued and user's routine already in use:	Complement of the task timer routine address

Bytes 4–7

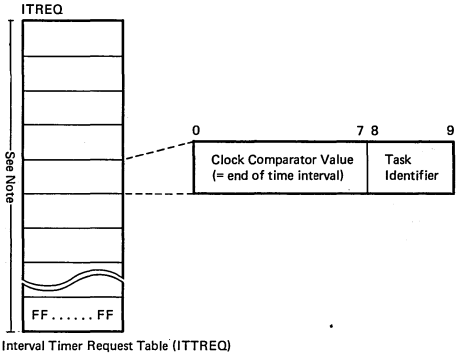
No STXIT issued:	zero
STXIT issued:	Address of the user's save area

Bytes 9–15

No SETT issued:	zero or negative
SETT issued:	Time remaining of the interval specified. The time is expressed in micro-seconds in bits 4–51. Bits 52–63 are ignored.

Note:
Bytes 176–179 (X'B0'–X'B3') of the System Communications Region (SYSCOM) contain the address of Task Timer Table. Label TTTAB identifies the first byte of the table.

INTERVAL TIMER REQUEST TABLE



Interval Timer Request Table (ITTREQ)

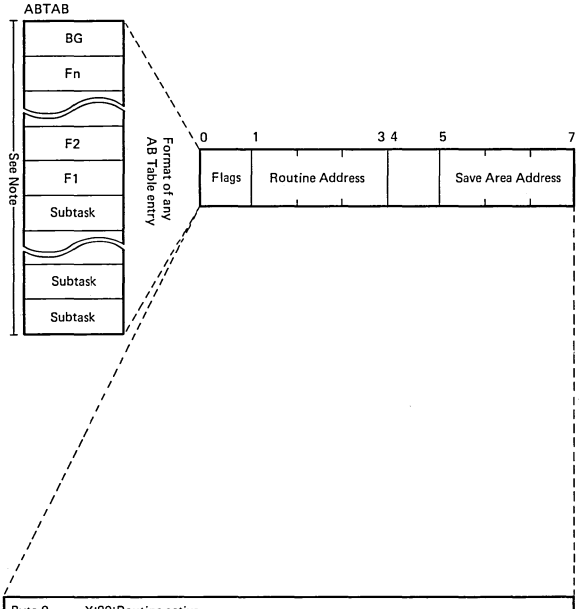
The entries in the ITREQ are either active or inactive. An active entry contains in bytes 0–7 the clock comparator value (= end of interval) and in bytes 8–9 the task identifier. The lowest value occupies the first position of the table, the highest value the position before the inactive entries.

All bits of an inactive entry are set to one (binary format). The last entry is always inactive and all entries are set inactive after IPL.

The clock comparator is set to the value contained in the first entry of the ITREQ table.

Bytes 80–83 (X'50'–X'53') of the System Communication Region (SYSCOM) contain the address of the IT Request Table. ITREQ identifies the first byte of the table.

ABNORMAL TERMINATION TABLE

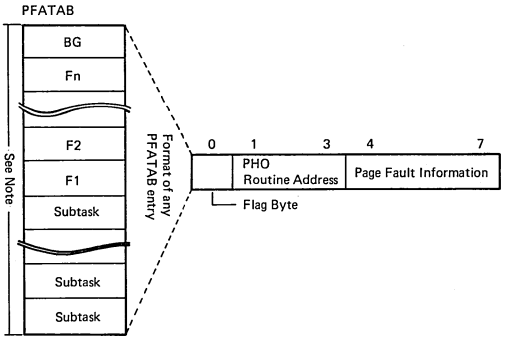


Byte 0	X'80'Routine active X'40'Suppress message and DUMP before entering AB routine	
Bytes 1–3	No STXIT issued:	zero
	STXIT issued:	Address of entry point of user's abnormal termination routine
	STXIT issued and the user AB routine already in use:	Complement of the user AB exit routine address
	The maintask and subtasks may have the same or different AB routines. When a subtask is ATTACHed after a STXIT AB macro has been issued by teh maintask, the subtaks will receive the AB routine address specified by the maintask only if the ATTACH macro for that subtask has the ABSAVE parameter specified. The subtask can override this by issuing its own STXIT AB macro.	
Bytes 4–7	No STXIT issued or no save area parameter passed:	zero
	STXIT issued and save area parameter passed:	Address of a 72-byte save area used by the supervisor to store the interrupt status information and the contents of the general requesters.

Note:
One table entry is generated for each partition supported. The table always comprises 15 entries; the subtask entries occupy the higher address locations in the table.

Bytes 84–87 (X'54'–X'57') of the System Communication Region (SYSCOM) contain the address of the AB Table. Label ABTAB identifies the first byte of the table.

PAGE FAULT HANDLING OVERLAP TABLE



SETPFA issued

Byte 0 Bit 0 1 = A page fault from the task is in the Page Queue
Bit 1 1 = A page fault occuring in a supervisor service of the task is queued in the Page Queue Table
Bits 2–7 Reserved

Bytes 1–3 Address of the user's page fault appendage routine

Bytes 4–7 Page fault information (for detail layout see Page - IN - Queue Table)

This information is stored here only when a page fault occurs in a supervisor service working for the task, while another page fault is queued for the task in the Page Queue, otherwise zero.

NO SETPFA issued

Bytes 0–7 zero

Note:
One table entry is generated for each partition supported. The table always comprises 15 entries; the subtask entries occupy the higher address locations in the table. Label PFATAB identifies the first byte of the table.

JOB ACCOUNTING INTERFACE COMMON TABLE

Displacement	Label	Description
	(ACCTCOMN)	
0–15	ACCTSVRG	Temporary register save area
16–17	ACCTSVRX	Reserved
18–19	ACCTSVRE	Reserved
20–23	ACCTPCNT	Count of partitions using the Job Accounting interface
24	ACCTSAID	Reserved
25	ACCTFAID	Reserved
26	ACCTRAID	Accounting ID: X'00' Overhead time X'04' Allbound time Byte 1 of PIK for CPU time
27	ACCTSWCH	Accounting switches: Bit 0 1 = Cancel accounting Bit 1 1 = No active partitions Bit 2 1 = Catalog in process Bit 3 1 = Alternate label area Bit 4 1 = IPL indicator Bit 5 1 = Reserved Bit 6 1 = Reserved Bit 7 1 = Reserved
28–31	ACCTIME	Reserved
32–33	ACCTRESC	Reserved
34–35	ACCTUSEP	Address of user save area (ACCTUSER)
36–37	ACCTUSEL	Length of user save area (Set with 1st operand of FOPT macro parameter JALIOCS)
38–39	ACCT$JOB	Job accounting partition indication
40–43	ACCTBLES	Address of BG Job Accounting Table

This table is to be extended (depending on the number of supported partitions) according to the pattern shown below.

NPARTS=n

44–47		Address of Fn Accounting Table . . .
+ 4		Address of F2 Job Accounting Table
+ 4		Address of F1 Job Accounting Table
+ (n*2)-1	ACCTSEAS	Control Field: prevents the accounting routine being loaded twice

* These values are the same as the PIK values for the relevant tasks.

Byte 124–127 (X'7C'–X'7F') of the system communication region (SYSCOM) contain the address of the Job Accounting Interface Common Table. Label ACCTCOMN identifies the first byte of the tabel.

JOB ACCOUNTING INTERFACE PARTITION TABLE

Displacement	Label	Description
	(ACCTABLE)	
0–3	ACCTWK1	Work area used by job control
4–7	ACCTWK2	Work area used by job control
8–11	ACCTSVPT	Job card pointer; address of job card field following job-name
12–13	ACCTPART	ID of partition in charge (partitions switch name)
14–15	ACCTLEN	Length of SIO area = 6n+1, where n = number of devices for this partition in SYSGEN option JA = (n, n, . . .)
16–21	ACCTLOAD	Label area instruction; moves JAI label area adress to OPEN/CLOSE transients
22–23	ACCTRES3	Reserved
24–27	ACCTLADD	Address of alternate label area
28–31	ACCTCPUT	Counter for program execution time (processor time) in the current jobstep. Counted in 300th of a second.
32–35	ACCTOVHT	Counter for system excution time (overhead time) in the current job step, distributed in proportion of the processor time.
36–39	ACCTBNDT	Counter for all-bound time; system wait state time divided between running partitions.
40–47	ACCTSVJN	Save area for job name during simulated EOJ
48–55	ACCTJBNM	Job name; taken from job card
56–71	ACCTUSRS	User information; 16 bytes form job card
72–73	ACCTPTID	Partition ID: 'BG', 'F1', 'F2', etc. in EBCDIC format
74	ACCTCNCL	Cancel code; see Cancel Codes and Messages
75	ACCTYPER	Type of record 'S' = job step, 'L' = last step of job
76–83	ACCTDATE	Date in format specified at SYSGEN (MM/DD/YY or DD/MM/YY)
84–87	ACCTSTRT	Start time of a job-step, in packed decimal (DHHMMSSF; F = sign)
88–91	ACCTSTOP	Stop time of a job step in the same format as ACCTSTRT. This value is used as the start time for the next step.
92–95	ACCTRES	Reserved
96–103	ACCTEXEC	Phase name; taken from execute card
104–107	ACCTHICR	2K * number of partition pages referenced (or pfixed for real execution) by the problem program in the current job step.
108–111	ACCTIMES	Same as ACCTOPUT at the end of the job step
112–115		Same as ACCTOVHT at the end of job step
116–119		Same as ACCTENDT at the end of job step
120	ACCTSIOS	SIO tables: 8 bytes for each device specified by SYSGEN options, as follows: 2 bytes for device address (Ocuu), 4 bytes for count of SIOS in current jobstep.
		Overflow byte: norally X'20', but is X'30' if more defices are used within a partition that specified by SYSGEN options.

Notes:
DSECT ACCTABLE symbolically addresses the JAI Partition Tables with labels, as shown. Each partition in which JAI is supported has its own JAI Partition Table. This table is labeled ACCTBG for the active partition BG; ACCTF1 for the active partition F1; ACCTF2 for F2, etc.

The address of this table is in the partition Communication Region at displacement 116 (X'74').

ASCII EBCDIC TRANSLATION TABLES

ASCII/EBCDIC (0/0 to 3/2)

ASCII					EBCDIC				
Charact.	Col	Row	Bit Pattern		Col (in hex)	Row	Bit Pattern		Comments
NUL	0	0	0000	0000	0	0	0000	0000	
SOH	0	1	0000	0001	0	1	0000	0001	
STX	0	2	0000	0010	0	2	0000	0010	
ETX	0	3	0000	0011	0	3	0000	0011	
EOT	0	4	0000	0100	3	7	0011	0111	
ENQ	0	5	0000	0101	2	D	0010	1101	
ACK	0	6	0000	0110	2	E	0010	1110	
BEL	0	7	0000	0111	2	F	0010	1111	
BS	0	8	0000	1000	1	6	0001	0110	
HT	0	9	0000	1001	0	5	0000	0101	
LF	0	10	0000	1010	2	5	0010	0101	
VT	0	11	0000	1011	0	B	0000	1011	
FF	0	12	0000	1100	0	C	0000	1100	
CR	0	13	0000	1101	0	D	0000	1101	
SO	0	14	0000	1110	0	E	0000	1110	
SI	0	15	0000	1111	0	F	0000	1111	
DLE	1	0	0001	0000	1	0	0001	0000	
DC1	1	1	0001	0001	1	1	0001	0001	
DC2	1	2	0001	0010	1	2	0001	0010	
DC3	1	3	0001	0011	1	3	0001	0011	
DC4	1	4	0001	0100	3	C	0011	1100	
NAK	1	5	0001	0101	3	D	0011	1101	
SYN	1	6	0001	0110	3	2	0011	0010	
ETB	1	7	0001	0111	2	6	0010	0110	
CAN	1	8	0001	1000	1	8	0001	1000	
EM	1	9	0001	1001	1	9	0001	1001	
SUB	1	10	0001	1010	3	F	0011	1111	
ESC	1	11	0001	1011	2	7	0010	0111	
FS	1	12	0001	1100	1	C	0001	1100	
GS	1	13	0001	1101	1	D	0001	1101	
RS	1	14	0001	1110	1	E	0001	1110	
US	1	15	0001	1111	1	F	0001	1111	
SP	2	0	0010	0000	4	0	0100	0000	
!	2	1	0010	0001	4	F	0100	1111	Logical OR
"	2	2	0010	0010	7	F	0111	1111	
#	2	3	0010	0011	7	B	0111	1011	
$	2	4	0010	0100	5	B	0101	1011	
%	2	5	0010	0101	6	C	0110	1100	
&	2	6	0010	0110	5	0	0101	0000	
'	2	7	0010	0111	7	D	0111	1101	
(	2	8	0010	1000	4	D	0100	1101	
)	2	9	0010	1001	5	D	0101	1101	
*	2	10	0010	1010	5	C	0101	1100	
+	2	11	0010	1011	4	E	0100	1110	
,	2	12	0010	1100	6	B	0110	1011	
–	2	13	0010	1101	6	0	0110	0000	Hyphen, Minus
.	2	14	0010	1110	4	B	0100	1011	
/	2	15	0010	1111	6	1	0110	0001	
0	3	0	0011	0000	F	0	1111	0000	
1	3	1	0011	0001	F	1	1111	0001	
2	3	2	0011	0010	F	2	1111	0010	

ASCII/EBCDIC (3/3 to 6/6)

ASCII					EBCDIC				
Charact.	Col	Row	Bit Pattern		Col (in hex)	Row	Bit Pattern		Comments
3	3	3	0011	0011	F	3	1111	0011	
4	3	4	0011	0100	F	4	1111	0100	
5	3	5	0011	0101	F	5	1111	0101	
6	3	6	0011	0110	F	6	1111	0110	
7	3	7	0011	0111	F	7	1111	0111	
8	3	8	0011	1000	F	8	1111	1000	
9	3	9	0011	1001	F	9	1111	1001	
:	3	10	0011	1010	7	A	0111	1010	
;	3	11	0011	1011	5	E	0101	1110	
<	3	12	0011	1101	4	C	0100	1100	
=	3	13	0011	1101	7	E	0111	1110	
>	3	14	0011	1110	6	E	0110	1110	
?	3	15	0011	1111	6	F	0110	1111	
@	4	0	0100	0000	7	C	0111	1100	
A	4	1	0100	0001	C	1	1100	0001	
B	4	2	0100	0010	C	2	1100	0010	
C	4	3	0100	0011	C	3	1100	0011	
D	4	4	0100	0100	C	4	1100	0100	
E	4	5	0100	0101	C	5	1100	0101	
F	4	6	0100	0110	C	6	1100	0110	
G	4	7	0100	0111	C	7	1100	0111	
H	4	8	0100	1000	C	8	1100	1000	
I	4	9	0100	1001	C	9	1100	1001	
J	4	10	0100	1010	D	1	1101	0001	
K	4	11	0100	1011	D	2	1101	0010	
L	4	12	0100	1100	D	3	1101	0011	
M	4	12	0100	1101	D	4	1101	0100	
N	4	14	0100	1110	D	5	1101	0101	
O	4	15	0100	1111	D	6	1101	0110	
P	5	0	0101	0000	D	7	1101	0111	
Q	5	1	0101	0001	D	8	1101	1000	
R	5	2	0101	0010	D	9	1101	1001	
S	5	3	0101	0011	E	2	1110	0010	
T	5	4	0101	0100	E	3	1110	0011	
U	5	5	0101	0101	E	4	1110	0100	
V	5	6	0101	0110	E	5	1110	0101	
W	5	7	0101	0111	E	6	1110	0110	
X	5	8	0101	1000	E	7	1110	0111	
Y	5	9	0101	1001	E	8	1110	1000	
Z	5	10	0101	1010	E	9	1110	1001	
[	5	11	0101	1011	4	A	0100	1010	
\	5	12	0101	1100	E	0	1110	0000	Reverse Slant
]	5	13	0101	11'1	5	A	0101	1010	
—	5	14	0101	1110	5	F	0101	1111	Logical Not
—	5	15	0101	1111	6	D	0110	1101	Underscore
`	6	0	0110	0000	7	9	0111	1001	Grave accent
a	6	1	0110	0001	8	1	1000	0001	
b	6	2	0110	0010	8	2	1000	0010	
c	6	3	0110	0011	8	3	1000	0011	
d	6	4	0110	0100	8	4	1000	0100	
e	6	5	0110	0101	8	5	1000	0101	
f	6	6	0110	0110	8	6	1000	0110	

ASCII EBCDIC (6/7 to 7/15)

ASCII					EBCDIC				
Charact.	Col	Row	Bit Pattern		Col	Row	Bit Pattern		Comments
g	6	7	0110	0111	8	7	1000	0111	
h	6	8	0110	1000	8	8	1000	1000	
i	6	9	0110	1001	8	9	1000	1001	
j	6	10	0110	1010	9	1	1001	0001	
k	6	11	0110	1011	9	2	1001	0010	
l	6	12	0110	1100	9	3	1001	0011	
m	6	13	0110	1101	9	4	1001	0100	
n	6	14	0110	1110	9	5	1001	0101	
o	6	15	0110	1111	9	6	1001	0110	
p	7	0	0111	0000	9	7	1001	0111	
q	7	1	0111	0001	9	8	1001	1000	
r	7	2	0111	0010	9	9	1001	1001	
s	7	3	0111	0011	A	2	1010	0010	
t	7	4	0111	0100	A	3	1010	0011	
u	7	5	0111	0101	A	4	1010	0100	
v	7	6	0111	0110	A	5	1010	0101	
w	7	7	0111	0111	A	6	1010	0110	
x	7	8	0111	1000	A	7	1010	0111	
y	7	9	0111	1001	A	8	1010	1000	
z	7	10	0111	1010	A	9	1010	1001	
{	7	11	0111	1011	C	0	1100	0000	
|	7	12	0111	1100	6	A	0110	1010	Vertical Line
}	7	13	0111	1101	D	0	1101	0000	
~	7	14	0111	1110	A	1	1010	0001	Tilde
DEL	7	15	0111	1111	0	7	0000	0111	

EBCDIC					ASCII				
Charact.	Col (in hex)	Row	Bit Pattern		Col	Row	Bit Pattern		Comments
NUL	0	0	0000	0000	0	0	0000	0000	
SOH	0	1	0000	0001	0	1	0000	0001	
STX	0	2	0000	0010	0	2	0000	0010	
ETX	0	3	0000	0011	0	3	0000	0010	
HT	0	5	0000	0101	0	9	0000	1001	
DEL	0	7	0000	0111	0	15	0111	1111	
VT	0	B	0000	1011	0	11	0000	1011	
FF	0	C	0000	1100	0	12	0000	1100	
CR	0	D	0000	1101	0	13	0000	1101	
SO	0	E	0000	1110	0	14	0000	1110	
SI	0	F	0000	1111	0	15	0000	1111	
DLE	1	0	0001	0000	1	0	0001	0000	
DC1	1	1	0001	0001	1	1	0001	0001	
DC2	1	2	0001	0010	1	2	0001	0010	
DC3	1	3	0001	0011	1	3	0001	0011	
BS	1	6	0001	0110	0	8	0000	1000	
CAN	1	8	0001	1000	1	8	0001	1000	
EM	1	9	0001	1001	1	9	0001	1000	
FS	1	C	0001	1100	1	12	0001	1100	
GS	1	D	0001	1101	1	13	0001	1101	
RS	1	E	0001	1110	1	14	0001	1110	
US	1	F	0001	1111	1	15	0001	1111	

EBCDIC to ASCII (X'00' to '1F')

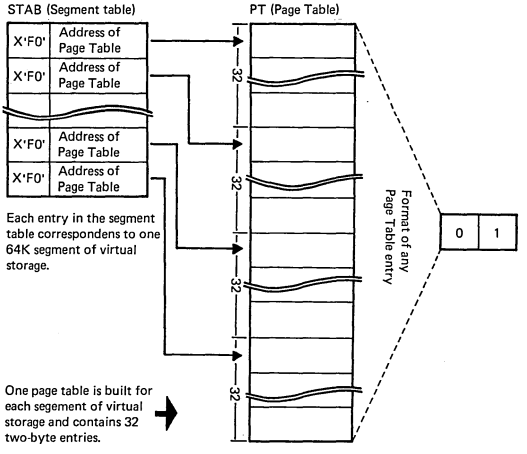
EBCDIC to ASCII (X'25' to X'93')

EBCDIC					ASCII				
Charact.	Col (in hex)	Row	Bit Pattern		Col	Row	Bit Pattern		Comments
LF	2	5	0010	0101	0	10	0000	1010	
ETB	2	6	0010	0110	1	7	0001	0111	
ESC	2	7	0010	0111	1	11	0001	1011	
ENQ	2	D	0010	1101	0	5	0000	0101	
ACK	2	E	0010	1110	0	6	0000	0110	
BEL	2	F	0010	1111	0	7	0000	0111	
SYN	3	2	0011	0010	1	6	0001	0110	
EOT	3	7	0011	0111	0	4	0000	0100	
EC4	3	C	0011	1100	1	4	0001	0100	
NAK	3	D	0011	1101	1	5	0001	0101	
SUB	3	F	0011	1111	1	10	0001	1010	
SP	4	0	0100	0000	2	0	0010	0000	
[	4	A	0100	1010	5	11	0101	1011	
.	4	B	0100	1011	2	14	0010	1110	
<	4	C	0100	1100	3	12	0011	1100	
(	4	D	0100	1101	2	8	0010	1000	
+	4	E	0100	1110	2	11	0010	1011	
!	4	F	0100	1111	2	1	0010	0001	Logical OR
&	5	0	0101	0000	2	6	0010	0110	
]	5	A	0101	1010	5	13	0101	1101	
S	5	B	0101	1011	2	4	0010	0100	
*	5	C	0101	1100	2	10	0010	1010	
)	5	D	0101	1101	2	9	0010	1001	
;	5	E	0101	1110	3	11	0011	1011	
—	5	F	0101	1111	5	14	0101	1110	Logical Not
-	6	0	0110	0000	2	13	0010	1101	Hyphen, Minus
/	6	1	0110	0001	2	15	0010	1111	
I	6	A	0110	1010	7	12	0111	1100	Vertical Line
,	6	B	0110	1011	2	12	0010	1100	
%	6	C	0110	1100	2	5	0010	0101	
—	6	D	0110	1101	5	15	0101	1111	Underscore
Π	6	E	0110	1110	3	14	0011	1110	
?	6	F	0110	1111	3	15	0011	1111	
`	7	9	0111	1001	6	0	0110	0000	Grave accent
:	7	A	0111	1011	3	10	0011	1010	
#	7	B	0111	1011	2	3	0010	0011	
@	7	C	0111	1100	4	0	0100	0000	
'	7	D	0111	1101	2	7	0010	0111	
=	7	E	0111	1110	3	13	0011	1101	
"	7	F	0111	1111	2	2	0010	0010	
a	8	1	1000	0001	6	1	0110	0001	
b	8	2	1000	0010	6	2	0110	0010	
c	8	3	1000	0011	6	3	0110	0011	
d	8	4	1000	0100	6	4	0110	0100	
e	8	5	1000	0101	6	5	0110	0101	
f	8	6	1000	0110	6	6	0110	0110	
g	8	7	1000	0111	6	7	0110	0111	
h	8	8	1000	1000	6	8	0110	1000	
i	8	9	1000	1001	6	9	0110	1001	
j	9	1	1001	0001	6	10	0110	1010	
k	9	2	1001	0010	6	11	0110	1011	
l	9	3	1001	0011	6	12	0110	1100	

EBCDIC to ASCII (X'94' t X'F9')

EBCDIC					ASCII				
Charact.	Rol (in hex)	Row	Bit Pattern		Col	Row	Bit Pattern		Comments
m	9	4	1001	0110	6	13	0110	1101	
n	9	5	1001	0101	6	14	0110	1110	
o	9	6	1001	0110	6	15	0110	1111	
p	9	7	1001	0111	7	0	0111	0000	
q	9	8	1001	1000	7	1	0111	0001	
r	9	9	1001	1001	7	2	0111	0010	
~	A	1	1010	0001	7	14	0111	1110	Tilde
s	A	2	1010	0010	7	3	0111	0011	
t	A	3	1010	0011	7	4	0111	0100	
u	A	4	1010	0100	7	5	0111	0101	
v	A	5	1010	0101	7	6	0111	0110	
w	A	6	1010	0110	7	7	0111	0111	
x	A	7	1010	0111	7	8	0111	1000	
y	A	8	1010	1000	7	9	0111	1001	
z	A	9	1010	1001	7	10	0111	1010	
{	C	0	1100	0000	7	11	0111	1011	
A	C	1	1100	0001	4	1	0100	0001	
B	C	2	1100	0010	4	2	0100	0010	
C	C	3	1100	0011	4	3	0100	0011	
D	C	4	1100	0100	4	4	0100	0100	
E	C	5	1100	0101	4	5	0100	0101	
F	C	6	1100	0110	4	6	0100	0110	
G	C	7	110	0111	4	7	0100	0111	
H	C	8	1100	1000	4	8	0100	1000	
I	C	9	1100	1001	4	9	0100	1001	
}	D	0	1101	0000	7	13	0111	1101	
J	D	1	1101	0001	4	10	0100	1010	
K	D	2	1101	0010	4	11	0100	1011	
L	D	3	1101	0011	4	12	0100	1100	
M	D	4	1101	0100	4	13	0100	1101	
N	D	5	1101	0101	4	14	0100	1110	
O	D	6	1101	0110	4	15	0100	1111	
P	D	7	1101	0111	5	0	0101	0000	
Q	D	8	1101	1000	5	1	0101	0001	
R	D	9	1101	1001	5	2	0101	0010	
\	E	0	1110	0000	5	12	0101	1100	Reverse Slant
S	E	2	1110	0010	5	3	0101	0011	
T	E	3	1110	0011	5	4	0101	0100	
U	E	4	1110	0100	5	5	0101	0101	
V	E	5	1110	0101	5	6	0101	0110	
W	E	6	1110	0110	5	7	0101	0111	
X	E	7	1110	0111	5	8	0101	1000	
Y	E	8	1110	1000	5	9	0101	1001	
Z	E	9	1110	1001	5	10	0101	1010	
0	F	0	1111	0000	3	0	0011	0000	
1	F	1	1111	0001	3	1	0011	0001	
2	F	2	1111	0010	3	2	0011	0010	
3	F	3	1111	0011	3	3	0011	0011	
4	F	4	1111	0100	3	4	0011	0100	
5	F	5	1111	0101	3	5	0011	0101	
6	F	6	1111	0110	3	6	0011	0110	
7	F	7	1111	0111	3	7	0011	0111	
8	F	8	1111	1000	3	8	0011	1000	
9	F	9	1111	1001	3	9	0011	1001	

SEGMENT AND PAGE TABLE



Bytes 0 and 1

Bit 0 1 = Address of any byte in the corresponding page is invalid

Bits 0–12 Leftmost 13 bits of address of page in real storage if bit 13 = 0

Bits 8–11 Storage key of corresponding partition if bit 13 = 1

Bit 13 0 = Page is in real storage
1 = Page is not in real storage

Bit 14 Always zero

Bit 15 PDS (Page Data Set) bit:
1 = Page must be read from page data set
(a valid copy of the page is on the page data set)
0 = Page need not be read from page data set
(no valid copy of the page on the page data set)

Bytes 208–211 (X'D0'–X'D3') of the System Communication Region (SYSCOM) contain the address of the Segment Table.

PAGE DATA SET TABLE

Page management uses the Page Data Set Table (DPDTAB) to calculate the correct address for a given page on the Page Data Set, if a read or write operation is necessary. Bytes 224–227 (X'E0'–X'E3') of the System Communication Region (SYSCOM) contain the adress of the DPDTAB. The DPDTAB consits of a header and one entry for the device description. Label DPDTAB identifies the first byte of the table. The table has the following layout:

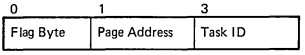
Header

Bytes:		
0–1	Reserved, must be 1	
2–3	Reserved, must be 1	
4–5	Number of virtual pages supported	
6–7	Length of DPDTAB entry	
8–11	Address of constant table for load leveler	
12–13	If /370 mode: If ECPS:VSE	Relocation value for Supervisor pages Mode:Reserved
14–15	If /370 mode: If ECPS:VSE	Relocation value for Non-Supervisor pages Mode:Reserved

Entry

Bytes:		
0–1	Channel and unit number of Page Data Set device	
2	Indicates FBA (1), CKD (2), or RPS (3) device	
3	Device type code	
4–5	If FBA device: If CKD device:	Block length Number of records per track
6–7	If FBA device: If CKD device:	Number of blocks per page Number of tracks per cylinder
8–11	If FBA device: If CKD device:	Block number of lower limit Relative track number of lower extent limit
12–15	If FBA device: If CKD device:	Number of blocks used for the extent Number of tracks used for the extent
16–17	PUB index	
18–23	Volume ID of Page Data Set	
24–25	Page number of upper limit on the extent	
26–27	Not used (for alignment only)	

PAGE IN QUEUE ENTRY (PGQUI)



Bytes:

0 Bits 0–3 Bits 0–3 of PIK or TIK
Bits 4–7 B'0000' Page Translation Exception
B'1000' Dummy request

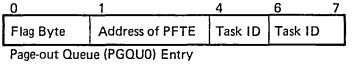
1–2 Leftmost 16 bits of the address of the page (which is also the address of the page frame, if GETREAL) to be handled. The remaining 8 bits of the address are assumed to be zero.

3 Task identifier (PIK or TIK) of user task or task identifier (select byte in STID field) or system task

The request that require the activity of the page-out (PGT) system task (it may be requested by GETREAL and for the handling of a PGQUI entry) are queued in the page-out queue (PGQU0), and handeld on a FIFO (first-in-first-out) basis.

The page-in queue has a maximum of 19 four-byte entries and label PGQUI identifies the first byte of the table.

PAGE-OUT QUEUE ENTRY (PGQU0)



Page-out Queue (PGQU0) Entry

Bytes:

0 Flag Byte
Bit 0 = 1 Posting required
Bits 1–7 Reserved

1–3 Address of PFTE to be handled

4,5 Task identifier of system task that requested the entry (only valid if bit 0 in byte 0 is on).

6, 7 Task identifier of user taks that requested the entry (only valid if bit 0 in byte 0 is on).

The page-out queue consits of 9 eight-byte entries, and the label PGQU0 identifies the first byte of the table.

PAGE FRAME TABLE ENTRY (PFTE)

0	1	3	4	5	6	8	12	15
PFTB Flag	Page Number (PNR)	370 Mode Flag	Waiting Task ID (WID)	PFIX Counter	TFIX Counter	Forward Pointer	Backward Pointer	

Page Frame Table Entry (PFTE)

Byte 0 – 3: Page frame address.

370 mode only:

Byte 3: Bit 4 = 0 If page frame belongs to supervisor or IPL partition.
Bit 4 = 1 If page frame belongs to initial page pool.

Bytes 4 – 7: zero

Byte 0 (PFTE flag):

Bit 0
(Hold Bit) 1 = Ensures that each task causing a page fault can use the page before it is disconected again.

Bit 1
(POE bit) 1 = Indicates that the PFTE has an entry in PGQUO.

Bit 2
(PO bit) 1 = Indicates that an active entry from the PMR task is in PGQUO.

Bit 3
(POA bit) 1 = Indicates that I/O for a page out has been started for this PFTE.

Bit 4
(RP bit) 1 = Indicates for the PGT task that the posting of a system task is required.

Bit 5
(RPUT bit) 1 = Indicates for the PGT task that the posting of a user task is required.

Bits 6 – 7 Reserved.

Byte 1 and 2: ff the page belongs to the page frame, byte 1 and 2 contains the leftmost 16 bits of the 3-byte address.

Byte 3 (370 mode flag byte):

Bit 0
(NFRP bit) 1 = Page in this page frame must not be TFIXed, since the page frame is in the PSQ.

Bit 1
(NFVP bit) 1 = Page belonging to this page frame is requested by PFIX. The page frame is not in the PSQ.

Bit 2
(DRAP bit) 1 = Indicates that the address space belonging to the PFTE is failing storage.

Bit 3
(PC bit) 1 = Indicates that a page is connected to the page frame. Either a page-in or an unconditional page-out request is in progress.

Bit 4
(PNRINV bit) 1 = Page frame is unused. It indicates that the PNR field, the FIX counters, the WID field, the PFTE flag and the 370 mode flag (except for the NFRP, DRAP and RPRESPF bits) are invalid.

Bit 5
(RPRESPF bit) 1 = Indicates to the PMR or PGT system that the posting of a task (for PFIX/GETREAL) that is waiting for the end of a page I/O for a reserved page frame is requeseted. NFRP is also set.

Bit 6
(RPPGFIX bit) 1 = Indicates for the PGT task that the posting of a task (for PFIX/TFIX), that is waiting for the end of a page-out for the page to be PFIXed, is requested. NFVP is also set.

Byte 4 (370 mode only):
Contains the TIK of the task requesting PFIX. The page frame of the page to be PFIXed does not belong to the corresponding real partition.

Byte 5: Indicates how often the page is permanently fixed.

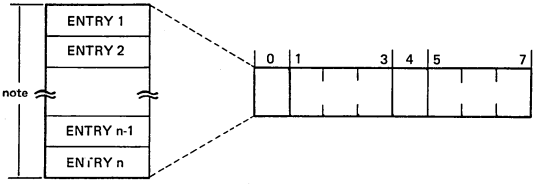
Byte 6 and 7: Indicates how often the page is temporarily fixed.

Byte 8 – 11: Pointer to the next PFTE.

Byte 12 – 15: Pointer to the preceding PFTE.

Note:
The pointers in bytes 7 through 15 are only valid if the PFTE is in the PSQ, or, for 370 mode, in the unused page frame queue.

PAGETAB



Byte 0	TIK (identifier of taks that issued the PAGEIN macro).
Bytes 1–3	Pointer to a list of areas that are to be paged in.
Byte 4	bit 0 1 = PAGEIN request has been completed bit 1 Reserved bit 2 1 = One more of the requested pages are outside the requesting program's partition. bit 3 1 = One or more negative length specifications were detected. bit 4 Reserved bit 5 Paging activity too high. Terminator required by load leveler. bit 6 Taks is terminating, request has to be deleted. bit 7 Second scan in progress.
Byte 5–7	Pointer to ECB (if used) or zero.

Note:
The number of entries is determined at supervisor generation time. Label PAGETAB identifies the first byte of the table.

TRANSLATION CONTROL BLOCK (TCB)

0	1	2	3 4	7 8	11 12	15 16	19
Flag byte	used by BTAM	TIK/PIK	Pointer to Status Modifier List	Pointer to Control Command List	Pointer to TIC Line	Pointer to Copy Block End	

20	23 24	27 28	47 48	51 52	55 56	59	60
Address of copied CCB (for cancel)	Number of free IDA words in IDAL blocks	Work areas	Address of last TFIX request	Pointer to DIDAL chain	Address of current DIDAL		Fast Translation Flag Byte

61	62	63 64	67 68	71 72	127 128	131
Reserved	Number of free DIDAL entries	Address of current DIDAL entry	Virtual CCW address	Save Area (Registers 2-F)	Pointer to next used TCB	

Byte 0 (TCBFLAG):
- = 1: Data chaining specified
- = 1: Read/Sense command specified
- = 1: Read backward command specified
- = 1: Status modifier command with data chaining
- = 1: Status modifier command with command chaining
- = 1: Request for FIX information block
- 6,7: Reserved

Byte 1 (ADBTAMCB):
Number of copy blocks needed in addition to those required for current CCW-translation request (refer to the section "BTAM Considerations").

Byte 4 (DEVSTPTR):
Pointer to status modifier list belonging to handled device. Zero if device does not support status modifier commands.

Byte 8 (DEVCDPTR):
Pointer to control command list which belongs to handled device. Zero if device does not support control commands with data area.

Byte 12 (LINEPTR):
Chain of knots of free structure caused by TIC following status modifier command. (Refer to the section "Translating Status Modifier Commands".) Zero if no knots exist in the CCW chain.

Byte 16 (BENDPTR):
Chain of knots built because status modifier command is last one fitting in CCW copy block. Zero if no status modifier commands at end of CCW copy blocks.

Byte 60 (TCBFLAG1):
- Bit 0 = 1 Replica creation requested
- Bit 1 = 1 Replica test in process
- Bit 2 = 1 Request for replica block
- Bit 3 = 1 Request for DIDAL block
- Bit 4–7 Reserved

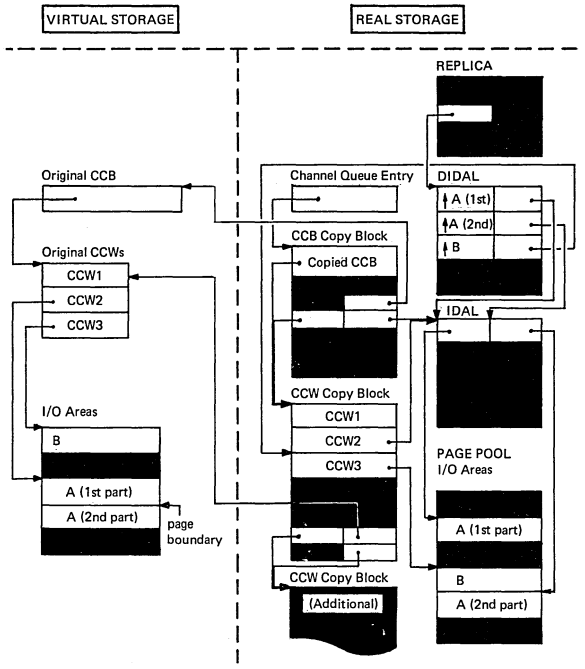
Bytes 52–71 as shown are for fast CCW translation only. For a normal translation byte 52 is the first byte of the save area.

To locate the TCB associated with the partition/task, add X'50' to the address of the System Save Area (displacement X'09' of the appropriate PIB). Labels CCWTCB1–CCWTOBn identify the first byte of the appropriate TCB.

TRANSLATION CONTROL AND COPY BLOCKS

The following control and copy blocks are used to copy and translate a CCB and channel program for a virtual mode I/O request:

- A translation control block (CCWTCB, or abbreviated as TCB). This block is used as a work and save area during translation.
- A CCB copy block. The user CCB and sense CCW (if any) are copied into this block. The CCB copy block also contains information about the copied and translated channel program.
- CCW copy blocks. Each block contains copy locations for up to 7 contiguous CCWs and queueing information.
- IDAL blocks. Blocks used for building Indirect Data Address Lists for CCWs having IDALs or for data areas which cross page boundaries.
- Fix information blocks. Each block contains a bit string for fix information for a block of 1088K of real storage. One or more fix informations blocks are generated if a page is fixed at a location greater than 384K (information for page frames up to that address is kept in the CCB copy block).



Layout of CCB Copy Block

	0	1	2	3	4	5	6	7
0	CCBCNT		CCB COM1	CCB COM2	CCB STA1	CCB STA2	CCB CLS *	CCB LNO
8	CCBCCW Address of first CCW				CCBBY3	CCBCSWW		
16	CCBSENS Sense CCW if any							
24	CCBPIK User PIK		CCB FLAG **	Unused	CCBVA Virtual Address of CCB			
32	CCBACB Address of first CCW copy block in channel program with lowest VBA				CCBICB Address of first IDAL block in channel program			
40	CCBXINF (Fix information; 24 bytes)							
48	Each bit in this field represents one page frame. If a bit is on, the associated page frame contains a page fixed for this I/O request. If more than 384K of real storage are available, the address in CCBXPTR will point to any additional field which contains bits for the page frames beyound 384K.							
56								
64	CCBXPTR Address of additional Fix information				CCBNEXT Address of next CCB copy block			

Copied CCB

Note:
If the fast CCW translation option is active, bytes 56–67 of the CCB copy block have a different meaning, as shown below:

Bytes 56–59 The address of the REPLICA block associated with this channel program.

Bytes 60–63 Pointer to the next CCB in the saved CCB queue used by the fast CCW translation routines.

Bytes 64–67 Pointer to the previous CCB in the saved CCB queue.

* Set to X'20' (= copied CCB)
** Legend CCBFLAG:

Bit 0 Indicates that CCW-translation of this request is complete; indicator is set before I/O request is enqueued in channel queue.

Bit 1 Indicates that at least one time during CCW-translation control has been transferred to TFIX routine. In case TTFIX is 0 scan through CCBXINF for freeing pages is skipped. Indicator is set immediately before control is transferred to TFIX routine.

Bit 2 Reserved

Bit 3 Indicates that the next CCW-translation request from BTAM is from the BTAM channel appendage. This indicator is set immediately after 1st time request from BTAM has been completed.

Bit 4 Valid for fast translation.

Bit 5 CCB copy block is on saved CCB queue.

Bit 6 Data area pages require fixing.

Bit 7 Reserved

Layout of CCW Copy Block

	0	1	2	3	4	5	6	7
0	1st Copy location for CCW							
8	2nd Copy location for CCW							
16	3rd Copy location for CCW							
24	4th Copy location for CCW							
32	5th Copy location for CCW							
40	6th Copy location for CCW							
48	7th Copy location for CCW							
56	X'80' *	X'000000'			Virtual address of first CCW in the Copy block			
64	X'88' **	X'000000'			Address of next CCW Copy block in the chain			

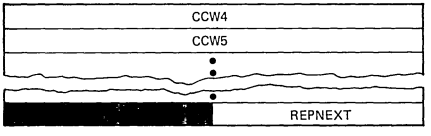
* X'80' indicates the end of the CCW copy locations in the block. It is replaced by a TIC (Transfer in Channel command) if the 7th copy location contains a copied CCW with data- or command chaining. Bytes 57–59 will then point to the copy location of the CCW following in the CCW in the 7th location. Bytes 56–59 will not be changed if the CCW in the 7th copy location is a TIC.

** X'88', indicates the last 8-byte entry in the block. It is replaced by a TIC if the CCW in the 7th copy location is a status modifier CCW. Bytes 65–67 will then point to the copy location of the second CCW following the status modifier CCW.

MAIN REPLICA BLOCK

VCCBA	RCCBA		
TIMEST	REPPIK	REPLCNT	CCWSTRL
REPDIDAL			
CCB			
	CCW1		
CCW1	CCW2		
CCW2	CCW3		
CCW3	REPFPT		
REPBPT	REPNEXT		

ADDITIONAL REPLICA BLOCK



Legend:

VCCBA	Virtual CCB address
RCCBA	Address of copied CCB
TIMEST	Timestamp
REPPIK	Partition Identifikation Key
REPLCNT	The number of tasks currently testing this REPLICA for a match with their channel program
CCWSTRL	Length of CCW string (number of CCWs)
REPDIDAL	Address of DIDAL block
REPFPT, REPBPT	Pointers used for chaining REPLICAs (forward and backward pointer)
REPNEXT	Pointer to (next) additional REPLICA block

DIDAL BLOCK

ENTRY 1	
2	
3	
4	
5	
6	
7	
8	
RESERVED	CHAIN POINTER ***

DIDAL ENTRY

VIRTUAL ADDRESS	FLAG BYTE *	POINTER TO REAL LOC. **

0 3 4 5 7

* FLAGBYTE

Bit 0 Indicates that TFIXing is not neccessary because the page has already been TFIXed for this request.

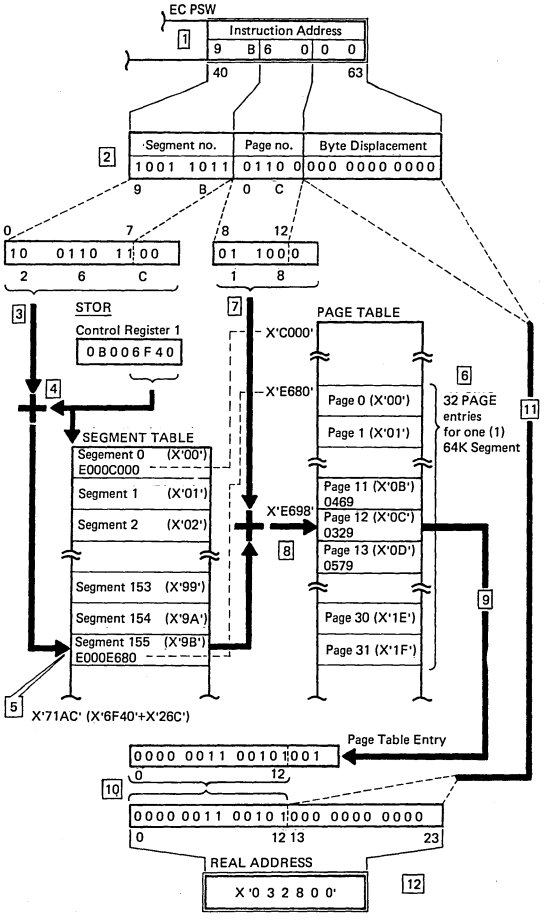
1-6 Reserved

Bit 7 Indicates that TFIX request for this entry has been completed.

** Real location (either copied CCW or IDA word) that should contain the translated I/O area address.

*** Points to (next) additional DIDAL. Contains 4X'00' in last DIDAL.

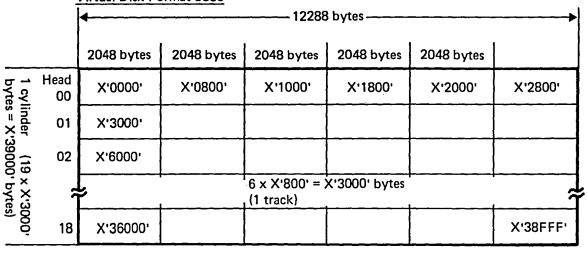
CONVERTING VIRTUAL TO REAL ADDRESS



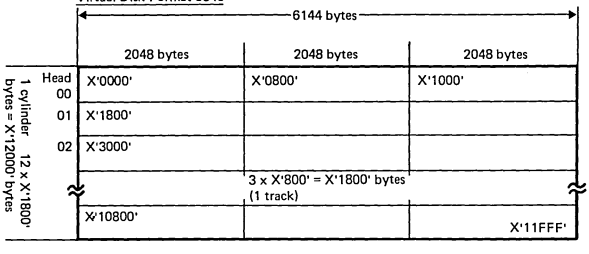
Note:
Values used in the figure are hypothetical.

PAGE DATA SET FORMAT

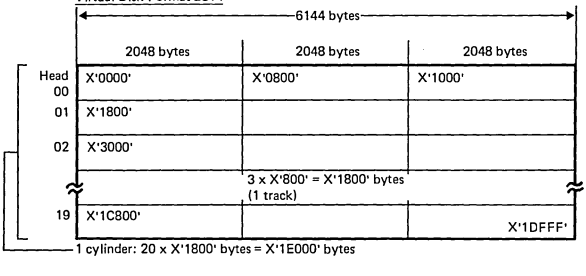
Virtual Disk Format 3330



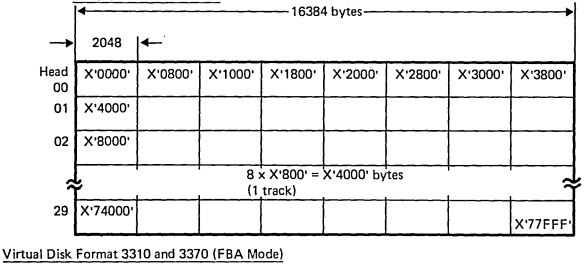
Virtual Disk Format 3340



Virtual Disk Format 2314



Virtual Disk Format 3350



Virtual Disk Format 3310 and 3370 (FBA Mode)

In Fixed Block Mode the 3310 and 3370 DASD will provide logical block number in ascending sequence. That means 4 blocks with 512 bytes are required to store one page on the PAGE DATA SET.

LAYOUT OF REPLICA HEADER BLOCKS

RHB1

X'00'	Flag Byte *	Number of Tests on Replica	PIK of Requestor
X'04'	Forward pointer in partiton's replica queue		
X'08'	Backward pointer in partition's replica queue		
X'0C'	Address of virtual CCB/IORB		
X'10'	Saved CCB/IORB		
X'20'	Pointer to RHB2		

RHB2

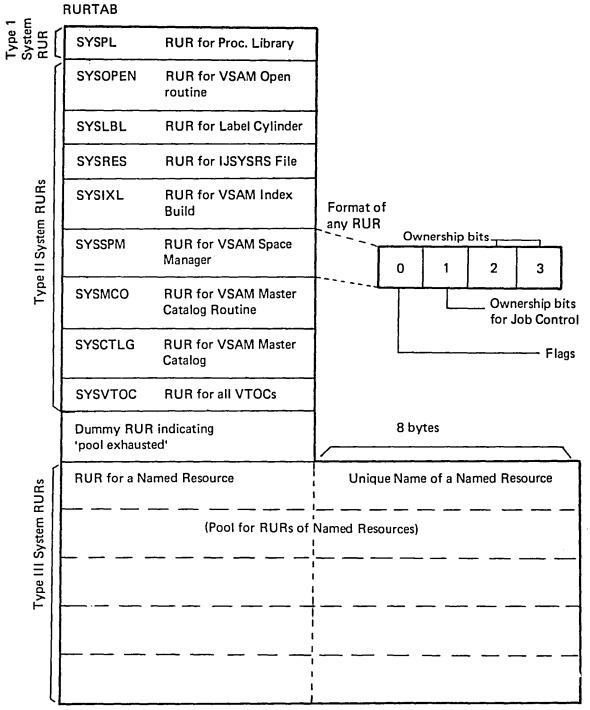
X'00'	Pointer to RHB1
X'04'	Pointer to associated FHB (fixlist)
X'08'	Time stamp
X'0C'	Length of saved channel program
X'10'	Reserved
X'14'	Reserved
X'18'	Saved user SENSE
X'20'	Pointer to next replica block

REPLICA BLCOK

X'00'	CCW1
X'08'	CCW2
X'10'	CCW3
X'18'	CCW4
X'20'	Pointer to next block or zero

* Flag Byte:
Bit 0 = 1 Freeing of replica request
Bit 1–7 Reserved

RESOURCE USAGE RECORD TABLE (RURTAB)



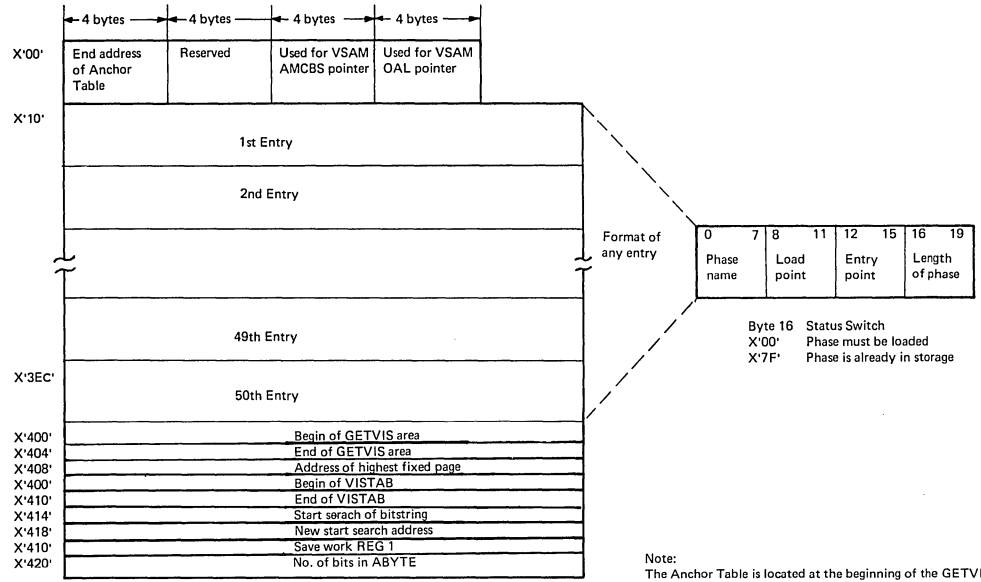
Byte 0
Bit 3 1 = Another task waiting for this resource
Bit 5 1 = Resource shared among owners indicated
Bit 6 1 = Resource used exclusive by indicated task
Bit 7 1 = Shared or exclusive use of resource

Byte 1 Used by Job Control to indicate that the resource is to be held for the duration of a job.
Bit 0 1 = TIK or PIK is X'70'
Bit 1 1 = TIK or PIK is X'60'
Bit 2 1 = TIK or PIK is X'50'
Bit 3 1 = TIK or PIK is X'40'
Bit 4 1 = TIK or PIK is X'30,
Bit 5 1 = TIK or PIK is X'20'
Bit 6 1 = TIK or PIK is X'10'

Byte 2:
Bit 0 1 = TIK is X'F0'
Bit 1 1 = TIK is X'E0'
Bit 2 1 = TIK is X'D0'
Bit 3 1 = TIK is X'C0'
Bit 4 1 = TIK is X'80
Bit 5 1 = TIK is X'A0'
Bit 6 1 = TIK is X'90'
Bit 7 1 = TIK is X'80'

Byte 3:
Bit 0 1 = TIK or PIK is X'70'
Bit 1 1 = TIK or PIK is X'60'
Bit 2 1 = TIK or PIK is X'50'
Bit 3 1 = TIK or PIK is X'40'
Bit 4 1 = TIK or PIK is X'30'
Bit 5 1 = TIK or PIK is X'20'
Bit 6 1 = TIK or PIK is X'10' (BG)
Bit 7 1 = TIK or PIK is X'00' (Attention)

Note:
For explanation of TIK and PIK see Task Interrupt Key and Partition Identifikation Key. Label RURTAB identifies the first byte of the table.



Note:
The Anchor Table is located at the beginning of the GETVIS area in the virtual partition.

FCB (LABEL 'DFCB')

X'00'	List of internal entry points
	List of external entry points
	State indicators (DFCBSW1—DFCBSW4)
X'40'	Start address of FETCH table (FTTAB)
X'44'	Pointer to FETCH work area (FCHWORK)
X'48'	Pointer to input control block (FICB)
X'4C'	Save area RG5—RG15
	Work areas
	TFIX/TFREE interface tables
	CHAIN table describing the searching sequence for the various directories
	FRPL area describing the interface to the Fetch I/O processing

FTTAB

X'00'	Condense counter
X'01'	PTR to SLD
X'04'	Device Characteristics
X'08'	Start of Directory
X'0C'	Channel Address
X'0E'	Reserved
X'0F'	X'01' FBA X'02' CKD X'03' RPS DEV.
X'10'	Start of LINK directory

FICB

X'00'	End of RLD buffer
X'04'	PTR to IDRA
X'08'	PTR to End of IDRA
X'0C'	Start of TXT CCW buffer
X'10'	End of TXT CCW buffer
X'14'	Start add. TXT CCW buffer
X'18'	WORKAREA
X'20'	WORKAREA
X'28'	Search name for directory read
X'30'	CCB/IORB

FCHWORK

X'00'	LOAD POINT
X'1C'	PTR to CHAIN TABLE
X'20'	PTR to DIRECTORY ENTRY
X'24'	PTR to Phasename
X'29'	PTR to LOCAL LIST
X'2C'	PTR to Relocation Factor
X'30'	PTR to RLD
X'50'	Phasename Relative Start Address Length of TXT

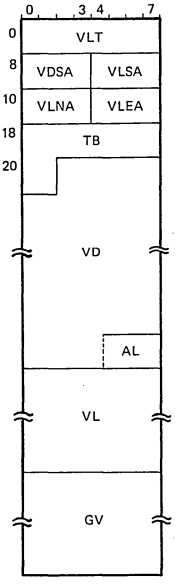
CHAIN

X'00'	Number of chain entries — 1
X'01'	Indicators: X'00' SDL search X'10' Directory search X'0C' LINK directory search
X'02'	LUB identification X'0006' SCIL X'080B' PCIL
X'04'	PTR to FTTAB

SLD

X'00'	Range of directory 1 = CKD dev (1 track) n*4 = FBA (n = number of 2K blocks)
X'04'	Number of SLD entries
X'06'	Highest phasename

LAYOUT OF SHARED VIRTUAL AREA (SVA)



Field	Length	Content
VLT	8	Date + time SVA has been updated (note 1)
VDSA	4	Start address of the System D Directory List
VLSA	4	Start address of the virtual library (note 2)
VLNA	4	Address of next available location in SVA
VLEA	4	Address of end of the SVA
TB	10	Communications area for Job Control and $MAINDIR
VD	up to 32 K	System Directory List (note 3)
AL	<8	Alignment bytes for doubleword boundary
VL	any	Virtual library containing re-entrant and relocatable phases
GV	any	GETVIS area for the system (starts on page boundary)

Note 1:
As one of the tests to determine whether a warm start copy of the SVA is available, job control compares this entry after IPL against the date and time in the SCIL descriptor entry.

Note 2:
Address of first doubleward aligned byte after SDL.

Note 3:
The layout of this area is compatible with a directory block in the Core Image Library. The SDL has fixed length entries of 34 bytes. The last entry contains 8X'FF' as phasename.

LAYOUT OF DIRECTORY ENTRY

0 7	8 10	11	12 13	14 15	16	17	18 20	21 23	24 26	27	29 30	31	33
Phase name	TTR	N	TT	LL	I	T	PPP	EEE	RR	R	AAA	C	VEE
	LBLN												

Bytes 0–7	Phase name	
Bytes 8–10	TTR	Relative track address and record number of phase (note 2)
Byte 11	N	Number of halfwords containing user data
Bytes 8–11	LBLN	Logical block number of phase on FBA device
Bytes 12–13	TT	Number of text blocks
Bytes 14–15	LL	Number of text bytes in last next block
Byte 16	I	Switch indicating type of phase
		X'80' selfrelocating phase
		X'40' relocatable phase
		X'20' SVA eligible
		X'10' phase is in the SVA (note 1)
		X'08' phase found in a PICL (note 1)
		X'04' phase not found (note 1)
		X'02' active entry (filled in)
		X'01' not used
Byte 17	T	Always X'00' (used as type byte for stow table)
Bytes 18–20	PPP	Load point of LNKEDT time (note 3)
Bytes 21–23	EEE	Entry point at LNKEDT time (note 3)
Bytes 24–25	RR	Number of RLD items (note 4)
Byte 26	R	Number of additional RLD blocks (note 4)
Bytes 27–29	AAA	Partition start address of LNKEDT time (note 4)
Byte 30	C	Condense counter at the time when entry was activated (note 6)
Bytes 31–33	VEE	Entry point of phase in SVA (notes 1 and 5)

Note 1: Only used for directory entries that are in storage.
Note 2: The TTR is relative to the beginning of the directory.
Note 3: PPP and EEE are not present if both are zero and the phase is not relocatable.
Note 4: RR, R and AAA are only present if the phase is relocatable.
Note 5: VEE is only present if the phase is SVA eligible.
Note 6: Condense counter is used for incore directory entries (not SDL).

Note:
The last entry in the directory is a 12-byte with a dummy phasename (contains 8X'FF'), a dummy TTR (contains XL3'00') and a dummy N (contains X'00'). Directory entries in storage always have the standard length of 34 bytes (including the last entry).

ROUTINE IDENTIFIERS (RID)

NAME	ID	MEANING	ACTION
SYSTEMID	00	System error condition, for example, page fault in I/O interrupt handler	Hard Wait
REENTRID	04	Page fault or GETREAL request in a reenterable routine	Save PSW and regs to user task's system save area, set PIB DAT flag to call SVRETURN, and ENQU request
USERTID	08	Page fault from a disabled user task or disabled B-transient	Cancel user task error code X'15'
APPENDID	0C	Page fault in I/O appendage routine	Cancel user task error code X'36'
RESVCID	10	Page fault in SVC 29 or 47	Set saved PSW to reissue SVC when task is dispatched, ENQU page fault request
DISPID	14	Page fault in a routine which requires no information to be saved, for example, a page fault in the dispatcher	ENQU page fault request
PFARID	18	Page fault in a page fault appendage routine	Cancel user task error code X'15'
	1C	RESERVERD	
	1D : 7F	Page fault in a gated supervisor service	Save PSW and registers to user task's system save area, set PIB DAT flag to call INITSVC when task is dispatched next. Close gate to routine (routine cannot be used until gate is opened). ENQU page request. (Any task trying to use a gated resource is placed in a wait state and marked resource bound. It is released from the wait state when the resource is ungated after the page request has been handled.)

VTAM ADDRESS VECTOR TABLE (ISTAVT)

Label	Length	Description
ISTACVT	4	ADDR OF VTAM CVT
ISTAS49	4	ADDR OF SCV 49 CODE
ISTAS53	4	ADDR OF SVC 53 CODE
ISTCFCSA	4	ADDR OF COMMAND HANDLER
ISTAPSEX	4	ADDR OF APS EXIT
ISTAPSTA	4	ADDR OF APS TABLE
ISTARID	4	ADDR OF RID
ISTVTTP	4	ADDR OF CODE TO CHECK FOR
ISTRETR6	4	BASE REG FOR DOS DISPATCHER
ISTRETR7	4	RETURN REG FOR DISPATCHER
ISTTTXSZ	4	
ISTVTTIK	2	VTAMRP TASK ID
ISTPHNM	5	PHASE NAME OF TRANSIENT
ISTX1	1	TOLTEP SAVES SIO COND CODE

Bytes 96–99 (X'60'–X'63') of the system communication region (SYSCOM) contain the address of VTAM Address Vector Table.

ENTRY IN THE ASYNCHRONOUS PROCESS SCHEDULER (APS) OPTION TABLE

Label	Length	Description
APSFLAG	1	X'80' Exit REQ Flag in APSFLAG
	3	Reserved
APSCNT	1	Count of VTAM ACBs open
VTAMFLAG	1	X'01' SUBTSK run key 0/SUPSTATE
		X'04' VTAM SVC active
		X'02' VTAM appendage active
		X'08' VTAM user exit in CTL
		X'10' VTAM timer delay
	3	Reserved

CHAPTER V
DOS/VSE SERVICE AIDS

OLTEP

Functions:

- Diagnosing I/O errors
- Verifying I/O device repairs and Engineering Changes
- Checking I/O devices

System Generation requirements:

None

Requirements for Execution of OLTEP:

- OLTs and CDSs available in Core Image Library
- Devices to be tested are in 'Ready'-Status
- Any real partition having at least 20 K of Storage
- If OLTEP runs in a VSE/POWER controlled partition or has to test devices being spooled by VSE/POWER, please consult the DOS/VSE-OLTEP SRL (GC33-5383) for special recommendations.

JCL to invoke OLTEP:

Statement	Comments
// JOB XXXX	Mandatory
// ASSGN SYSnnn,X'cuu'	One ASSGN statement is necessary for each device tested or accessed by a test. None is required if the device was permanently assigned.
// UPSI 01	This statement is necessary if a console device is available but the test-run definition is to be entered via the input job stream (SYSIPT).
10	This statement is necessary if READD data input is contained on diskette.
11	This statement is used when both of the above options are being used.
// EXEC IJZADOLT, REAL,[SIZE=NK]	Mandatory, OLTEP will run only in real. Size = parameter must be minimum of 20K. This will allow a 4K OLT to execute. If OLTs larger than 4K are to be run, the SIZE = parameter must specify a size equal to 16K plus the size of the OLT. The size specified must be multiple of 2K.
dev/test/opt/	This statement is included if the test-run definitions are entered via the input job stream.
/*	Mandatory
/&	Mandatory

Table of Options

Option	Entry	Description
Testing Loop	TL(n) NTL	Recycle the test. If you specify a value (n), OLTEP runs the test the number of times indicated. If you do not specify a value, the test cycles 10 times. The maximum value allowed is 32,767 decimal. (See note 1)
Error Loop	EL(n) EL(I) NEL	Authorizes any error loop coded in the OLT to be executed the specified number of times. If you specify a value (n), the test loops the number of times indicated. If you do not specify a value, the test loops the number of times indicated in the preface of the OLT. If you specify the character I, a flag is set which indicates to the OLT, that it must loop indefinitely on the error. You can terminate the loop by specifying NEL following a request for communications. (See note 1)
Print	PR NPR	Print messages from the OLT. If you enter NPR, all messages originated by the OLT and normally designated for SYSLST are supressed. (See note 2)
Error Print	EP NEP	Print diagnostic error messages from the OLT. The FE option overrides NEP when a first error is encountered (once per section). (See note 3)

Table of Options (. . . . Cont'd)

Option	Entry	Description
Control Print	CP NCP	Print OLT start and termination messages on SYSLST and SYSLOG.
Parallel Print	PP(n) NPP	Use the console device, in addition to SYSLST for OLT messages. Four levels of print are available on the parallel printer by entering one of these numbers at (n). 0: HEADER only 1: HEADER, DESCRIPTION and COMMENTS 2: HEADER and RESULTS 3: HEADER, DESCRIPTION, COMMENTS and RESULTS
First error Communications	FE NFE	Forces a communications interval when the first error is encountered. (See note 3.) A message is printed indicating the test being run and the device being tested. This is followed by the 01E105D message that allows you to: • Change the device and/or test fields. • Continue the test by entering R01,'//'/ or R01,'//(Option change)/'. • Enter any OLTEP verb. • Cancel OLTEP by entering R01,'CANCEL'. There cannot be a first error communication if a console device is not available.
Manual Intervention	MI NMI	Informs the OLT section to run all manual intervention routines within the test request. (Manual Intervention and RE are mutually exclusive options.)
Trace	TR NTR	Trace all functions called by OLT.)See note 4)
EXT=	EXT=	Information following this option is passed to the OLT section by way of a 56-byte buffer. This information must be the last entry in the option field and can contain any character but a slash. EXAMPLE: R01,'181/2400C/TL,EXT=BLOCK 4FFPRINT/' BLOCK 4FFPRINT goes into a buffer area within OLTEP and then passes on to the OLT section.

The default options are underlined.

Note 1: The FE option overrides the TL and EL options, unless NPR is also in effect. However if you enter R 01,'//'/ or R 01,'//(option change)/' at a first error communications interval, the TL and EL options, if specified, are in effect.

Note 2: NPR without EL and/or TL is ignored.

Note 3: Error print and FE are ignored if no print and either EL or TL are specified.

Note 4: Routine-to routine linkage is not traced. Do not attempt to use Trace function when SYSLST is assigned to the test device.

Example of OLTEP Operation (in BG, but can also run in foreground)

```
BG  assgn sys001,x'181'
BG  assgn sys002,x'182'
BG  // exec ijzadolt,real,size=nK (minimum 20K)
BG  E102I OLTS RUNNING
BG  E134I WARNING - DASD VOLUME LABELED CEPACK NOT
                    PROTECTED FROM WRITE
BG  E107I OPTIONS ARE NTL,NEL,NPP,FE,NMI,EP,CP,PR,NTR
BG  01E105D ENTER--DEV/TEST/OPT
BG  r 01,'181/2400a/ /'
BG  E119I NON-STANDARD TAPE LABEL 0181
BG  04E139D REPLY B TO BYPASS,R TO RETRY,P TO PROCEED (MAY DESTROY DATA)
BG  r 04,'p'
BG  E158I S T2400A $ UNIT 0181
BG  E129I FIRST ERROR COMMUNICATION T2400A 001 UNIT 0181
BG  E107I OPTIONS ARE NTL,NEL,NPP,FE,NMI,EP,CP,PR,NTR
BG  01E105D ENTER--DEV/TEST/OPT/
BG  r 01,'182//nfe,tl (2)/'
BG  E158I *T T2400A $ UNIT 0181
BG  E158I S T2400A $ UNIT 0182
BG  E158I T T2400A $ UNIT 0182
BG E 158I S T2400A $ UNIT 0182
BG  E107I OPTIONS ARE TL,NEL,NPP,NFE,NMI,EP,CP,PR,NTR
BG  01E105D ENTER--DEV/TEST/OPT
BG  R 01,'/2400c,e/ntl/'
BG  E158I *T T2400A $ UNIT 0182
BG  E158I S T2400C $ UNIT 0182
BG  E158I T T2400C $ UNIT 0182
BG  E158I S T2400E $ UNIT 0182
BG  E158I T T2400E $ UNIT 0182
BG  E107I OPTIONS ARE NTL,NEL,NPP,FE,NMI,EP,CP,PR,NTR
BG  01E105D ENTER--DEV/TEST/OPT/
BG  r 01,'cancel'
BG  1100A READY FOR COMMUNICATIONS
```

OLTEP is loaded into the back-ground partition

Initial communications interval

OLTEP finds a nonstandard labeled tape mounted on the test device

Error encountered

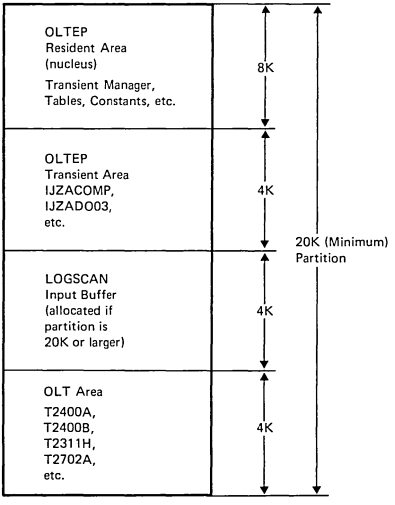
First error communications interval

Console interrupt key pressed

Interrupt communications interval

Test completion communications interval

Storage Layout



OS/VS, DOS/VSE, VM/370 - EREP
(Environmental Recording, Editing and Printing)

The service aid program IFCEREP1 can be used to retrieve all or selected records from the input data set(s), edit the records, and write them to any specified output device supported by the Sequential Access Method (SAM). A direct access device may be required for allocation of a temporary work data set.

Executing IFCEREP1

Program IFCEREP1 is the main execution routine for running EREP. The parameters may be specified only via card (SYSIPT) input.

The operator should execute the ROD command prior to running EREP from SYSREC.

Executing IFCOFFLD

Program IFCOFFLD is a special purpose load of EREP modules which is provided for clearing SYSREC, under emergency conditions, without losing the data recorded thereon.

There are no parameters allowed when executing IFCOFFLD.

- SYSREC is input.
- SYS009 is the history output data set logical unit number.
- System Summary Report is printed.
- If message IFC119I is received, alter the SIZE paramter on the // EXEC card and, if necessary, alter the partition size.

EREP Functions

EREP Basic Functions
1. Create an Accumulation data set from the SYSREC data set.
2. Clear SYSREC.
3. Copy an Input Accumulation data set to an Output Accumulation data set.
4. Merge data from an Accumulation data set and SYSREC.
5. Format Reliability Measurement data.
6. Print detail description of hardware error records.
7. Summarize and print statistics for device failures.

EREP Reporting Functions
1. System Summary Reporting
2. Trends Reporting
3. Event History Reporting
4. Media Error Statistics Reporting
5. Threshold Reporting
6. Record detail and/or Summary Reporting
7. RDE Summary Reporting
8. Offload

JCL for EREP

Statement		Usage
// JOB	EXAMPLE	This statement initiates the job.

// TLBL
or
// DLBL
// EXTENT
// ASSGN
	HISTINT
	
	HISTIND
	SYS008,XXXX,1,,XXXX,XX[1]
	SYS008,cuu
		These Job Control Statements are used to process History Input. Either History Input or SYSREC Input or both must be processed each IFCEREP1 execution. The ASSGN statement must always be used for History Input. The TLBL is used for Tape resident History Input; the DLBL and EXTENT for Disk resident History Input.

// TLBL
or
// DLBL
// EXTENT
// ASSGN
	HISTOT
	
	HISTOD
	SYS009,XXXX,1,,XXXX,XX[1]
	SYS009,cuu
		These Job Control Statements are used to create a History Output data set. The TLBL statements is used for Tape resident History Output; the DLBL and EXTENT statements are used for Disk resident History Output.

// ASSGN
// DLBL.
// EXTENT
	SYS001,cuu
	IJSYS01
	SYS001,XXXX,1,,XXXX,XX[1,2]
		These Job Control Statements are used to define the temporary work data set on a direct access device. These statements are necessary when a History Input data set is read.[3] EXTENT and DLBL statements should not be necessary as SYS001 should already be defined for the linkage editor. The standard SYS001 EXTENT should provide enough space for most IFCEREP1 executions; at any rate, enough space must be allocated to store all records selected from the input data set(s).

// EXEC	PGM=IFCEREP1,SIZE=64K	This statement specifies the program name to be executed. The minimum virtual region size for DOS/VSE is 100K. The default GETVIS area of 36K is specified.
The following system logical units are used by IFCEREP1 but should already be assigned.		
	SYSREC	The assignment for the System Recording Data Set must already be made.
	SYSLST	Both message output and report output are sent to this logical unit.
	SYSLOG	If SYSLST is unavailable, a termination message is written to the console.
	SYSIPT	IFCEREP1 input parameters and control cards are input to this required system logical unit.

[1] Ask your system programmer how to code EXTENT statements.
[2] DISKWRK must be a ONE EXTENT Data Set.
[3] RDESUM and PRINT=NO do not require the Work Data Set for History Input.

Logical Units Required by Function

Logical Units	EREP Keywords					
	ACC=Y	HIST=Y	MERGE=Y	RDESUM=Y	ALL OTHERS	HIST=N
SYS009	X					
SYS008		X	X	X		
SYS001		X	X			
SYSREC			X		X	X
SYSLST	X	X	X	X	X	X
SYSLOG	X	X	X	X	X	X
SYSIPT	X	X	X	X	X	X

EREP (. . . . Cont'd)

EREP Keywords

EREP uses Keyword parameters to determine the functions requested and any restrictions placed on the records to be processed. The default value is underlined, where applicable. Multiple parameters within one Keyword have to be seperated by commas. Keywords which have operands Y or N may be abbreviated to the Keyword only to imply the Y value, o.g. ACC implies ACC=Y. For details please consult OS/VS, DOS/VSE, VM/370-EREP (GC28-0772).

Keyword Parameters	Purpose
ACC[={Y/N}] Default exceptions are: MES, RDESUM and THRESHOLD	Accumulate selected records
CPU = (serial.model[,serial.model]) Maximum of 7 entries	Select CPU by serial number (nnnnnn) and model (nnnn)
CPUCUA = (serial.addr[,serial.addr]) Maximum of 4 entries	Select unique device addresses on a specific CPU
CUA = (entry[,entry]) Maximum of 8 entries	Select device/control units by unique addresses (nXX,nnX or nnn where n=hex digit and X=character 'X'
DATE = (yyddd[{-,}yyddd]) Single date or date range	Date span for selected records
DEV = (type[,type]) Maximum of 8 entries	Select device type (nnnn or nnXX where X = character 'X')
DEVSER = (serial[,serial]) Maximum of 8 entries	Select device serial numbers (nnnnnn), (3410/3420 devices only)
EVENT[={Y/N}]	Provide an event history report
HIST[={Y/N}] Default exception is: RDESUM	Indication for input data set
LIBADR = address	Select records by Line interface base address (XXXX - hexadecimal)
LINECT = nnn Number of lines Default = 50 lines	Number of Lines to be printed on a page (nnn - decimal)
MES[={Y/N}]	Allows selection by channel/unit address and volume serial number (3410/3420 only)
MERGE[={Y/N}]	Allows merging of EREP input (Accumulation data set + SYSREC)
MOD = (model[,model]) Maximum of 4 entries	Select specific CPU-models (nnn or nnnn)
PRINT={SU/PS/PT/NO}	Select format of printout (see Note 2)
RDESUM[={Y/N}]	Allows printing of RDE-information from IPL
SHORT[={Y/N}]	Allows suppression of detail printing of OBR-records
SYMCDE={nnn/nnX/nnXX/nXXX}	Select records by fault symptom code (33XX-DASDs only) n = hexadecimal digit X = character 'X'
SYSUM[={Y/N}]	Allows printing of a system summary report
TABSIZE = nnnK Default = 4K for DOS/VSE	Specify size of internal sort table
TERMN = name	Select records by terminal name (VTAM only)

EREP Keywords (. . . . Cont'd)

Keyword Parameters	Purpose
THRESHOLD = (xxx,yyy)	Specify threshold v alue for temporary read/ write errors (3410/3420 only) - decimal digits
TIME = (hhmm{-}hhmm)	Time span for selected records
TRENDS [={Y/N}]	Generate a Trends report
TYPE = [C] [D] [E] [I] [M] [O] [T]	Select records by their type (see note 1)
VOLID = (volser [,volser]) Maximum 4 entries	Select records by volume serial number (3410/3420/33XX only) - 1 to 6 alphameric characters
ZERO [={Y/N}]	Clear SYSREC after processing

Note 1

Record types

Code	Meaning	Selection Keywords*
C	CCH	CPUCUA,CUA
D	DDR	CPUCUA,CUA,DEV
E	EOD	
I	IPL	
M	MCH	ERRORID
O	OBR	CPUCUA,CUA,DEV,SYMCDE,TERMN,VOLID,DEVSER
T	MDR	CPUCUA,CUA,DEV,LIBADR,VOLID

*Other selection keywords apply to all record types

Note 2

SU	Suppress full printing (print summary only)
PS	Print full record and summary
PT	Suppress summary printing (print full record only)
NO	Suppress full printing and summary printing

EREP (. . . . Cont'd)

Keyword Parameter Specifications not accepted by EREP (X = Conflicting Parameters)

Keyword	ACC	CPU	CPUCUA	CUA	DATE	DEV	DEVSER	HIST	LIBADR	LINECT	MERGE	MOD	SHORT	SYMCDE	TABSIZE	TERMN	TIME	TYPE	VOLID	ZERO
ACC							x													
CPU			x									x								x
CPUCUA		x		x			x					x								x
CUA			x																	x
DATE																				x
DEV							x[1]		x[2]											x
DEVSER	x		x			x[1]			x			x	x	x		x				x
HIST											x									x
LIBADR						x[2]	x							x		x			x	x
LINECT																				
MERGE								x												
MOD		x	x				x													x
SHORT							x													
SYMCDE							x		x							x				x
TABSIZE																				
TERMN							x		x					x					x	x
TIME																				x
TYPE																				x
VOLID									x					x						x
ZERO		x	x	x	x	x	x	x	x			x		x		x	x	x	x	

Notes: [1] Devices 3410 and/or 3420 are allowed.
[2] Device 3705 is allowed.

EREP Functions with allowable Keyword Specifications (x = Allowable):

EREP Functions \ Keyword	Selection														Input	Output		Control		
	CPU	CPUCUA	CUA	DATE	DEV	DEVSER	LIBADR	MOD	SYMCDE	TERMN	TIME	TYPE	VOLID	MERGE	HIST	ACC	SHORT	LINECT	TABSIZE	ZERO
EVENT	x		x	x	x					x	x	x	x	x	x		x	x		
MES[3]	x		x	x	x[2]	x					x		x	x	x			x	x	
PRINT=PS	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x[1]
PRINT=PT	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x[1]
PRINT=SU	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x		x	x	x[1]
PRINT=NO	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x[1]
RDESUM														x	x			x		
SYSUM				x							x			x	x	x		x	x	x[1]
TRENDS			x	x	x						x	x		x	x	x		x	x	
THRESHOLD	x		x	x	x[2]	x					x		x	x	x			x	x	

Notes: [1] ZERO is acceptable only when no selective parameters are requested and a full print or Accumulation data set is generated.
[2] Devices 3410 and/or 3420 are allowed.
[3] PRINT = PT, PS, or SU is allowed.

During an execution of EREP one of the above functions is performed. The default function is PRINT=su which is underscored and generates summary reports for all data contained on SYSREC; this is the feaulted input.

EXAMPLES FOR AN EREP-RUN:
(For further examples please consult OS/VS, DOS/VSE, VM/370-EREP (GC28-0772))

1) Generating a System Summary Report from SYSREC

In this example:

- All data on SYSREC is summarized for the System Summary Report.
- All records on SYSREC are written to an Accumulation data set and SYSREC is cleared.

```
// JOB EREP
// TLBL HISTOT
// ASSGN SYS009.TAPE
// EXEC IFCEREP1
SYSUM
/*
/&
```

The TLBL and ASSGN statements define the History Output Data Set which resides on magnetic tape (Accumulation Data Set).

The messages generated by EREP and the System Summary Report will be written to the device assigned to SYSLST.

The EXEC statement specifies that EREP is to be run.

The SYSUM parameter read from SYSIPT specifies that a System Summary Report is to be generated. Effective defaults are ACC=Y and ZERO=Y; therefore, SDR records are dumped to SYSREC before data retrieval is begun.

2) Printing selected records from an Accumulation data set

In this example:

- All DDR and OBR records for 33XX-devices with specific VOLIDs are printed on SYSLST.
- Data are taken from a previously created Accumulation data set (SYS008).

```
// JOB EREP
// TLBL HISTINT
// ASSGN SYS008,TAPE
// EXEC IFCEREP1
TYPE=DO, PRINT=PS, ACC=N, DEV=(33XX)
VOLID=( SYSRES, WORK41, PVTVSE)
/*
/&
```

The TLBL and ASSGN statements define the History Input Data Set which resides on magnetic tape (Accumulation data set).

TYPE=DO specifies that DDR and OBR records are to be printed.
PRINT=PS specifies full record an summary printing.
ACC=N specifies that no accumulation has to take place.
DEV=33XX specifies that only 33XX-related records are to be printed.
VOLID=(SYSRES, . . .) specifies that only records related to the given VOLIDs are to be printed.

GLOSSARY OF ABBREVIATIONS USED IN EREP OUTPUT

BYTES RD/SRCHD	Megabytes Read/Searched
CCH	Channel Check Record
CCH-CRH	CCH-Channel Reconfiguration Hardware
CCH-INC	CCH Incomplete Record
CHNL	Channel
CK or CHK	Check
CMD or CMND	Command
CNTRL	Control
CNTRLR	Controller
COMP/MOD	Component/Module
CONS+UR	Console plus Unit Record
CORR	Correctable
CSECTID	Control Section Identification
CSW	Channel status word
C.U.	Control Unit
CUA	Channel-control unit-device address
DATA CKS CORR/RTRY	Data checks correctable/retry
DDR	Dynamic Device Reconfiguration Record
DDR-OPR	DDR-Operator Requested
DDR-SYS	DDR-System Requested
EOD	End-of-Day Record
EQUIP	Equipment
ERDS	Error Recording Data Set (SYS1.LOGREC for OS/VS, SYSREC for DOS/VSE, Recording Cylinders for VM)
ERROPS	Error Operations
FMT	Format
HDR SER	Header(tape)/serial number of drive that created tape
ID	Identification
INV	Invalid
IPL	Initial Program Load
LEN	Length
MCH	Machine Check Record
MCH-TRM	MCH-System Terminated
MDR	Miscellaneous Data Record
MIH	Missing Interrupt Handler Record
MIH-CE	MIH-Channel End Pending
MIH-DE	MIH-Device End Pending
OBR	Outboard Record
OBR-DMT	OBR-Dismount Record
OBR-EOD	OBR End of day
OBR-PRM	OBR-Permanent error record
OBR-SHT	OBR-Short Record
OBR-TMP	OBR-Temporary Error
OVERRN	Over run
OVERRUN CDDA/CDDA	Overrun Command Data/Command Data
PERM	Permanent
PRGM INT	Program Initiated
PROG-EC	Program-Extended Control Mode
PSW	Program Status Word
RCYRYXIT	Recovery Exit module
REC-TYP	Record Type
RTN	Routine
SCP	System Control Program
SEEKS CNTR/HH	Seek errors Cylinder Track/Head
SFT	Software Record
SFT-ABN	SFT-ABEND record
SFT-MCH	SFT-Machine Error, recoverable
SFT-PI	SFT-Program Interrupt
SFT-RST	SFT-Restart
SSYS ID	Sub-system Identification
TEMP	Temporary
TERM	Terminal
WRTS	Writes

Note: Most other abbreviations are meant only for the Customer Engineer, and are not meaningful to other personnel, even in translation.

SDAIDS

General

The DOS/VSE SDAID provides all of the functions that have been available with programs PDAID and SDAID under DOS/VS Releases 34 and before, plus a number of additional functions.

Requirements

- DOS/VSE
- SDAID virtual space 100 KBytes
- SDAID BASIC real space 11 KBytes

Restrictions

- Time dependent programs may not be debugged.

How to execute SDAID from the OPERATOR Console

SDAID provides a prompting facility, which will assist you in entering the control commands, needed by SDAID.
To request a prompting information, you should enter a question mark (?) in response to SDAIDs promting message. If you had made an error in control commands, key in two question marks (? ?) and hit END/ENTER. This causes SDAID cancel all control commands.

The following is an example in which SDAID is used to trace SIO activities.

SDAID will be started by entering in attention routine (AR) the command.

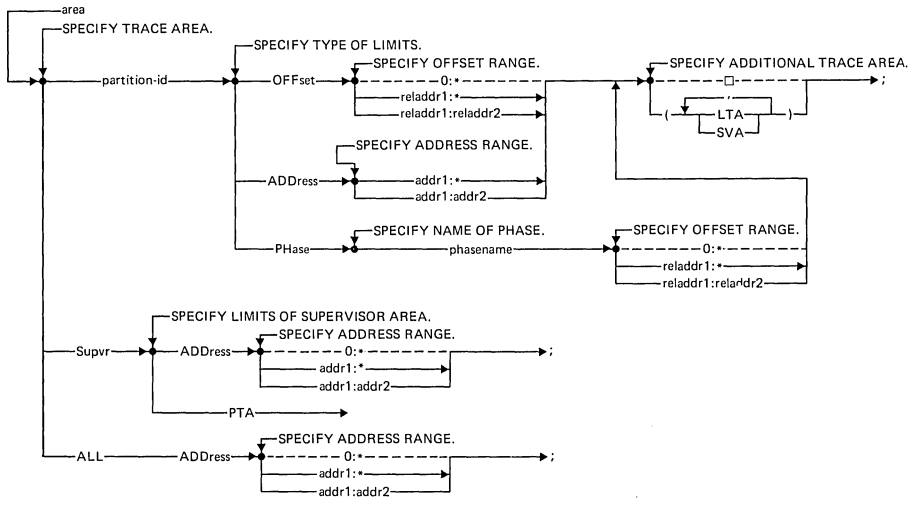
```
SDAID □
4C05I Processing of 'SDAID' Command Successful

TRACE □
4C08D Specify TRACE Type.+

SIO □
4C08D Specify Type of IO/SIO TRACE. +

UNIT □
4C08D Specify Unit Address(es). +

02C □
4C08D Specify Output. +

CCW □
4C08D Specify Occurrence Range. +

□
4C08D Specify 'HALT' or 'NOHALT' or Press END/ENTER.

□
4C08D Specify Termination Options. +

□
4C05I Processing of 'TRACE' Command Successful

OUTDEV P=E □
4C05I Processing of 'OUTDEV' Command Successful

READY □
4C05I Processing of 'READY' Command Successful

STRTSD □
4C05I Processing of 'STRTSD' Command Successful
```

Note: □ = Operator Presses END/ENTER Key.

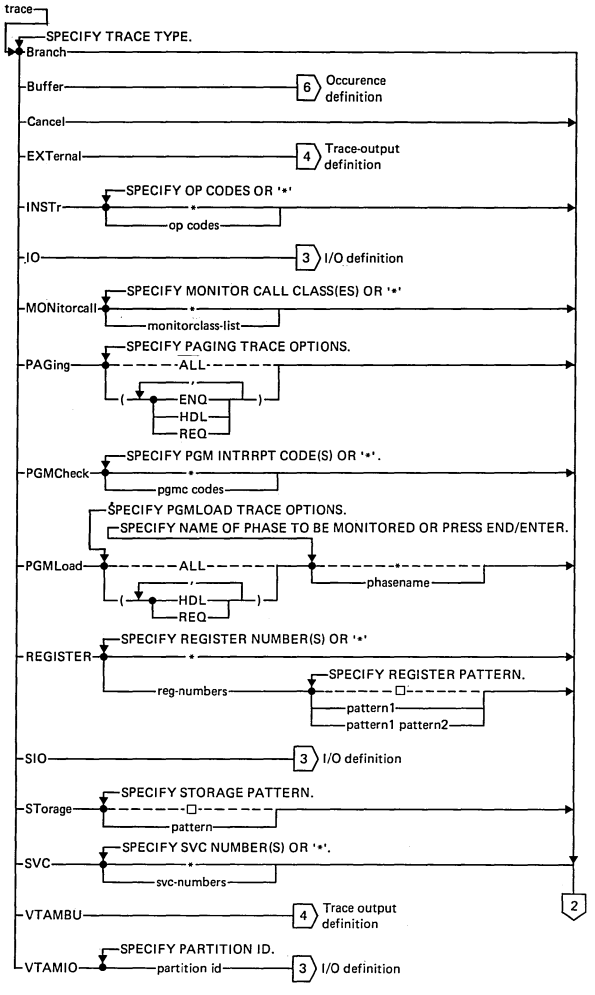
Command	Summary of purpose	Add'l info requested by SDAID: yes	Add'l info requested by SDAID: no
AREA	Establishes the address range within which the occurence of the specified event (s) is (are) to be traced. An AREA command may be overridden by the area definition of a TRACE command, but only for the event(s) specified in that TRACE command. If, for an SDAID session, no AREA command is submitted, you must provide an area definition in each of your TRACE commands for the session.	x	
OUTDEV	Establishes the method of trace information output. The command is mandatory.	x	
TRACE	Establishes the program event(s) to be traced. At least one TRACE command must be entered per SDAID session; up to ten different trace commands may be submitted per SDAID session. The command is mandatory.	x	
READY	Ends SDAID initiation command input (AREA, OUTDEV, and TRACE commands, which must precede the READY command). The command is mandatory.		x
STRTSD	Starts SDAID execution. The command may follow the READY command or a STOPSD command, if one was entered. The command is mandatory.		x
STOPSD	Stops SDAID execution. The command is optional. If entered, it must follow a STRTSD command.		x
ENDSD	Ends SDAID execution. The command must be issued at the end of an SDAID session; it requests SDAID to release all system resources that the program used during the preceding session.		x

Summary of available SDAID commands

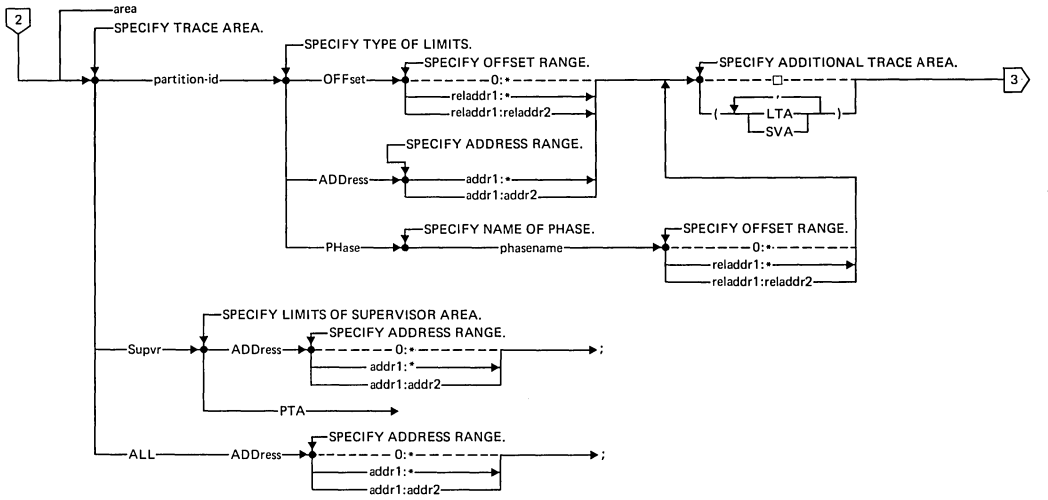
Note: For more information and detailed description of SDAID refer to the DOS/VSE Serviceability Aids and Debugging Procedures.



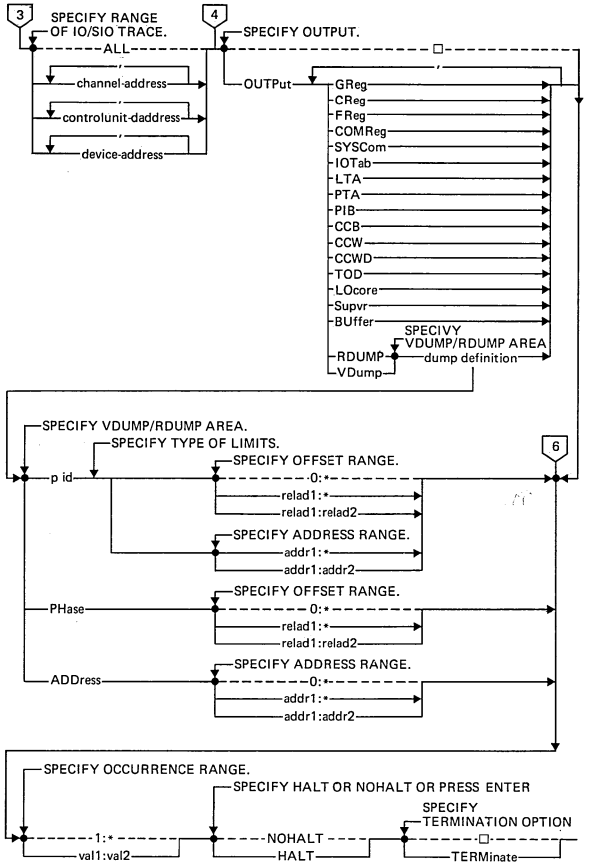
Specification path diagram for the AREA command



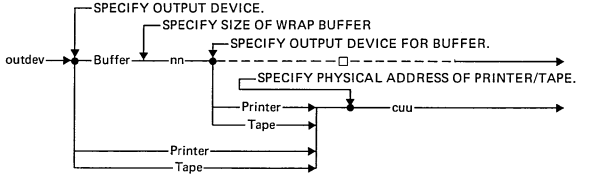
Specification path diagram for the TRACE command



Specification path diagram for the TRACE command



Specification path diagram for the TRACE command



Specification path diagram for the OUTDEV command

TABLE OF CONTENTS

THE PURPOSE OF MSHP

MSHP, an integral part of the DOS/VSE System Control Programming (DOS/VSE SCP), provides functions to:

- Install programming support.
- Apply corrections to existing programming support.
- Record installation and service activities in a system history file in order to reflect the current status of your operational system.
- Automatically compare on operational system's history with IBM-supplied information on requirements for installing additional programming support or applying corrections to the installed programming support.
- Print various types of listings based on your system's history.

For an effective system service, it is important to use the functions available through MSHP. Function Control Statements and Detail Control Statements not described in this Handbook are for internal use of IBM Software Support Personnel.

INSTALLATION SUPPORT BY MSHP

Prerequisites for installing different types of program packages vary. IBM program packages are designated as:

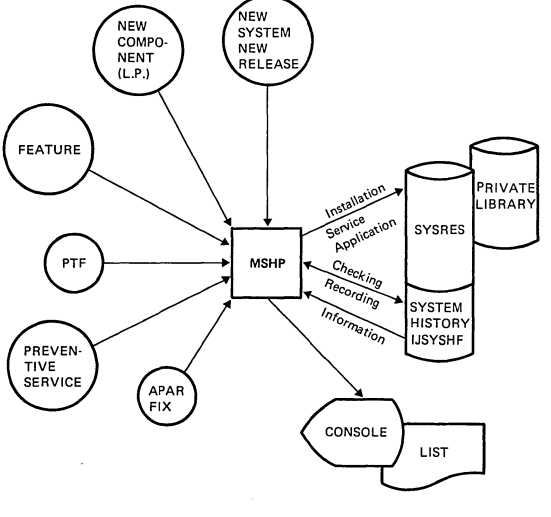
- System (for example, DOS/VSE SCP)
- Product (for example, System Installation Productivity Offering)
- Component (for example, DOS/VS RPG II)
- Feature (for example, VSE/Advanced Functions).

You are informed in the transmittal letter of the type of programming support the shipment contains, and can then proceed with the installation in accordance with the applicable MSHP procedure provided in this Handbook.

THE NEED FOR A SYSTEM HISTORY

For an efficient and successful installation of a program package, it is essential that the history file of your operational system truly reflects the status of the system, and that prerequisite programming support requirements are met.

MSHP Processing Overview



INSTALLING A SYSTEM

The DOS/VSE SCP is distributed on magnetic tape or disk; the system's change status is reflected in its accompanying system history file. Depending on the configuration of your system, you install DOS/VSE SCP standalone or online.

- Standalone

If your system operates with DOS/VS Release 34 or an earlier release level, MSHP is not yet part of your system. Consequently, the MSHP-provided installation support is not available to you. For initial installation of DOS/VSE SCP, it is therefore necessary that you install the system standalone.

Standalone installation also applies if your system is configured without DOS/VS or DOS/VSE.

For detailed information on standalone installation/generation, refer to DOS/VSE System Generation.

- Online — see Installation Activities: INSTALL SYSTEM ONLINE

THE HISTORY FILES

MSHP uses two types of history files:

- The System History File
- The Auxiliary History File

The system history file is part of the IBM-distributed system and is maintained under the file name IJSYSHF on the logical unit SYSREC. The file should be defined by a permanent entry in the system standard label area:

```
// DLBL IJSYSHF, 'DOS.SYSTEM.HISTORY.FILE',99/365
// EXTENT SYSREC,serial-number,1,0,relative track,number of tracks
```

The above statements define an area for the system history file; this area is expiration-date protected just like the SYSRES file.

The auxiliary history file is maintained under the file name IJSYS02 on the logical unit SYS002. If an auxiliary history file is carried over from one system to another code the following MSHP control statements to have MSHP create label definitions in the label area:

```
DEFine History EXTent=number1:number2 Unit=SYSnnn —
    ID='DOS.AUXILIARY.HISTORY.FILE'
```

INSTALLATION ACTIVITIES

Install System Online

Online installation can be done only under the control of an operational DOS/VSE that includes MSHP and a current system history.

If your installation does not include magnetic tape drives, you may produce a disk volume as shown under "Prepare for Tapeless Configurations".

You may save your installations current history by executing the job given under "Save Your Current System History File".

You may want to verify the installation. Use the FETRACE function of MSHP as shown under "Obtain an Installation Verification List"; execute the job on completion of the installation steps.

Prepare for Tapeless Configurations (New System)

This procedure may be used to produce, from the IBM-supplied magnetic tape that contains DOS/VSE, a disk volume that can then be used to install the system from disk.

```
// JOB RESTORE DISTRIBUTION TAPE
// ASSGN SYS006,cuu                    Input unit for distribution tape
// ASSGN SYS005,cuu                    Disk pack
// DLBL IJSYSRS,'DOS.SYSRES.FILE.'99/365,SD
// EXTENT SYSRES,SYSRES, . . .
// EXEC RESTORE
   ALLOC CL=ccc(tt),RL=ccc(tt),SL=ccc(tt),PL=ccc(tt)
/*
// ASSGN SYS002,cuu                    History file space
// EXEC MSHP
RESTORE HIST AUX
DEF HIST AUX EXT= number1:number2       History file copy on disk
    ID='DOS.SYSTEM.HISTORY.FILE'
/*
/&
```

INSTALLATION ACTIVITIES (. . . . Cont'd)

Prepare for Tapeless Configuration (Component)

This procedure may be used to produce, from the IBM-supplied magnetic tape that contains DOS/VSE, a disk volume that can then be used to install the component from disk.

```
// JOB RESTORE DISTRIBUTION TAPE
// ASSGN SYS006,cuu                Input unit for distribution tape
// ASSGN SYS007, cuu               Output unit for PCL
// ASSGN SYS008,cuu                Output unit for PRL
// ASSGN SYS009,cuu                Output unit for PSL
// ASSGN SYS002,cuu                Auxiliary history file
// EXEC MSHP
RESTORE DTAPE
DEF CLIB PRIV EXT=number1:number2 DIR=number3
DEF RLIB PRIV EXT=number1:number2 DIR=number3
DEF SLIB PRIV EXT=number1:number2 DIR=number3
DEF HIST AUX EXT=number1:number2
/*
/&
```

Save Your Current System History File

It is recommended to keep a copy of the current system history file for future control purposes; you may use the BACKUP or the COPY function of MSHP. Enter the following statements before commencing with Step 1 of the installation procedure.

```
// JOB SAVE HISTORYFILE
// EXEC MSHP
COPY HISTORY SYSTEM AUXILIARY
DEFINE HIST AUX EXT=number1:number2.
/*
/&
```

Obtain an Installation Verfication List

To obtain the printout (on SYSLST), execute:

```
// JOB RETRACE
// EXEC MSHP
RETR              If a system overview is desired
RETR COMP         If a retrace on components is desired
RETR FEAT         If a retrace on features is desired
/*
/&
```

PROCEDURE 1: NEW SYSTEM TO CURRENT SYSRES FROM TAPE

All install functions must be executed in the background partition (BG).
Instead of using this procedure, it is recommended to use procedure 3 ("New System to New SYSRES from Tape") with a backup and restore step added at the end of that procedures. The following shows the backup job that would be required; it produces a standalone restore version. For the standalone restore job, refer to DOS/VSE System Generation.

```
// JOB BACKUP
// ASSGN SYS006,cuu                Tape
// ASSGN SYS005,cuu                New SYSRES
// DLBL IJSYSRS,'DOS.SYSRES.FILE'
// EXTENT SYS005
// EXEC BACKUP
   SA                              Standalone restore
/&
```

INSTALLATION ACTIVITIES (. . . . Cont'd)

Step 1. Restore the Distribution Tape to Disk

Mount the IBM-supplied distribution tape on the device assigned to SYS006. Then execute the following job.

```
// JOB INSTALL NEW SYSTEM
// ASSGN SYS006,cuu                    Distribution tape
// ASSGN SYS005,cuu                    Disk for new SYSRES (see Note 1)
// ASSGN SYS002,cuu                    Auxiliary history file
// ASSGN SYS007,UA
// ASSGN SYS008,UA
// ASSGN SYS009,UA
// EXEC MSHP
INSTALL SYSTEM FROMTAPE
DEF CL SYS EXT=:number2 DIR=number3
DEF RL SYS EXT=:number2 DIR=number3
DEF SL SYS EXT=:number2 DIR=number3
DEF PL SYS EXT=:number2 DIR=number3
DEF LABELAREA EXT=number2           May be specified for FBA DASDs
DEF HIST AUX EXT=number1:number2 —
    ID='DOS.AUX.HISTORY.FILE'
/*
/&
```

Note 1: The disk for the new SYSRES need not be the same as the disk for the SYSRES of the running system, however, it must be of the same device type.

On completion of Step 1:

- The system libraries have been restored from the distribution tape to the disk assigned as the new SYSRES volume.
- The history file supplied on the distribution tape has been restored to the auxiliary history file (assigned as SYS002) on the disk assigned as the new SYSRES volume; the history file of your current system now reflects the status of the new system nerged with the old system; the components of the old system are flagged as obsolete.

Step 2. System Generation Activities (Procedure 1)

This step consists of a number of system generation activities such as:

- Assemble and catalog the SUPERVISOR
- Delete unwanted components (see Note below)
- Assemble and catalog I/O modules

Note:
If a delete procedure deletes only part of a component, that component's entry is not removed from the history file. For details or system generation activities, refer to DOS/VSE System Generation.

Step 3. Merge the Restored Libraries

For this Step all Foreground partitions (FG's) must be inactive. Ensure that user — written programs with IBM names such as $JOBACCT or $SYSOPEN are saved or renamed. Note that the Label information cylinder will also be copied; it may, therefore, be necessary to run the system standard labels again.

```
// JOB MERGE NRS to RES
// ASSGN SYS002,cuu
// DLBL IJSYSRS, 'DOS.SYSRES.FILE'
// EXTENT SYS002
// EXEC CORGZ
   MERGE NRS,RES
   COPY ALL
/*
/&
```

Step 4. Remove Obsolete Entries

Execute this tep if you want that the history file reflects exactly your generated system:

```
// JOB REMOVE BACKLEVEL RELEASE ENTRIES
// EXEC MSHP
REM OBSOLETE (see Note)
/*
/&
```

Note:
To insure that valid history information is not removed from the history file, do not specify any other remove options other than OBSOLETE.

INSTALLATION ACTIVITIES (. . . Cont'd)

Step 5. Personalize the System History

If the originally defined personalize record for the history file should reflect the new installation environment (ENV=), changes can be made by using a job example like the following:

```
// JOB PERSONAL
// EXEC MSHP
PERS='customer-name' –
ADDR='Location'      –
PHONE='Extension'    –
PROG='Programmer's name' –
ENV='Environment'
/*
/&
```

This function updates the newly installed history file header records.

PROCEDURE 2: NEW SYSTEM TO CURRENT SYSRES FROM DISK

This procedure is identical to procedure 1, except for STEP 1 of that procedure, which must be replaced by the following job.

Step 1. Install the Distribution Disk Volume

For Installation of the IBM-supplied distribution disk volume(s), execute the job example:

```
// JOB INSTALL NEW SYSTEM
// ASSGN SYS002,cuu                    Auxiliary history file
// EXEC MSHP
INSTALL SYSTEM FROM DISK
DEF HIST AUX EXT= number1:number2 –
    ID='PID.supplied.id'
/*
/&
```

On completion of STEP 1 the history file of the current system then reflects the status of the new system merged with the old system. The components of the old system are flagged as absolete.
At completion of this STEP 1, continue with STEP 2 through 5 of procedure 1.

PROCEDURE 3: NEW SYSTEM TO NEW SYSRES FROM TAPE

Some steps of this procedure are identical to procedure 1; refer to that procedure as indicated below.

Step 1. Restore the Distribution Tape to Disk

This step is identical to STEP 1 of procedure 1.

Step 2. System Generation Acitivites

This step is identical to STEP 2 of procedure 1.

Step 3. Merge user Program

This step may be used to merge user and other programs (for example, licensed IBM programs) from the current system to the new SYSRES.

```
// JOB MERGE RES TO NRS
// ASSGN SYS002,cuu
// DLBL IJSYSRS,'DOS.SYSRES.FILE'
// EXTENT SYS002
// EXEC CORGZ
  MERGE RES,NRS
  COPYC NEW
  COPYR NEW
  COPYS NEW
  COPYP NEW
/*
/&
```

The COPY NEW Function copies only those members that do not already exist in the receiving Library.

INSTALLATION ACTIVITIES (. . . . Cont'd)

Step 4. Copy the history file to the NEW SYSRES

The updated system history file from the old pack (RES) is copied to the new SYSRES pack (NRES) and obsolete entries are removed by the following job example:

```
// JOB COPY SYSTEM HISTORY
// ASSGN SYS002,cuu
// EXEC MSHP
REM OBSOLETE
COPY HISTORY SYSTEM AUXILIARY
DEFINE HISTORY AUXILIARY EXTENT=number1: number2 –
     ID='DOS.SYSTEM.HISTORY.FILE'
/*
/&
```

Step 5. IPL

5.1. IPL from the new system.
5.2. Apply standard labels to the system
If IBM standard labels are not used the following statements should be contained in the labeljob:

```
// DLBL IJSYSHF,'DOS.SYSTEM.HISTORY.FILE',99/365,SD (see Note 1)
// EXTENT SYSREC,SYSRES,1,0,number1,number2 (see Note 2)
```

Note 1:
The label information must be identical with the ID specification given in STEP 4.
Note 2:
number 1, number 2 in the EXTENT statement must be identical with number 1: number2 of the extent specified in the DEFINE statement of STEP 4.
5.3. ENTER the command SET RF=CREATE

Step 6. Personalize the System History

```
// JOB PERSONAL
// EXEC MSHP
PERS='Company name' –
ADDR='Location' –
PHONE='Extension' –
PROG='Programmer's name' –
ENV='Environment'
/*
/&
```

This function updates the newly installed history file header records.
The PERSONALIZE function may be used to change any information contained in the header records. MSHP updates the field(s) selected by the PERSONALIZE keyword operands.

Step 7. BACKUP the System

```
// JOB BACKUP
// ASSGN SYS006,cuu                    Tape
// ASSGN SYS005,cuu                    New SYSRES
// DLBL IJSYSRS,'DOS.SYSRES.FILE'
// EXTENT SYS005
// EXEC BACKUP
SA                                     Standalone restore
/*
/&
```

For the standalone restore Job, refer to DOS/VSE System Generation.

PROCEDURE 4: NEW SYSTEM TO NEW SYSRES FROM DISK

This procedure is identical to procedure 3 except that in Step 1 (as shown in procedure 1) you:

- Omit the assignments for SYS005 and SYS006
- Code FROMDISK (instead of FROMTAPE) in the MSHP INSTALL statement.
- Omit the Library definitions for CL, RL, SL, and PL.
- Omit the define statement for the label area.
- In the define statement for the auxiliary history file, use the ID provided on the PID volume.

INSTALLATION ACTIVITIES (. . . . Cont'd)

STANDALONE PROCEDURE (Example of Console output)

```
STEP 1:
* Mount Backup or Distr. tape to ADDR cuu and make IPL from it.
* When SYSTEM enters WAIT STATE push ATT KEY at Console.
***** STAND ALONE PROGRAMS LOADED *****
IF YOU WANT A LISTING.SPECIFY CUU OF PRINTER
IF NOT, PRESS EOB.
PRESS EOB ALSO, IF BUFFER NOT OR INCORRECTLY LOADED
00e
SPECIFY TYPE OF PRINTER XXXXYY
3211
SPECIFY DATE  MM/DD/YY
01/11/79
SPECIFY ONE OF THE FOLLOWING COMMANDS:
FASTCOPY, INITDISK, RESTORE, INITEM, SURFANAL, END
restore
SPECIFY ADDRESS OF INPUT DEVICE CUU
281
SPECIFY TYPE OF INPUT DEVICE  XXXXYY
3420t9
SPECIFY ADDRESS OF SYSRES DISK  CUU  OR EOB
(PRESS EOB, IF ONLY PRIVATE LIBRARIES ARE TO BE RESTORED)
144
SPECIFY TYPE OF DISK  XXXXYY
3330
ANY PRIVATE LIBRARY TO BE RESTORED ?  YES/NO
no
8R430  TYPE NOVERIFY OR PRESS ENTER FOR WRITE VERIFICATION
noverify
8R01D  *** GIVE SYSTEM LIBRARY ALLOCATIONS ***
CL=
8R03I  ALLOC DEFAULTS TO 57(9)
RL=
8R03I  ALLOC DEFAULTS TO 64(5)
SL=
8R03I  ALLOC DEFAULTS TO 58(2)
PL=
8R03I  ALLOC DEFAULTS TO 2(2)
8R12D  TYPE DESIRED LABEL FOR LIBRARY
LABEL=
8R14I  FILE ID = DOS.SYSRES.FILE
8R15D  TYPE GO IF ALLOCATION IS CORRECT
go
8R19I  EQUAL FILE ID IN VTOC
DOS.SYSRES.FILE
8R20D  TYPE DELETE OR GIVE A NEW FILE ID
delete
8R14I  FILE ID = DOS.SYSRES.FILE
8R13I  EXTENT = TRK 1 - TRK 3476
8R35I  RESTORE OF SC LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R35I  RESTORE OF SR LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R35I  RESTORE OF SS LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R35I  RESTORE OF SP LIBRARY IN PROGRESS
8R36I  RESTORE HAS BEEN SUCCESSFUL
8R37I  *** SYSRES RESTORED ***          (Now the Distribution Tape is
8R38I  *** RESTORE COMPLETE ***          positioned at the SYSTEM HISTORY
*** END OF STAND ALONE PROCESSING ***    FILE, ready for RESTORE STEP 3)
```

INSTALLATION ACTIVITIES (. . . . Cont'd)

```
STEP 2:
* IPL from NEW SYSTEM (EXAMPLE ADDR = 144)
* APPLY LABELS30 INCL IJSYSHF'DOS.SYSTEM.HISTORY.FILE'
* ENTER SET RF=CREATE
STEP 3:
* RESTORE SYS HISTORY FILE FROM DISTR TAPE
// JOB RESTORE SYS HIST
// ASSGN SYS006,280            DISTR TAPE
// EXEC MSHP
RESTORE HIST SYS
/*
// MTC RUN,SYS006
/&
STEP 4:
* This Step consists of several SYSGEN Steps
* 1. Delete UNWANTED components
* 2. Assemble and catalog your own SUPVR
* 3. Create a BACKUP
* 4. Restore the SYSTEM
* 5. Include USER Programs and LP's
* 6. Perform DSERV and LSERV
STEP 5:
* Identical to Step 6 of procedure 3
STEP 7:
* CHECK the INSTALLATION by calling a HISTORY DUMP
// JOB DUMP HIST
// EXEC MSHP
RETR COMP
DUMP H
/*
/&
```

INSTALL TOTAL COMPONENT TO SYSRES

Install from Tape or from Disk

To install from tape, use the following job:

```
// JOB INSTALL COMPONENT FROM TAPE
// ASSGN SYS006,cuu          Distribution tape
// ASSGN SYS007,cuu          Output unit for PCL
// ASSGN SYS008,cuu          Output unit for PRL
// ASSGN SYS009,cuu          Output unit for PSL
// ASSGN SYS003,cuu          From unit for CORGZ PCL
// ASSGN SYS001,cuu          From unit for CORGZ PRL
// ASSGN SYS000,cuu          From unit for CORGZ PSL
// ASSGN SYS002,cuu          Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST COMP FROMTAPE MERGE
DEF CLIB PRIV EXT=number1:number2 DIR=number3
DEF RLIB PRIV EXT=number1:number2 DIR=number3
DEF SLIB PRIV EXT=number1:number2 DIR=number3
DEF HIST AUX EXT=number1:number2
/*
/&
```

On completion of this step (and after check and verification procedures are satisfied), the IBM-supplied total component is part of the current system library, and the history file of the total component is part of the current history file.

INSTALL TOTAL COMPONENT TO SYSRES (. . . . Cont'd)

To install from disk, use the following job:

```
// JOB INSTALL COMPONENT FROM DISK
// ASSGN SYS003,cuu          From-unit for CORGZ PCL distribution disk
// ASSGN SYS001,cuu          From-unit for CORGZ PRL distribution disk
// ASSGN SYS000,cuu          From-unit for CORGZ PSL distribution disk
// ASSGN SYS002,cuu          Auxiliary history file distribution disk
// OPTION CATAL
// EXEC MSHP
INST COMP FROMDISK MERGE
DEF CLIB PRIV EXT=number1:number2
DEF RLIB PRIV EXT=number1:number2
DEF SLIB PRIV EXT=number1:number2
DEF HIST AUX EXT=number1:number2
/*
/&
```

On completion of this step the IBM-supplied total component is part of the current system library, and the history file of the total component is part of the current system history file.

INSTALL TOTAL COMPONENT TO PRIVATE LIBRARIES

Refer to these procedures if you want to keep the total component on private libraries.

Install from Tape or from Disk

To install from tape, use the following job:

```
// JOB INSTALL TOTAL COMPONENT FROM TAPE
// ASSGN SYSRLB,cuu  }
// ASSGN SYSSLB,cuu  }       Existing private libraries
ASSGN SYSCLB,cuu     }
// ASSGN SYS006,cuu          Distribution tape
// ASSGN SYS007,cuu          Output unit for PCL
// ASSGN SYS008,cuu          Output unit for PRL
// ASSGN SYS009,cuu          Output unit for PSL
// ASSGN SYS003,cuu          From-unit for CORGZ PCL distribution disk
// ASSGN SYS001,cuu          From-unit for CORGZ PRL distribution disk
// ASSGN SYS000,cuu          From-unit for CORGZ PSL distribution disk
// ASSGN SYS002,cuu          Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST COMP FROMTAPE MERGE
DEF CLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRIV.CLIB'
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=numb er1:number2 DIR=number3 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2
/*
/&
ASSGN SYSCLB,UA
```

At completion of this step, the component is on the private libraries as defined and the system history file reflects the installation of the component.

To install from disk, use the following job:

```
// JOB INSTALL TOTAL COMPONENT FROM DISK
ASSGN SYSCLB,cuu          To-unit for CORGZ PCL
// ASSGN SYSRLB,cuu       To-unit for CORGZ PRL
// ASSGN SYSSLB,cuu       To-unit for CORGZ PSL
// ASSGN SYS003,cuu       From-unit for CORGZ PCL distribution disk
// ASSGN SYS001,cuu       From-unit for CORGZ PRL distribution disk
// ASSGN SYS000,cuu       From-unit for CORGZ PSL distribution disk
// ASSGN SYS002,cuu       Auxiliary history file distribution disk
// OPTION CATAL
// EXEC MSHP
INST COMP FROMDISK MERGE
DEF CLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.CLIB'
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 ID='pid.supplied.id'
/*
/&
ASSGN SYSCLB,UA
```

INSTALLATION ACTIVITIES (. . . . Cont'd)

INSTALL PART COMPONENT TO SYSRES

Some IBM program packages consist of an unlicensed portion of code (the "base part") plus the licensed portion of code (the "second part").

To install a "base" part, only Step 1 of this procedure needs to be executed; if the "base" part and the "second" part are to be installed, Step 1 and Step 2 must be executed.

Install from Tape or from Disk

Step 1. Install Base Part

To install a base part from tape, use the following job:

```
// JOB INSTALL BASE COMPONENT PART FROM TAPE (1)
// ASSGN SYS006,cuu    (2)          Distribution tape
// ASSGN SYS007,cuu    (2)          Output unit for PCL
// ASSGN SYS008,cuu    (2)          Output unit for PRL
// ASSGN SYS009,cuu    (2)          Output unit for PSL
// ASSGN SYS003,cuu                 From-unit CORGZ PCL
// ASSGN SYS000,cuu                 From-unit CORGZ PSL
// ASSGN SYS001,cuu                 From-unit CORGZ PRL
// ASSGN SYS002,cuu                 Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST COMP FROMTAPE MERGE    (3)
DEF CLIB PRIV EXT=number1:number2 DIR=number3
DEF RLIB PRIV EXT=number1:number2 DIR=number3
DEF SLIB PRIV EXT=number1:number2 DIR=number3
DEF HIST AUX EXT=number1:number2
/*
/&
```

To install a base part from disk, change the above job as follows:

- Change in statement (1) the words "FROM TAPE" to read "FROM DISK".
- Omit the assignments indicated by (2).
- Change in statement (3) the word "FROMTAPE" to read "FROMDISK".
- Omit DIR=number3 in the library definition statements.

Step 2. Install Second Part

To install a second part from tape, use the following job:

```
// JOB INSTALL SECOND COMPONENT PART FROM TAPE     (1)
// ASSGN SYS006,cuu    (2)          Distribution tape
// ASSGN SYS008,cuu    (2)          Output unit for PRL
// ASSGN SYS009,cuu    (2)          Output unit for PSL
// ASSGN SYS000,cuu                 From-unit CORGZ PSL
// ASSGN SYS001,cuu                 From-unit CORGZ PRL
// ASSGN SYS002,cuu                 Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEATURE FROMTAPE MERGE     (3)
DEF RLIB PRIV EXT=number1:number2 DIR=number3
DEF SLIB PRIV EXT=number1:number2 DIR=number3
DEF HIST AUX EXT=number1:number2
/*
/&
```

To install a second part from disk, use the same job as in Step 2 except for the following changes:

- Replace statement (1) by:
  // JOB INSTALL SEC COMP PART FROM DISK
- Omit the assignments indicated by (2).
- Replace statement (3) by:
  INSTALL FEATURE FROMDISK MERGE
- Omit DIR=number3 in the library definition statements.

INSTALLATION ACTIVITIES (. . . . Cont'd)

15. INSTALL PART COMPONENT TO NEW PRIVATE LIBRARIES

Some IBM program packages consist of an unlicensed portion of code (the "base part") plus the licensed portion of code (the "second part").

To install a "base" part, only Step 1 of this procedure needs to be executed; if the "base" part and the "second" part are to be installed, Step 1 and Step 2 must be executed.

Install from Tape or from Disk

Step 1. Install Base Part

To install a base part from tape, use the following job:

```
// JOB INSTALL BASE COMPONENT PART FROM TAPE
// ASSGN SYS006,cuu              Distribution tape
// ASSGN SYS007,cuu              Output unit for PCL
// ASSGN SYS008,cuu              Output unit for PRL
// ASSGN SYS009,cuu              Output unit for PSL
// ASSGN SYS002,cuu              Auxiliary history file
// EXEC HSHP
INST COMP FROMTAPE ATTACH
DEF CLIB PRIV EXT=number1:number2 DIR=number3 ID='NEW.PCL'
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='NEW.PRL'
DEF SLIB PRIV EXT=number1:number2 DIR=number3 ID='NEW.PSL'
DEF HIST AUX EXT=number1:number2
/*
/&
```

To install a base part from disk, use the following job:

```
// JOB INSTALL BASE COMPONENT PART FROM DISK
// ASSGN SYS002,cuu              Auxiliary history file
// EXEC MSHP
INST COMP FROMDISK ATTACH
DEF HIST AUX EXT=number1:number2
/*
/&
```

It is assumed that standard labels are applied for the new private libraries before executing this step.

Step 2. Install Second Part

To install a second part from tape, use the following job:

```
// JOB INSTALL SECOND COMPONENT PART FROM TAPE        (1)
// ASSGN SYS006,cuu      (2)          Distribution tape
// ASSGN SYS008,cuu      (2)          Output unit for PRL
// ASSGN SYS009,cuu      (2)          Output unit for PSL
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSRL,'NEW.PRL'
// EXTENT SYSRLB, ,1,0,number1,number2   } Same as in the
// DLBL IJSYSSL,'NEW.PSL'                } define statements
// EXTENT SYSSLB,,1,0,number1,number2    } in Step 1.
// DLBL IJSYSCL,'NEW.PCL'
// EXTENT SYSCLB,,1,0,number1,number2
ASSGN SYSCLB,cuu
// ASSGN SYS000,cuu  } From-libraries for CORGZ
// ASSGN SYS001,cuu  }
// ASSGN SYS002,cuu          Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEATURE FROMTAPE MERGE        (3)
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRL'
DEF SLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PSL'
DEF HIST AUX EXT=number1:number2
/*
/&
```

To install a second part from disk, use the same job as in Step 2 except for the following changes:

- Replace statement (1) by:
  // JOB INSTALL SEC COMP PART FROM DISK
- Omit the assignments indicated by (2).
- Replace statement (3) by:
  INSTALL FEATURE FROMDISK MERGE
- Omit the DIR=number3 in the library define statements.

INSTALLATION ACTIVITIES (. . . . Cont'd)

INSTALL PART COMPONENT TO EXISTING PRIVATE LIBRARIES

Some IBM program packages consist of an unlicensed of code (the "base part") plus the licensed portion of code (the "second part").

To install a "base" part, only Step 1 of this procedure needs to be executed; it the "base" part and the "second" part are to be installed, Step 1 and Step 2 must be executed.

Install from Tape or from Disk

Step 1. Install Base Part

To install a base part from tape, use the following job:

```
// JOB INSTALL BASE COMPONENT PART FROM TAPE    (1)
// ASSGN SYS006,cuu     (2)          Distribution tape
// ASSGN SYS007,cuu     (2)          Output unit for PCL
// ASSGN SYS008,cuu     (2)          Output unit for PRL
// ASSGN SYS009,cuu     (2)          Output unit for PSL
ASSGN SYSCLB,cuu
// ASSGN SYSRLB,cuu                  Target private libraries
// ASSGN SYSSLB,cuu
// ASSGN SYS003,cuu
// ASSGN SYS000,cuu                  From-libraries (see Note 1)
// ASSGN SYS001,cuu
// ASSGN SYS002,cuu                  Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST COMP FROMTAPE MERGE    (3)
DEF CLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PCL'
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRL'
DEF SLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PSL'
DEF HIST AUX EXT=number1:number2
/*
/&
```

Note 1:
The logical units for the from-libraries are identical with output units SYS007, SYS008 and SYS009.

To install a base part from disk, use the same job as in Step 2 except for the following changes:

- Change in statement (1) the words "FROM TAPE" to read "FROM DISK".
- Omit the assignments indicated by (2).
- Change in statement (3) the word "FROMTAPE" to read "FROMDISK".
- Omit DIR=number3 in the library definition statements.

Step 2. Install Second Part

To install a second part from tape, use the following job:

```
// JOB INSTALL SECOND COMPONENT PART FROM TAPE    (1)
// ASSGN SYS006,cuu     (2)          Distribution tape
// ASSGN SYS008,cuu     (2)          Output unit for PRL
// ASSGN SYS009,cuu     (2)          Output unit for PSL
ASSGN SYSCLB,cuu
// ASSGN SYSRLB,cuu                  Target private libraries
// ASSGN SYSSLB,cuu
// ASSGN SYS000,cuu                  From-libraries
// ASSGN SYS001,cuu
// ASSGN SYS002,cuu                  Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEATURE FROMTAPE MERGE    (3)
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRL'
DEF SLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PSL'
DEF HIST AUX EXT=number1:number2
/*
/&
```

INSTALLATION ACTIVITIES (. . . . Cont'd)

To install a second part from disk, use the same job as in Step 2 except for the following changes:

- Replace statement (1) by:
  // JOB INSTALL SEC COMP PART FROM DISK
- Omit the assignments indicated by (2).
- Replace statement (3) by:
  INSTALL FEATURE FROMDISK MERGE
- Omit the DIR=number3 in the library define statements.

INSTALL A COMPONENT IN SYSIN FORMAT (from Tape or Disk)

Step 1. Deblock

```
// JOB DEBLOCK TAPE
* Mount the input and output tape
// ASSGN SYS004,cuu                Input tape
// ASSGN SYS005,cuu                Output tape
// EXEC OBJMAINT
*/ LIST PARM=JOB                   List jobs on input tape (optional)
*/ DEBLOCK                         Deblock input tape to SYSIN format
/*
/&
```

Step 2. Execute the SYSIN Job

```
// ASSGN SYSIN,cuu                 "Tape" for install from tape
                                   "Disk" for install from disk
```

INSTALL A FEATURE TO SYSRES

The MSHP function INSTALL FEATURE is used to install IBM program packages designated as a feature.

The following procedures apply when installing a feature from tape or disk to SYSRES. When installing a feature, either of the MSHP functions MERGE or ATTACH may be used; however, if the feature contains transient modules, MERGE should be used.

Install from Tape or Disk

To install from tape, use the following jobs:

Step 1. Install the Feature

```
// JOB INSTALL FEATURE FROM TAPE
// ASSGN SYS006,cuu                Distribution tape
// ASSGN SYS008,cuu                Output unit for PRL
// ASSGN SYS009,cuu                Output unit for PSL
// ASSGN SYS001,cuu                From-unit for CORGZ PRL
// ASSGN SYS000,cuu                From-unit for CORGZ PSL
// ASSGN SYS002,cuu                Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEAT FROMTAPE MERGE
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 -
    ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of Step 1, the IBM-supplied feature libraries and the accompanying history file have been merged with your operational system. You may now verfy the installation by executing an MSHP RETRACE.

Note that additional installation activities may be required to make the installed feature operational.

DOWNLEVEL PROBLEM. If a downlevel problem has occured, MSHP does not merge the libraries and the history file to your operational system; instead, MSHP informs you (by means of a printout) which modules are affected and which PTFs (program temporary fixes) are required. In this case, provide the equivalent PTFs for the feature and apply these PTFs to the restored libraries as shown in Step 2; then continue with Step 3.

INSTALLATION ACTIVITIES (. . . . Cont'd)

Step 2. Update Feature Libraries

Apply the PTFs required to resolve the "downlevel" problem that became apparent in Step 1.

```
// JOB SELECT PTF's
// ASSGN SYS002,cuu                    Restored auxiliary history
// ASSGN SYS001,cuu                    MSHP workfile
// ASSGN SYS003,cuu                    MSHP workfile
// ASSGN SYS004,cuu                    Feature PTF tape
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSRL,'MSHP.PRIV.RLIB'
// EXTENT SYSRLB,,1,0,number1,number2  Restored RLB
// DLBL IJSYSSL,'MSHP.PRIV.SLIB'
// EXTENT SYSSLB,,1,0,number1,number2  Restored SLB
// OPTION CATAL
// EXEC MSHP
SELECT PTF DLIB
DEFINE HIST DLIB EXT=number1:number2 -
    ID='DOS.AUSILIARY.HISTORY.FILE'
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
/*
/&
```

On completion of this step. the feature libraries have been updated with the required PTFs.

Step 3. Install Updated Feature Libraries

This step installs the feature libraries you updated in Step 2.
Note that the library and history definitions in this step must be identical to Step 1.

```
// JOB INSALL FEATURE FROM DISK
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit for CORGZ PSL
// ASSGN SYS002,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEAT FROMDISK MERGE
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 -
    ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of this step, the feature has been successfully merged with the libraries of your operational system. You may now verify the installation by executing an MSHP RETRACE.

To install from disk, use the following jobs:

Step 1. Install the Feature

```
// JOB INSTALL FEATURE FROM DISK
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit for CORGZ PSL
// ASSGN SYS002,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEAT FROMDISK MERGE
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'  }  ID given at
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'  }  time of restore
DEF HIST AUX EXT=number1:number2 -
    ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of Step 1, the IBM-supplied feature libraries and the accompanying history file have been merged with your operational system. You may now verify the installation by executing an MSHP RETRACE.

Note that additional installation activities may be required to make the installed feature operational.

DOWNLEVEL PROBLEM: see Install a Feature to SYSRES from tape Step 1.

INSTALLATION ACTIVITIES (. . . . Cont'd)

Step 2. Update Feature Libraries

Apply the PTFs required to resolve the "downlevel" problem that became apparent in Step 1.

```
// JOB SELECT PTFs
// ASSGN SYS002,cuu                    Restored auxiliary history
// ASSGN SYS001,cuu                    MSHP workfile
// ASSGN SYS003,cuu                    MSHP workfile
// ASSGN SYS004,cuu                    Feature PTF tape
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSRL,'MSHP.PRIV.RLIB'
// EXTENT SYSRLB,,1,0,number1,number2 Restored RLB
// DLBL IJSYSSL,'MSHP.PRIV.SLIB'
// EXTENT SYSSLB,,1,0,number1,number2  Restored SLB
// OPTION CATAL
// EXEC MSHP
SELECT PTF DLIB
DEFINE HIST DLIB EXT=number1:number2 —
     ID='DOS.AUXILIARY.HISTORY.FILE'
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
/*
/&
```

On completion of this step, the feature libraries have been updated with the required PTFs.

Step 3. Install Updated Feature Libraries

This step installs the feature libraries you updated in Step 2.

```
// JOB INSTALL FEATURE FROM DISK
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit for CORGZ PSL
// ASSGN SYS002,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INST FEAT FROMDISK MERGE
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 —
     ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of this step, the feature has been successfully merged with the libraries of your operational system. You may now verify the installation by executing an MSHP RETRACE.

INSTALL A FEATURE TO PRIVATE LIBRARIES

The MSHP function INSTALL FEATURE is used to install IBM program packages designated as a feature.

The following procedures apply when installing a feature from tape or disk to private libraries. When installing a feature, either of the MSHP functions MERGE or ATTACH may be used; however, if the feature contains transient modules, MERGE should be used. The procedures assume that standard labels are used for the private libraries.

INSTALLATION ACTIVITIES (. . . . Cont'd)

Install from Tape or from Disk

To install from tape, execute the following jobs:

Step 1. Install the Feature

```
// JOB INSTALL FEATURE FROM TAPE
// ASSGN SYS006,cuu                    Distribution tape
// ASSGN SYS008,cuu
// ASSGN SYS009,cuu                    see Note 1
ASSGN SYSCLB,cuu
// ASSGN SYSRLB,cuu                    Target private libraries
// ASSGN SYSSLB,cuu
// ASSGN SYS000,cuu                    From-libraries for CORGZ
// ASSGN SYS001,cuu
// ASSGN SYS002,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INSTALL FEATURE FROMTAPE MERGE
DEF RLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 DIR=number3 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

Note 1:
The extents must be different to the extents of existing private libraries (the target private libraries).

On completion of Step 1, the IBM-supplied feature libraries and the accompanying history file have been merged with your operational system. You may now verify the installation by executing an MSHP RETRACE.

Note that additional installation activities may be required to make the installed feature operational.

DOWNLEVEL PROBLEM: see Install a Feature to SYSRES from tape Step 1.

Step 2. Update Feature Libraries

Apply the PTFs required to resolve the „downlevel" problem that became apparent in Step 1.

```
// JOB SELECT PTFs
// ASSGN SYS002,cuu                    Restored auxiliary history
// ASSGN SYS001,cuu                    MSHP workfile
// ASSGN SYS003,cuu                    MSHP workfile
// ASSGN SYS004,cuu                    Feature PTF tape
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSRL,'MSHP.PRIV.RLIB'
// EXTENT SYSRLB,,1,0,number1,number2 Restored RLB
// DLBL IJSYSSL,'MSHP.PRIV.SLIB'
// EXTENT SYSSLB,,,0,number1,number2  Restored SLB
// OPTION CATAL
// EXEC MSHP
SELECT PTF DLIB
DEFINE HIST DLIB EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE'
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
/*
/&
```

On completion of this step, the feature libraries have been updated with the required PTFs.

Step 3. Install Updated Feature Libraries

This step installs the feature libraries you updated in Step 2.

```
// JOB INSTALL FEATURE FROM DISK
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit for CORGZ PSL
// ASSGN SYS002,cuu                    Auxiliary history file
ASSGN SYSCLB                    }
// ASSGN SYSRLB                 }      Target private libraries
// ASSGN SYSSLB                 }
// OPTION CATAL
// EXEC MSHP
INST FEAT FROMDISK MERGE
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of this step, the feature has been successfully merged with the libraries of your operational system. You may now verify the installation by executing an MSHP RETRACE.

To install from disk, use the following jobs:

Step 1. Install the Feature

```
// JOB INSTALL FEATURE FROM DISK
ASSGN SYSCLB,cuu                }
// ASSGN SYSRLB,cuu             }      Target private libraries
// ASSGN SYSSLB,cuu             }
// ASSGN SYS000,cuu             }      From-libraries for CORGZ
// ASSGN SYS001,cuu             }
// ASSGN SYS002,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
INSTALL FEATURE FROMDISK MERGE
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of Step 1, the IBM-supplied feature libraries and the accompnying history file have been merged with your operational system. You may now verify the installation by executing an MSHP RETRACE.

Note that additional installation acitivites may be required to make the installed feature operational.

DOWNLEVEL PROBLEM: see Install a Feature to SYSRES from tape Step 1.

INSTALLATION ACTIVITIES (. . . Cont'd)

Step 2. Update Feature Libraries

Apply the PTFs required to resolve the "downlevel" problem that became apparent in Step 1.

```
// JOB SELECT PTFs
// ASSGN SYS002,cuu                    Restored auxiliary history
// ASSGN SYS001,cuu                    MSHP workfile
// ASSGN SYS003,cuu                    MSHP workfile
// ASSGN SYS004,cuu                    Feature PTF tape
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSRL,'MSHP.PRIV.RLIB'
// EXTENT SYSRLB,,1,0,number1,number2 Restored RLB
// DLBL IJSYSSL,'MSHP.PRIV.SLIB'
// EXTENT SYSSLB,,1,0,number1,number2  Restored SLB
// OPTION CATAL
// EXEC MSHP
SELECT PTF DLIB
DEFINE HIST DLIB EXT=number1:number2 UNIT=SYS002 –
     ID='DOS.AUXILIARY.HISTORY.FILE'
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
PTF UDnnnnn
/*
/&
```

On completion of this step, the feature libraries have been updated with the required PTFs.

Step 3. Install Updated Feature Libraries

This step installs the feature libraries you updated in Step 2.

```
// JOB INSTALL FEATURE FROM DISK
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit fo  r CORGZ PSL
// ASSGN SYS002,cuu                    Auxiliary history file
ASSGN SYSCLB,cuu
// ASSGN SYSRLB,cuu                    Target private libraries
// ASSGN SYSSLB,cuu
// OPTION CATAL
// EXEC MSHP
INST FEAT FROMDISK MERGE
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 –
     ID='DOS.AUXILIARY.HISTORY.FILE'
/*
/&
```

On completion of this step, the feature has been successfully merged with the libraries of your operational system. You may now verify the installation by executing an MSHP RETRACE.

INSTALLATION ACTIVITIES (. . . . Cont'd)

INSTALL A RELEASE 34 COMPONENT

If a DOS/VS Release 34 product component (for example DOS/VS COBOL) is to be installed in a DOS/VSE, but without installation support by MSHP, proceed as follows.

Step 1. Install

Install the component as described in the applicable documentation (for example, an Installation Guide) available with the product.

Step 2. Record the Installation

For an up-to-date history file, the installation needs to be recorded in your system history. Submit a job as follows:

```
// JOB NOTIFY HISTORY
// OPTION CATAL
// EXEC MSHP
   ARCHIVE xxxx-yy-zzz REL=RR.R
/*
/&
```

where:

xxxx-yy-zzz is the component name. For non-SCP related products, the name is normally derived from the program number; for example, if the program number is 5748-UT2, the component name would be 5748-UT-200.

RR.R is the release level of the product for example RR.R=02.0 for Release 2 of a product.

SERVICE ACTIVITIES

S1. APPLY A SINGLE PTF

A single PTF consists or a single DOS/VSE job supplied by IBM in SYSIN format on magnetic tape or diskette or as a deck of cards.

To apply a single PTF, place the applicable control statements ahead of the job stream supplied with the PTF and execute the job. On completion of the job, the PTF is applied to your operational system, and this application is reflected in your system history file. The control statements are:

```
// DLBL IJSYSIN,'file-ID',,codes
// EXTENT SYSIN,serial-number
ASSGN SYSIN,cuu
```

Note:
DLBL and EXTENT statements are only required if the input is from disk or diskette. If the input is from diskette and you use the IBM-supplied standard labels and the diskette file-id is DTTEPTF, the DLBL and EXTENT statements are not required.

The Following are Job Examples of Single PTF Applications:

```
// JOB UD 13854
// PAUSE EOB OR CANCEL
// OPTION CATAL
// EXEC MSHP
APPLY 5745 - SC - VSM: UD13854 REL=35
RESOLVES APARS = DY13415
AFFECTS MODULES = IKQLAB
OCCUPIES CLIB = 311 RLIB = 10
INVOLVES LINK = IKQVSMLK
DATA
TXT
:
:
END
/$
/*
/&
```

Single PTF Application of a Backout PTF generated with the REVokable Option:

```
// JOB UD13854 MSHP REVOKE PTF
// PAUSE ASSGN LIBRARIES IF NEEDED
// OPTION CATAL
// EXEC MSHP
REVOKE 5745-SC-VSM: UD13854
DATA
TXT
:
:
END
/$
/*
/&
```

SERVICE ACTIVITIES (. . . Cont'd)

S2. APPLY COMULATIVE PTFs

A cumulative PTF file consists of two or more single PTFs in SYSIN format on tape, disk, or diskette. The shipment also contains the cover-letter file with information such as the phases a PTF affects, detail description of the problem resolved, and required library sizes.

In this procedure, information about PTFs is retrieved, selected PTFs are applied to the operational system, and the PTF application is recorded in the current history file.

To apply from tape, use the following jobs:

Step 1. List Cover Letter

Cover letter information is given in specific files of the distributed material. Refer to the transmittal letter to find out which file applies and specify as shown below.

```
// JOB PRINT COVERLETTER
// ASSGN SYS004,cuu
// MTC FSF,SYS004,n                Note 1
// EXEC MSHP
LIST PTFFILE STATEM SEP             Note 2
PTF UD 12740
PTF UD 13805
—                                   Note 3
—
—
PTF UDnnnn
/*
/&
```

Note 1:
This statement positions the tape to the cover-letter file, where n is the sequence number of the cover-letter file minus one.

Note 2:
This statement prints, on SYSLST, the contents or the cover-letter file; SEP, if specified, forces a new print page to be started for each PTF information listing.

Note 3:
If you want only information for one or more specific PTFs, specify the pertinent PTF numbers. Omit the statements if you want a printout on all of the PTFs.

Step 2. Apply PTFs

```
// JOB SELECT PTFs FROM TAPE
// ASSGN SYSPCH,TAPE
// ASSGN SYS003,cuu                 MSHP workfile
// ASSGN SYS001,cuu                 MSHP workfile
// ASSGN SYS004,cuu                 Cumulative PTF tape
// MTC FSF,SYS004,n                 Not 1
// OPTION CATAL
// EXEC MSHP
SELECT PTF APPLY REVOKABLE          Note 2
PTF UD12740
PTF UD13854                         Note 3
PTF UDnnnnn
/*
/&
```

Note 1:
This statement positions the tape at the pertinent file.

Note 2:
The default is IRREVOKABLE. Specify REVOKABLE (as shown) if you wish that a "backout" is produced on SYSPCH for the PTFs you selected for application (SYSPCH should be assigned to a magnetic tape drive to avoid card punching). The output of a revocation can be applied as a normal PTF.

Note 3:
If one or more of the PTFs require corequisite PTFs, refer to the procedure "Apply Corequisite PTFs".

To apply from disk, change the above jobs as follows:

- Assign SYS004 to disk (instead of to tape).
- Supply, in both Step 1 and Step 2, DLBL and EXTENT statements for SYS004 on disk.

SERVICE ACTIVITIES (. . . . Cont'd)

S3. APPLY COREQUISITE PTFs

The following procedures show how to apply corequisite PTFs from card and from a cumulative PTF tape.

Note: For the creation of private libraries, please refer to MSHP USER's GUIDE or DOS/VSE System Management Guide.

Step 2 writes the two corequisite PTFs into the private libraries, and Step 3 applies the corequisite PTFs to your system.

Step 1. Copy History File to Auxiliary History

```
// JOB COPY HIST TO AUX
// EXEC MSHP
// ASSGN SYS018,cuu
COPY HIST SYS AUX
DEF HIST AUX EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE'
    UNIT=SYS018
```

Step 2. First and Second Corequisite PTF Supplied on Cards

```
// JOB SELECT COREQUISITE PTF
// ASSGN SYS001,cuu                    MSHP Workfile
// ASSGN SYS003,cuu                    MSHP Workfile
// ASSGN SYS004,cuu                    PTFs from card reader
// ASSGN SYS018,cuu                    AUX HIST FILE defined in Step 1
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSCL,'MSHP.PRIV.CLIB'
// EXTENT SYSCLB,,,number1,number2
// DLBL IJSYSRL,'MSHP.PRIV.RLIB'
// EXTENT SYSRLB,,,number1,number2
// DLBL IJSYSSL,'MSHP.PRIV.SLIB'
// EXTENT SYSSLB,,,number1,number2
ASSGN SYSCLB,cuu
// OPTION CATAL
// EXEC MSHP
SELECT PTF DLIB
PTF UDnnnnn                            } Specify the PTF numbers
PTF UDnnnnn                            } to be selected
DEF HIST SYS EXT=number1:number2 –                                  } Note
    ID='DOS.AUXILIARY.HISTORY.FILE'  Auxiliary history file         }
    UNIT=SYS018
/*
   } Insert card decks of selected PTF's
/*
/& } Terminate the job by these additional cards
```

Note: Same EXT=, EXT=, ID=, and UNIT= as used in Step 1

Step 3. Apply the PTFs

This step applies the corequisite PTFs to the system by using the UPGRADE FROMDISK function.

```
// JOB UPGRADE FROM DISK
// ASSGN SYS003,cuu     (1)
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit for CORGZ PSL
// ASSGN SYS018,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
UPGR FROMDISK FORCELINK
DEF CLIB PRIV EXT=0:0  (1)
DEF CLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.CLIB'
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB' } Note 2
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB' }
DEF HIST AUX EXT=number1:number2 –                                  Note 3
    ID='DOS.AUXILIARY.HISTORY.FILE' –
    UNIT=SYS018
/*
/&
```

Note 1: The statments marked (1) are only necessary for compatibility reasons.

SERVICE ACTIVITIES (. . . . Cont'd)

Note 2:
The EXT= and the ID= of the define statements for RLIB and SLIB must be identical to the DLBL and extent information of Step 2. The EXT= and the ID= of the DEF HIST AUX must be identical to that of Step 1.

Note 3:
Same EXT= and ID= as used in Step 1 and 2.

To Apply from Cumulative PTF Tape

Step 2 and Step 3 write the first and the second corequisite PTF from the cumulative PTF tape to private libraries, and Step 4 applies the corequisite PTFs to your system.

Note: For the creation of private libraries, please refer to MSHP USER's GUIDE or DOS/VSE System Management Guide.

Step 1. Copy History File to Auxiliary History

```
// JOB COPY HIST TO AUX
// ASSGN SYS018,cuu
// EXEC MSHP
COPY HIST SYS AUX
DEF HIST AUX EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE' –
    UNIT=SYS018
/*
/&
```

Step 2. First Corequisite PTF

```
// JOB SELECT COREQUISITE PTF
// ASSGN SYS001,cuu                    MSHP workfile
// ASSGN SYS003,cuu                    MSHP workfile
// ASSGN SYS004,cuu                    PTF tape
// ASSGN SYS018,cuu                    AUX HIST FILE defined in Step 1
// ASSGN SYSRLB,cuu
// ASSGN SYSSLB,cuu
// DLBL IJSYSCL,'MSHP.PRIV.CLIB'
// EXTENT SYSCLB,,,number1,number2
// DLBL IJSYSRL,'MSHP.PRIV.RLIB'
// EXTENT SYSRLB,,1,0,number1,number2
// DLBL IJSYSSL,'MSHP.PRIV.SLIB'
// EXTENT SYSSLB,,1,0,number1,number2
ASSGN SYSCLB,cuu
// OPTION CATAL
// EXEC MSHP
SELECT PTF DLIB
PTF UDnnnnn
DEF HIST SYS EXT=number1:number2 –   } Note
    ID='DOS.AUXILIARY.HISTORY.FILE' –
    UNIT=SYS018
/*
/&
```

Note:
Same EXT=, ID=, and UNIT= as used in Step 1.

Step 3. Second Corequisite PTF

Apply the second corequisite PTF in the same way as shown in Step 2. Note, the EXT= and the ID= of the DEF HIST SYS statement must be identical to that in Step 1.

Step 4. Apply the PTFs

This step applies the corequisite PTFs to the system by using the UPGRADE FROMDISK function.

```
// JOB UPGRADE FROM DISK
// ASSGN SYS003,cuu     (1)
// ASSGN SYS001,cuu                    From-unit for CORGZ PRL
// ASSGN SYS000,cuu                    From-unit for CORGZ PSL
// ASSGN SYS018,cuu                    Auxiliary history file
// OPTION CATAL
// EXEC MSHP
```

SERVICE ACTIVITIES (. . . . Cont'd)

```
UPGR FROMDISK FORCELINK
DEF CLIB PRIV EXT=0:0          (1)
DEF CLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.CLIB'
DEF RLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.RLIB'  } Note 2
DEF SLIB PRIV EXT=number1:number2 ID='MSHP.PRIV.SLIB'
DEF HIST AUX EXT=number1:number2 –
    ID='DOS.AUXILIARY.HISTORY.FILE' –                  Note 3
    UNIT=SYS018
/*
/&
```

Note 1:
The statements marked (1) are only necessary for compatibility reasons.

Note 2:
The EXT= and the ID= of the define statements for RLIB and SLIB must be identical to the DLBL and extent information of Step 2. The EXT= and the ID= of the DEF HIST SYS must be identical to that of Step 1.

Note 3:
Same EXT= and ID= as used in Step 1 and 2.

S4. RECORD A LOCAL FIX

A "local fix" is a correction to a phase, module, and/or source statement book applied at a specific installation to resolve a problem. A local fix might eventually be covered by IBM-supplied PTFs, upgrades, or DLIBs.

To keep an operational system's history file up-to-date, it is essential that you record the application of a local fix in the history file of your DOS/VSE. This can be done by using the MSHP control statement ARChive as shown below.

```
// JOB RECORD FIX                    (Example Jobstream)
// OPTION CATAL
// EXEC MSHP
ARC 5745-SC-VSM APAR=DY11592
RESOLVES APARS=DY11500
AFFECTS MODULE=IKQRCL                Affected Module
ALTER 000055 : EEEEFFFF              ADDR: NEW TEXT (Applied Change)
ARC 5745-SC-VSM APAR=DY12005
RESOLVES APARS=DY12000
AFFECTS PHASE=IPWSSPL                Affected Phase
ALTER 00035E 4700C005: 47FOCB25      ADDR OLDTEXT: NEW TEXT
ARC 5745-SC-DKE APAR=DY13713
RESOLVES APARS=DY13700
AFFECTS MAC=SGDSK SUBLIB=A           Affected Macro
DELETE 000250 : 000252               Deleted Macrolines FROMLINE: TOLINE
REPLACE 000253:000254                FROMLINE:TOLINE to be replaced
XXXX                                 } New Input Lines
YYYY                                 } For Replacement
/S
INSERT 000249                        Insert Afterline
XXXX                                 } New Lines
YYYY                                 } tobe inserted
/S
/*
/&
```

Notes:
Control statements ALTER, REPLACE, DELETE, and INSERT are optional. If the statements are specified, they are recorded in the history file. The statements should be used to document, in detail, the fix applied.

ALTER is used to record phase and module changes. One separate set of AFFECTS and ALTER statements has to be specified for each single phase/module.

The statements REPLACE, DELETE, and INSERT are used to record macro changes. One separate set of AFFECTS, DELETE, etc. statements must be specified for each single macro.

The RESOLVES statement may contain no other specification but a comment; this comment can be a string of up to 35 characters. Only one RESOLVES statement may be specified for one ARChive statement.

SERVICE ACTIVITIES (. . . . Cont'd)

S5. APPLY PTFs OF THE RELEASE 34 MSHP FORMAT

PTFs relating to components of the pre-DOS/VSE level are distributed in the "Release 34 MSHP format"; the following is an example of this format:

```
// JOB N12345
* COMP 5745-SC-AIT
* NPRE N67890
* COREQ N56789
* PRE N23456
* SUP N34567
* APPL REL 34.0,35.0
* APARS FIXED E45678
* MOD/MAC AFFECTED $$BATTNA
* PHASES $$BATTNA
* ENVIRONMENT DOS/VS SCP
* BLOCKS CL=2, RL=3, SL=4
* COMMENT THIS IS AN EXAMPLE OF A PTF
// PAUSE EOB OR CANCEL
// EXEC MAINT
 CATALR . . .
 . . . .
 CATALS . . .
 . . . .
/*
// OPTION CATAL
 INCLUDE
 . . . .
// EXEC LNKEDT
/&
```

"Release 34 MSHP format" PTFs, although their format differs from the "DOS/VSE MSHP format", can be applied (under the control of MSHP) to the DOS/VSE. Release 34 MSHP format PTFs are numbered as Nnnnnn whereas DOS/VSE MSHP format PTFs are numbered as UDnnnnn.

Note:
If the PTF to be applied to a specific component includes a prerequisite or corequisite PTF that has already been applied to the system but not to the same specific component, then the application of the PTF fails due to requirements not met; message M151I is issued. In such case, eliminate the requirement statement (for example, the statement PRE N23456 in the above job) and rerun the application job. Message M151I is also issued if the pre- or corequisite PTF(s) have not been installed; in this case, correct the situation before applying the PTF.

To Apply a Single PTF from Card, Magnetic Tape, Disk or Diskette

Single PTFs may be supplied on card, magnetic tape, disk, or diskette. The job streams to apply a single PTF are as follows.

Apply from Card

```
// JOB APPLY R34 PTF
// ASSGN SYS001,cuu                    MSHP workfile
// ASSGN SYS003,cuu                    MSHP workfile
// ASSGN SYSPCH,cuu                    Output for REVOKABLE
// ASSGN SYS004,cuu                    Input from card reader
// OPTION CATAL
// EXEC MSHP
ACCEPT OLDPTF APPLY REVOKABLE
/*                                     Terminating MSHP control statements
// JOB Nnnnnn
   :                                   Release 34 MSHP format
   :                                   single PTF job cards
/*
/&
/*                                     Terminating input for SYS004
/&                                     Terminating MSHP job
```

Apply from Magnetic Tape

```
// JOB SELECT R34 PTF's
// ASSGN SYS002,cuu          MSHP workfile
// ASSGN SYS003,cuu          MSHP workfile
// ASSGN SYSPCH,cuu
// ASSGN SYS004,cuu          Cumulative PTF tape
// MTC FSF,SYS004,n          If required
// OPTION CATAL
// EXEC MSHP
SELECT PTF APPLY REVOKABLE
PTF N12345
/*
/&
```

Apply from Disk or Diskette

The job stream is the same as that for "Apply from Magnetic Tape" above, except that:

- SYS004 is assigned to the disk or diskette I/O unit.
- DLBL and EXTENT statements must be given.

To Apply Cumulative PTFs from Magnetic Tape or Disk

Cumulative PTFs may be supplied on magnetic tape or on disk. The job streams to select PTFs and to apply the selected PTFs are as follows.

Apply from Magnetic Tape

```
// JOB SELECT
// ASSGN SYS001,cuu          MSHP work file
// ASSGN SYS003,cuu          MSHP work file
// ASSGN SYS004,cuu          Cumulative PTF tape
// MTC FSF,SYS004,n          If required
// OPTION CATAL
// EXEC MSHP
SELECT PTF APPLY
PTF N78910
:
:
/*
/&
```

Apply from Disk

```
// JOB SELECT
// ASSGN SYS001,cuu          MSHP work file
// ASSGN SYS003,cuu          MSHP work file
// ASSGN SYS004,cuu          PTF disk file
// DLBL . . .                } Description of the
// EXTENT . . .              } PTF file on disk
// OPTION CATAL
// EXEC MSHP
SELECT PTF APPLY
PTF N78910
:
:
/*
/&
```

NOTATIONAL CONVENTIONS

The syntax of both the function and detail control statements is represented as path diagrams as explained below:

- The structure of a given statement is derived from its diagram by following the lines from left to right, and from top to bottom. The entries given in the diagram (for example, RELease = release-number-list) are selected according to the MSHP services required, and are then coded as described in the explanatory text for the given entry.

  At points of branching path lines, a choice must (or may) be made to select one, two, or more keywords.

  In a set of alternatives, one alternative may be shown inside a line of bullets, for example: ··· IRRevokable ···

  This indicates the default used by MSHP if none of the alternatives in the set is coded.

- In the diagram (and text), words given in all lower-case letters are symbolic; they must be replaced by user-supplied values as detailed in the discussions for the pertinent keywords.

- Some of the all lower-case letter words (see above) end with the suffix '-list'. For example: release-number-list

  'list' means, that the symbolic word (release-number) may be coded up to 100 times. Thus: (release-numer-1, release-number-2,. . .)

  The code for a series of symbolic words must be enclosed in parentheses (see above). The multiple code units must be separated in one of the following ways:

  – By one or more blanks, or
  – By a comment, or
  – By a comma (which in turn may be surrounded by one or more blanks, or comments)

  Commas and blanks as separators may be intermixed in a given 'list'.

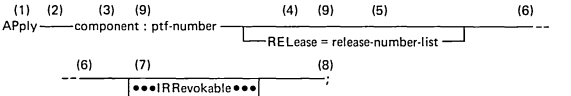
- In the diagrams, the path lines can be considered as instruction to code:

  – One or more blanks, or
  – One or more comments, or
  – One comma

- The equal sign ('=') and colon (':') in the diagrams must be coded as shown. They may be surrounded by one or more blanks (or comment).

- The diagrams show the terminating semicolon (';') for each statement. However, it is needed only, if a statement is to be followed by another statement on the same line. (For details, and for the continuation capability see the Rules for Coding MSHP Statements).

As an example, the following shows the path diagram of the Apply statement.



Comments:

(1) 'APply is the command verb of the statement.

(2) This is a part of the statement's total path line; this part indicates that you can code, between 'APply' and 'component', one or more blanks, one or more comments, or one comma.

(3) 'component' and 'ptf-number' are symbolic words that must be coded according to the details given in the discussion of the statement.

(4) 'RELease' is a keyword.

(5) 'release-number-list' is a symbolic word; when coded, it gives the specification relating to the keyword 'RELease'. Code the symbolic word according to the discussion of the statement.

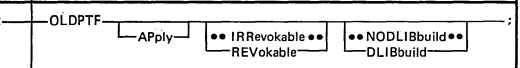
'list' in the symbolic word means that for 'release-number' up to 100 release identifications may be specified.

(6) The points indicate continuation of the path line; they must not be coded.

(7) The bullets are part of the total path line and indicate the default. In this example, the default is 'IRRevokable'.

(8) This is the terminator of the statement; it must be coded if the statement is to be followed by another statement on the same input line.

(9) The colon and equal signs must always be coded.

Function Control Statements	Keyword Operands
ACCept	

APPLICATION/EXPLANATION

The ACCEPT statement requests MSHP to read single PTFs in the Release 34 card-format from SYS004 and to apply them.

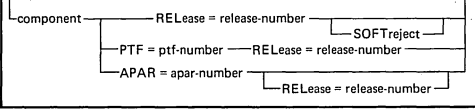
The Accept function needs two disk workfiles:
IJSYS01 on SYS001, and IJSYS03 on SYS003.

If the PTF is to be applied to private libraries, assignments must be made in the job invoking MSHP. Any assignments or PAUSE statements in the PTF job itself are ineffective. The same holds for // OPTION CATAL. It must be specified in the MSHP job.

Related Detail Control Statements:

Mandatory: none Optional: DEFine

OLDPTF	Indicates that a PTF of the 'old' format (the Release 34 format) is to be accepted.
APply	Specifies that the PTF is to be applied (see Note).
IRRevokable	When applying the PTF, no backout PTF will be generated, and the PTF cannot be revoked, that is, the status before the application of the PTF cannot be recreated at a later point in time.
REVokable	When applying the PTF, a backout PTF will be generated on SYSPCH. Restriction: When REVokable is specified, the PTF must not have any corequisites (must not specify CO= in the REQuires statement).
NODLIBbuild	Suppresses the DLIBbuild option (see below).
DLIBbuild	This keyword is provided primarily for IBM internal use. Specify this option if the PTF is to be applied to libraries intended for preventive maintenance. Note: Default: The keyword APply may be omitted when option DLIBbuild is specified. The omission of APply in this case indicates that the modules and/or macros of the PTF are to be catalogued in the respective libraries, but that link-edits according to "INVOLVES LINK = link-book-list" statement(s) are not to be performed (see INVolves detail control statement).

Function Control Statement	Keyword Operands
ARChive	

APPLICATION/EXPLANATION

The ARCHIVE statement is provided primarily for IBM internal use and is used to make entries in the system history file that accompanies IBM-provided programming support. The entries that can be made are for:

- features
- components
- PTFs
- Local/APAR fixes

The statement may be used to record local fixes in an installed system's history file.

Related Detail Control Statements:

Mandatory:	AFFects OCCupies RESolves	for ARChive component PTF	Optional: AFFects DEFine OCCupies OR REQuires INVolves SUPersedes ALTER DELete INsert REPlace RESolves
Mandatory:	AFFects	for ARChive component APAR	

feature — Specifies that an entry for a feature is to be made in the system history file, and details which one.

A feature id can also be used to specify the precise release level of the program package that is to be archived.

The feature id consists of three alphameric characters.

component — Specifies that, providing the statement does not include PTF= or APAR= specifications, an entry in the system history file is to be made for a component, and the component's identification. Otherwise, "component" identifies the component to which the PTF or local fix (to be ARChived) is applicable.

Component is a string of 11 characters, according to the following example: 5745-SC-JCL

RELease — Identifies the level of the release, maintenance, or version of a component, PTF, or local fix/APAR.

APPLICATION/EXPLANATION (. . .Cont'd)

release-number	Specifies the release number to which the component, PTF, or local fix/APAR is applicable.
SOFTreject	Specifies that a PTF Application/Revokation for that component can be forced by the user, even if it partially overwrites a local fix or APAR correction. For components, which are archived without this option, all PTFs overwriting partially local/APAR fixes will be rejected automatically.
PTF	Identifies the PTFs for which an entry is to be made in the history file.
ptf-number	Specifies the number of the PTF(s), for which a history file entry is to be made. ptf-number is a string of 6 or 7 characters, the first one or two being alphabetic the remaining five being digits. For example: N12345 UD12345
APAR	Identifies that an entry in the system history file is to be made for a local fix.
apar-number	Specifies under which number the local fix is to be recorded in the system history file. apar-number is a string of 6 or 7 characters: one or two alphabetics, followed by five digits.

Function Control Statement	Keyword Operands
BACKup	History ─┬─ •••••SYStem••••• ─┬─ ; └─ AUXiliary ─┘

APPLICATION/EXPLANATION

The BACKUP statement requests MSHP to copy a history file located on disk onto magnetic tape.
SYSPCH or tape must be assigned to SYS006.

Related Detail Control Statements:

Mandatory: none   Optional: DEFine

History	Identifies this backup as a 'history file' backup.
SYStem	Specifies that the history file, that will be copied to tape, is the one with the file name IJSYSHF (which normally is the system history file).
AUXiliary	Specifies that the file with the file name IJSYS02 is the history file, and that the file is to be copied to tape. If the file IJSYS02 is not a history file, the job will be canceled (most likely with an "invalid length" IOCS error).

Function Control Statements	Keyword OPerands
COPy	History — SYStem — AUXiliary ; (or AUXiliary — SYStem)

APPLICATION/EXPLANATION

The COPY statement requests MSHP to copy a history file from disk to disk; either to create a copy of the system history file for use as an auxiliary history file, or to copy an auxiliary file to the system history file.

The sequence of the keywords SYStem and AUXiliary defines the direction of the copy operation. The first keyword specifies the source, and the second the target. The two keywords must be specified adjacent to each other.

Related Detail Control Statements:

Mandatory: none Optional: DEFine

History	Identifies the copy operation as an MSHP operation.
SYStem	Designates the history file with the file-name IJSYSHF.
AUXiliary	Designates a history file with the file-name IJSYS02.

Function Control Statements	Keyword Operands
CReate	History — SYStem / AUXiliary ;

APPLICATION/EXPLANATION

This statement is provided primarily for IBM internal use. The CREATE statement requests MSHP to preformat a history file. MSHP creates the general header record. Further, space is provided for the personalization information to be inserted with the PERsonalize statement.

Related Detail Control Statement:

Mandatory: none Optional: DEFine

History	Identifies the create operation as an MSHP operation.
SYStem	Specifies that the history file to be created be the one with the file-name: IJSYSHF
AUXiliary	Specifies that the history file to be created is to be the one with the file-name: IJSYS02

Function Control Statements	Keyword Operands
DUMP	History — SYStem / AUXiliary ;

The DUMP statement requests MSHP to produce a formatted hexadecimal printout of a history file on SYSLST. The records of the history file are identified by their RBA's (Relative Byte Addresses) for ease of pointer identification.

Related Detail Control Statements: see CReate.

History	Identifies the dump operation as an MSHP operation.
SYStem	Specifies that the history file to be dumped is the one with the file-name: IJSYSHF
AUXiliary	Specifies that the history file to be dumped be the one with the file-name: IJSYS02

Function Control Statements	Keyword Operands
INCorporate	component RELease = release-number ;

APPLICATION/EXPLANATION

The INCORPORATE statement identifies to MSHP and initiates a component distributed in SYSIN format. The job invoking MSHP with the function control statement INCorporate, must have // OPTION CATAL in effect.

RElated Detail Control Statements:

Mandatory: DATA
OCCupies CLib

Optional: DEFine
INVolves
OR
REQuires

component — Identifies the component to be incorporated. Component is a string of 11 characters according to the example: 5745-SC-JCL.

RELease — Identifies the level of release, maintenance, or version of a component.

release number — Specifies the level of release, maintenance, or version of the component to be incorporated.

Function Control Statements	Keyword Operands
INSTall	SYStem / COMPonents / FEAtures / PRODuct [FROMTape / FROMDisk] [MERge / ATTach] ;

APPLICATION/EXPLANATION

The INSTall statement requests MSHP to install a new "system", one or more "components", one or more "features", or "products".
Negative "requirements" ("NOT=") are always checked against the system history file.

Checks for pre-requisite and co-requisite PTFs are performed against the operational system's history file and the distribution history file as well.

Checks for "PRE=" and "CO=" requirements for components and/or features depend on the INSTall function being executed. The checks go against the system history or against the distribution history file, or against both.

Related Detail Control Statements:

Mandatory: DEFine libraries (see Note 1) Optional: DEFine History

Note 1:
With INSTall SYStem, the libraries defined must be of the type SYStem. For all other INSTall functions, they must be of the type PRIVate.

SYStem — Specifies that a system is to be installed.

Restrictions: For restoring the distribution tape (see option FROMTape, the libraries in the DEFine statement must be of the type SYStem.

With INSTall SYStem, MERge must not be specified, instead, the option ATTach must be used.

Default: With INSTall SYStem, ATTach is the default.

COMPonents — Specifies that the somponent(s) from the distribution libraries are to be installed.

Restrictions: Same as described under 'PRODuct'.

FEAtures — Specifies that one or more features from the distribution libraries are to be installed.
Restrictions: Same as described under 'PRODuct'.

APPLICATION/EXPLANATION (. . . . Cont'd)

PRODuct — Specifies that the component(s) and feature(s) composing a complete executable version of a program package are to be installed from the distribution libraries.

Restrictions: For restoring the distribution tape (see option FROMTape below), the libraries defined in the DEFine statement must be of the type PRIVate.

FROMTape — Specifies that the installation source is the distribution tape assigned as SYS006. The library backups on the tape are restored by the INSTall function as private or system libraries, as specified in the DEFine statement.
If the libraries are to be restored as system libraries, SYS005 must be assigned to the disk onto which the libraries are to go. If they are to be restored as private libraries, the following assignments must be effective:
SYS007 for PRIVate CLib
SYS008 for PRIVate RLib
SYS009 for PRIVate SLib

FROMDisk — FROMDisk must be specified to support the INSTall function for a system configuration without magnetic tape facility.

MERge — Specifies that the restored modules, macros, and/or phases are to be copied to the system libraries, or to the assigned private libraries (if any).

For the merge, the following assignments must be effective for the restored libraries:
SYS000 for SLib
SYS001 for RLib
SYS003 for CLib

Restriction: This option must not be specified with INSTall SYStem.

ATTach — Specifies that the MSHP INSTall function ends with a message informing the user that the libraries can now be used in his system. Its use may simply be to assign the libraries as private libraries.

Function Control STatement	Keyword Operands
LIST	PTFfile — INDex — ; STatements — ●●●● CONTinuous ●●●● SEParate

APPLICATION/EXPLANATION

The LIST statement requests MSHP to retrieve information about the cumulative PTF file from that file, and to print the information in SYSLST.
The device to read the cumulative PTF file must be assigned to SYS004.

Related Detail Control Statements:

Mandatory: none

Optional: DEFine History
PTF (if statement is specified)

PTFfile — Identifies the listing operation as an MSHP operation.

INDex — Identifies that the numbers of the PTFs contained in the cumulative PTF file are listed.

The records of the cumulative PTF file are sequentially checked to contain a // JOB statement. If one is found, the job name (according to JCL syntax) is printed as PTF number. Together with the PTF number, the number of records from the // JOB statement up to and including the next record containing /& in columns 1 and 2, is listed.

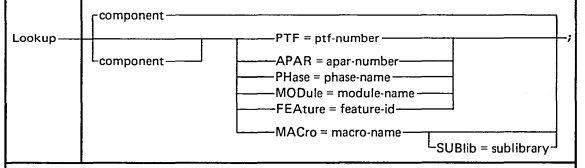
STatements — Specifies that the following is printed on SYSLST:

- Job Control statements (including JCL comments)
- MAINT control statements
- Linkage Editor control statements
- MSHP control statements on the cumulative PTF file

CONTinuous — Specifies that the statements of the PTFs are printed without starting a new page for each PTF.

SEParate — Specifies that the statements of the PTFs are printed with the first statement of each PTF beginning on a new page.

Function Control Statement	Keyword Operands



APPLICATION/EXPLANATION

The LOOKUP statement requests MSHP to display selected information from the system history file on SYSLOG.

If MSHP is invoked from SYSLOG, then the LOOKUP statement may be entered from SYSLOG.

Related Detail Control Statements:

Mandatory: none

Optional: DEFine History

component — Specifies the component for which information is to be displayed.

If component is specified without any further keyword operands, the following history information is displayed:

- Component identifier
- Release
- Latest Service: PTF number of the most recently applied PTF and its application date, or NO PTF applied.
- Invalidated APARS: APAR(S) which are partially overwritten by PTF forcing. (Valid for Components archived with the SOFTreject Option.)

component is a string of 11 characters, according to the example: 5745-SC-JCL.

PTF — Indicates that for the given ptf-number, the history information is to be displayed.

ptf-number — Specifies the PTF number for which information is to be displayed.

ptf-number is a string of 6 or 7 characters, the first one or two being alphabetic the remaining five being digits.
For example:
N12345
UD12345

APAR — Indicates that for the given a par-number, the history information is to be displayed.

apar-number — Specifies the APAR number for which information is to be displayed.

apar-number is a string of 6 or 7 characters: one or two alphabetics, followed by five digits.

PHase — Indicates that for the given phase-number the information is to be displayed.

phase-name — Specifies the phase name for which information is to be displayed.

phase-name is a string of one to eight alphameric characters:
0-9, A-Z, #, $, and @

MODule — Indicates that for the given module-name information is to be displayed from the system history file.

module-name — Specifies the module-name for which information is to be displayed.

module-name is a string of one to eight alphameric characters:
0-9, A-Z, #, $, and @

FEAture — Indicates that for the given feature-id the following information is to be displayed:

- feature-id
- date of installation

feature-id — Specifies the feature for which information is to be displayed.

feature-id is a string of three alphameric characters.

APPLICATION/EXPLANATION (. . . Cont'd)

MACro	Indicates that for the given macro-name the information is to be displayed from the history file:
macro-name	Specifies the macro-name for which information is to be displayed. macro-name is a string of one to eight alphameric characters: 0-9, A-Z, #, $, and @ the first character must be alphabetics: A-Z, #, $, or @
SUBlib	Identifies that the given sublibrary (see below) contains the source statement book. Default: If omitted, it is assumed that information is to be displayed for an E-type macro.
sublibrary	Specifies the sublibrary of the source statement library in which the macro is cataloged. sublibrary is one alphabetic character (A-Z).

Function Control Statement	Keyword Operands
PERsonalize	'customer-name' ADDRess = 'customer-address' PHone = 'phone-number' PRogrammer = 'programmer-name' ENVironment = 'description' ;

APPLICATION/EXPLANATION

The PERsonalize statement is used to identify the history file and relate it to a specific user. Changes can be made by re-coding the statement.
Restrictions: Specification of PERsonalize without at least one operand is invalid. If the history file has not been specified before, specification of "customer-name" and "customer-address" is mandatory.

Related Detail Control Statements:

Mandatory: none Optional: DEFine History

'customer-name'	Specifies the user's name. customer-name is a string (enclosed in quotes) of one to twenty characters (not counting the quotes). (See Note)
ADDRess	Specifies that the given customer-address (see below) is to be entered.
'customer-address'	Specifies the customer's address. customer-address is a string (enclosed in quotes) of one to 45 characters (not counting the quotes). (Restriction: see Note below)
PHone	Specifies that the given phone-number (see below) is to be entered.
'phone-number'	Specifies the customer's telephone number. If a null string is specified, a previously specified number is erased. phone-number is a string (enclosed in quotes) of one to 17 characters (not counting the quotes).
Note:	Restriction: If the history file has not been personalized before, specification of customer-address is mandatory.

APPLICATION/EXPLANATION (. . . Cont'd)

PRogrammer — Specifies that the given programmer-name (see below) is to be entered.

'programmer-name'

Specifies the programmer's name. If a null string is specified, a previously specified name is erased.

programmer name is a string (enclosed in quotes) of one to 24 characters (not counting the quotes).
A null string (two consecutive quotes) is accepted.

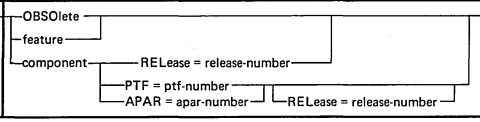
ENVironment — Specifies that the given description (see below) is to be entered.

'description'

Specifies the environment description (for example, the release level). If a null string is specified, a previously specified name is erased.

description is a string enclosed in quotes of 1 to 62 characters (not counting the quotes).
A null string (two consecutive quotes) is accepted.

Function Control Statement	Keyword Operands



APPLICATION/EXPLANATION

The REmove statement requests MSHP to erase entries from the system history file. The space in the history file of the removed entries is freed for future usage.

Related Detail Control Statement:

Mandatory: none — Optional: DEFine History

OBSOlete — Specifies that entries previously marked as obsolete are to be physically removed from the history file.

feature — Specifies that a specific entry relating to a feature is to be removed.

feature is a three alphameric character string identifying the feature entry.

component — If followed by: RELease = release-number

component specifies which component entry is to be removed from the history file.
If followed by: PTF = ptf-number or APAR = apar-number

component specifies the component, of which a PTF or APAR entry is to be removed from the history file.
component is a string of 11 characters according to the following example: 5745-SC-JCL.

RELease — Indicates that for the given release-number (see below) entries are to be removed.

release-number — Specifies the release number for which the entries specified in the component, PTF, or APAR operand are to be removed, providing the entries pertain to the here given release-number.

PTF — Indicates that the given pft-number (see below) is to be removed.

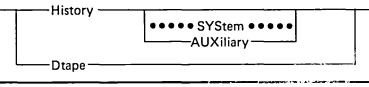
ptf-number — Specifies the PTF number that is to be removed.

ptf-number is a string of 6 or 7 characters, the first one or two being alphabetic, the remaining five being digits.
For example:
N 12345
UD 12345

APAR — Identifies that the given apar-number (see below) is to be removed. The removal applies only to entries made when archiving a local fix.

apar-number — Specifies the APAR number that is to be removed.

apar-number is a string of 6 or 7 characters: one or two alphabetics, followed by five digits.

Function Control Statement	Keyword Operands
RESTore	

APPLICATION/EXPLANATION

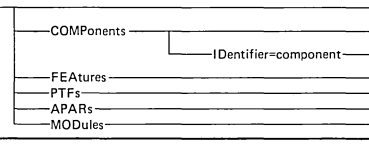
The RESTore statement requests MSHP to write a history file located on magnetic tape onto disk. The tape must be assigned to SYS006 and, for processing, must be positioned to the file containing the history file.

The RESTore Dtape function is normally restricted to copying private libraries. The libraries to be restored must be defined with the appropriate DEFine statements. The tape must be assigned as SYS006 and, for processing, must be positioned to the first file.

Related Detail Control Statements:

Mandatory: DEFine library if RESTore Dtape Optional: DEFine History

History	Identifies this restore operation as an MSHP operation.
SYStem	Specifies that the history file is to be copied to a disk file with the file name: IJSYSHF
AUXiliary	Specifies that the history file is to be copied to a disk file with the file name: IJSYS02
Dtape	Assumes that the SYS006 assigned tape is a backup of libraries and of a history file, created with the BACKUP utility. The libraries are restored as PRIVate libraries subject to and as defined in the DEFine detail control statement. Assignments for the disk, onto which the libraries are to be copied, must be effective as follows: SYS007 for PRIVate CLib SYS008 for PRIVate RLib SYS009 for PRIVate SLib The history file from tape is copied into an auxiliary history file IJSYS02 on disk, assigned as SYS002, or as specified in the Unit-SYSnn parameter of the DEFine statement. Label definitions for this disk file must be effective, or the history file must have been defined in the detail control statement DEFine History AUXiliary, or DEFine History DLIBbuild

Function Control Statement	Keyword Operands
RETrace	

APPLICATION/EXPLANATION

The RETrace statement requests MSHP to print information from the system history file on SYSLST.
Default: If RETrace is specified but none of the keywords, a system overview printout from the history file is produced.

Related Detail Control Statements:

Mandatory: one Optional: DEFine History

COMPonents	Specifies that the component installation records are to be printed.
IDentifier	Identifies that information about the component specified in component (see below) is to be listed.

APPLICATION/EXPLANATION (. . . Cont'd)

component	Specifies the component for which a retrace is to be made. component is a string of 11 characters accordint to the example: 5745-SC-JCL.
FEAtures	Specifies that the features installed are listed.
PTFs	Specifies that all PTFs applied are listed (in ptf-number sequence).
APARs	Specifies that all APARs are listed (in apar-number sequence) which were fixed by a PTF, or were corrected locally.
MODules	Specifies that all phases (core image library members), modules (relocatable library members), and macros (source statement library members) that are affected by a PTF or local fix are listed.

Function Control Statements	Keyword Operands
SELect	PTFfile [APply] [••IRRevokable•• / REVokable] [••NODLIBbuild•• / DLIBbuild] ;

APPLICATION/EXPLANATION

The SELECT statement is used to select individual PTFs from a cumulative PTF file (filename: PTF). Entries are made in the system history file for all the PTFs selected.

The device the SELect function reads the cumulative PTF file from must be assigned to SYS004.

The modules and/or macros of the selected PTFs are catalogued into the corresponding libraries (system or private, if assigned). Linkedits according to "linkbooks" contained in the selected PTF are only done, if the option 'APply' is specified. If 'APply' is specified, OPTION CATAL must be in effect and must have been coded in the job invoking MSHP with the SELect function.

The SELect function can also be utilized to "revoke" individual PTFs.

Selection of Release 34 format PTFs

The cumulative PTF file may contain DOS/VSE-MSHP-format PTFs and PTFs of the Release 34-MSHP-format. For the Release 34-MSHP-format PTFs, MSHP takes history information from JCL comment statements and builds the necessary MSHP control statements.

Restrictions:

- The number of APARS fixed must not exceed 255.
- The PTF must not have more than 72 requisites (PRE, CO and NOT).
- At most 32 PTFs can be superseded.
- The selected PTF may not be applicable to more than six releases.

Related Detail Control Statements:

Mandatory: PTF Optional: none

PTFfile	Distinguishes the selecting of PTFs from the cumulative PTF file from other select operations.
APply	Specifies that the selected PTFs are to be applied to the system. Default: The keyword APply may be omitted when option DLIBbuild is specified (see below). The omission of APply indicates that the modules and/or macros of the selected PTFs are to be catalogued into the corresponding libraries (as in the case when APply is specified) but that linkedits according to the statement INVolves LINK=link-book-list are not to be performed.
IRRevokable	Specifies that when applying the PTF, no backout PTF is generated. That is, the PTF cannot be revoked (the status before the application of the PTF cannot be recreated).
REVokable	Specifies that, when applying the PTF, a backout PTF is generated on SYSPCH. The backout PTF is a job identical in structure with the job containing a PTF, except that the initiating MSHP function control statement inside that job is REVoke component: ptf-number. The "ptf-number" is the same as the one of the original PTF (for which the backout was generated). Restrictions: When REVokable is specified, DLIBbuild must not be specified.

APPLICATION/EXPLANATION (. . . Cont'd)

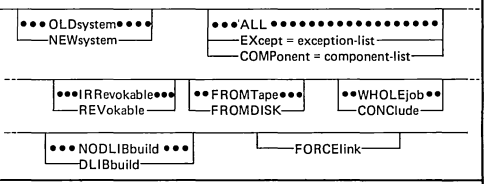
NODLIBbuild — NODLIBbuild suppresses the DLIBbuild option (see below).

DLIBbuild — Specify this option if the PTF is to be applied to preventive maintenance libraries.

The effect of this option is:

1. When a PTF involves linkediting (particularly a 'linkbook' that controls linking a component), all the phase names encountered will be reflected as 'affected' in the system history file so that in the UPgrade function all the phases can be copied from the distribution libraries to the user's libraries.
2. When a PTF needs a co-requisite PTF and this co-requisite is not present yet, application of the PTF will not be rejected (as with the NODLIBbuild option), but a warning message is issued. However, link edits (according to component link-books) will not be performed.

Restrictions: REVokable and DLIBbuild are mutually exclusive options.

Function Control Statement	Keyword Operands
UPGrade	

APPLICATION/EXPLANATION

The UPGrade function is used to apply the contents of IBM-distributed upgrade libraries to an operational system and updates the system history file accordingly, or produces a new history file.
Restrictions: The UPGrade function can only be performed when MSHP is executed in the background partition.
Foreground partitions must be inactive if a library space problem is being resolved by means of condense and FCOPY runs (see CONClude below).

Related Detail Control Statements:

Mandatory: DEFine — Optional: REJect

OLDsystem — Specifies that the history information about the PTFs applied by the UPGrade function is to be inserted in the operational system's history file.

NEWsystem — Specifies that the whole system history file of the current operational system be merged to the MSHP-created auxiliary history file. These merged files, then can become the history file of a new (operational) system.

The auxiliary history file created by MSHP is the restored history file that accompanied the upgrade library package, from which MSHP has removed all entries relating to PTFs that have to be rejected.

ALL — Specifies that all PTFs contained in the upgrade library are applied to the current operational system. Excluded from this application are PTFs that don't satisfy the conditions requested by MSHP. The exceptions are listed in the MSHP USER.s GUIDE under the same option (keyword operand).

EXcept — Identifies that PTFs for the components named in the exception-list (see below) are not to be applied.

(The EXcept option is used to exclude whole components from preventive system service, whereas the REJect detail control statement excludes individual PTFs.)

Exception-list — Specifies the components to be excepted from preventive maintenance.

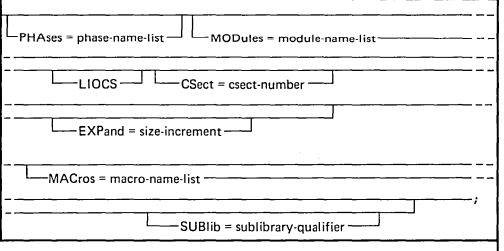
exception is a string of 11 characters according to the example: 5745-SC-JCL.

COMPonent — Identifies that the components named in the component-list (see below) are to be included in the preventive system service.

APPLICATION/EXPLANATION (. . . Cont'd)	
component-list	Specifies the components to be included in the preventive system service. If specified, only the components named are candidates for preventive system service. PTFs for all components not named are rejected. A component may be named for preventive system service in this option, even though there might not be a PTF for it in the PTF upgrade libraries. component is a string of 11 characters according to the example: 5745-SC-JCL.
IRRevokable	Specifies that no backout PTFs will be generated for the PTFs applied during preventive system service.
REVokable	Requests backout PTFs to be generated for all PTFs applied. The backout PTFs are MSHP jobs with the REVoke function control statement included. If SYS006 is assigned to a tape drive, the backout PTFs are written in blocked format to this tape unit. Otherwise, the backout PTFs are written unblocked to SYSPCH.
FROMTape	Specifies that the upgrade libraries and the distribution history file are to be restored. The tape containing the upgrade libraries must be mounted on a tape drive that is assigned as SYS006. MSHP restores the libraries as defined in the DEFine detail control statement; the libraries defined must be of the type PRIVate with the following assignments effective: SYS007 for private CLIB SYS008 for private RLIB SYS009 for private SLIB MSHP restores the distribution history file from tape to an auxiliary history file. An assignment for the auxiliary history file as SYS002 or as specified in the Unit-SYSnnn parameter of the DEFine statement must be effective in any case. From the auxiliary history file all PTF entries are removed for PTFs to be rejected. For the copying, assignments for the restored libraries must be effective as follows: SYS000 for SLIB SYS001 for RLIB SYS003 for CLIB
FROMDisk	When this option is coded, MSHP assumes that the distribution PTF upgrade libraries and the corresponding history file have been restored previously. DEFine detail control statements are now needed to make known to MSHP where the restored libraries and AUXiliary history file are. (The DEFines for the libraries must not specify a directory size.) Assignments for the restored libraries and the auxiliary history file must be effective as follows: SYS000 for SLIB SYS001 for RLIB SYS003 for CLIB For the auxiliary history file, the assignment must be SYS002, or as specified in the Unit=SYSnnn parameter of the DEFine statement.
WHOLEjob	Specifies that restoring the libraries and applying the PTFs is to be done in one continuous job.
CONClude	Specifies that MSHP checks the target libraries for sufficient space before merging the PTF phases, modules, and/or macros from the restored PTF upgrade libraries, or before link-editing.
NODLIBbuild	Suppresses the DLIBbuild option (see below).
DLIBbuild	Specifies that two PTF application libraries (which are interdependent through corequisite PTFs) are to be merged. This function makes application of PTFs acceptable, even if their corequisites are not yet installed, or are not on the same upgrade library. (A warning message is issued, nevertheless.) The libraries involved in applying the PTF are verified to be private libraries. Backout PTFs for such PTFs are generated with the DLIBbuild option in the REVoke statement.

APPLICATION/EXPLANATION (. . . Cont'd)

FORCElink — Specifies that all components upgraded with PTFs from the PTF upgrade libraries are to be linkedited provided at least one PTF for such a component was applied (when building the PTF upgrade libraries) with linkediting suppressed (specifying NOLink in APPLY, or not specifying APply in SELect or ACCept). For additional and further detail information refer to MSHP USER's GUIDE.

Detail Control Statements	Keyword Operands
AFFects	

APPLICATION/EXPLANATION

The AFFects statement identifies which phases, modules, and/or macros are affected by a PTF, or local fix; or which of the replacement modules are for a feature.

PHAses — Identifies that phase(s) as specified in phase-name-list (see below) are affected.

phase-name-list — Specifies the affected phase(s).

Restrictions: If the AFFects statement is used for archiving a local fix and the fix information, list may not be specified, but only the name of the phase to be modified.

phase-name is a string of one to 8 alphameric characters: 0-9, A-Z, #, $, and @

MODules — Identifies that modules (members of the relocatable library) as specified in module-name-list (see below) are affected.

module-name-list — Specifies the affected module(s). (Restrictions and character string same as for phase-name-list above)

LIOCS — Identifies that a LIOCS module is affected by a PTF.

Note that only the macro to generate the module is distributed in the PTF, but not the affected module itself.

CSect — Identifies that an alteration applies to the CSECT specified in CSect-number (see below).

Default: If not specified, CSect = 001 is assumed; that is, the first CSECT.

Restrictions: CSect must only be specified when archiving a local fix.

csect-number — Specifies the CSECT number to which the alteration applies.

csect-number has one to three digits. If less than three digits are specified, the number is padded with leading zeros.

APPLICATION/EXPLANATION (. . . . Cont'd)

EXPand — Identifies that the specified CSECT of the named module is to be made larger, as specified in size-increment (see below), so that fix code can be added at the end of the CSECT.

Restrictions: EXPand must only be specified when archiving a local fix.

size-increment — Specifies the number of bytes by which the CSECT is to be increased.

size-increment is a number of one to six digits.

MACros — Identifies that macros (members of the source statement library) as specified in macro-name-list (see below) are affected.

macro-name-list — Specifies the affected macro(s).

Restrictions: When the AFFects statement is used for archiving a local fix and the fix information, list must not be specified, but only the name of the macro to be modified.

macro-name is string of one to eight alphameric characters:
0-9, A-Z, #, $, @;
the first characters must be alphabetic:
A-Z, #, $, or @.

SUBlib — Identifies that a specific sublibrary of the source statement library contains the affected macro(s).

Default: If not specified, sublibrary E is assumed.

sublibrary-qualifier — Specifies the sublibrary of the affected macro.

sublibrary-qualifier is one alphabetic character: A-Z.

Detail Control Statement	Keyword Operands
ALter	address — old-text — new-text — ;

APPLICATION/EXPLANATION

The ALter statement identifies the modifications that are to be made to a phase (core image library member) or module (relocatable library member). Modification includes verification for core image library members.

address — Specifies the address where the new-text is to begin to replace the old-text.

address is a string of one to six hexadecimal digits.

old-text — Specifies the text that is to be replaced.

The text in the phase or module at the specified address is verified to be identical with the old text; replacement by new text takes only place if identical.

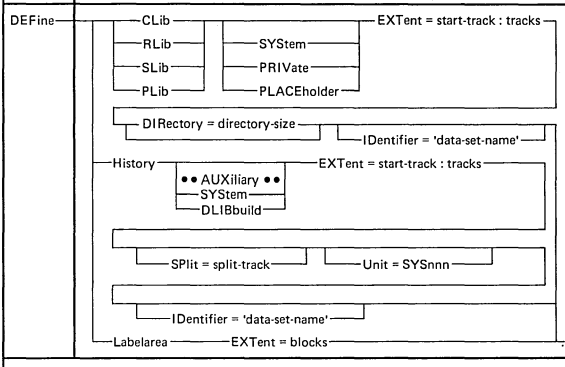
Restrictions: old-text must be specified if modifying a phase, but not when modifying a module.

Old text can be of any of the three formats described in the MSHP USER's GUIDE under the description for the ALter Detail Control Statement.

new-text — Specifies the text that is to replace the text at the specified address.

new-text can consist of any of the three formats as described for old-text in the MSHP USER's GUIDE.

Detail Control Statement	Keyword Operands



APPLICATION/EXPLANATION

The DEFine statement creates label/extent definitions for libraries or history files in the user label area of the partition in which MSHP is executed. Further, it detemines the library allocation for MSHP functions such as INSTall, RESTore, and Dtape, which involve restoring libraries from tape to disk. DEFine Labelarea is used to specify how many FBA blocks are to be allocated for the System Label Area when installing a system.

The following describes the keyword operands of the DEFine statement.

CLib — Identifies that the label definition and/or allocation has to be made for a core image library. A label definition is made only if PRIVate is specified as well; in this case, the file name used is: IJSYSCL.

RLib — Identifies that the label definition and/or allocation has to be made for a relocatable library. A label definition is made only if PRIVate is specified as well; in this case, the file name used is: IJSYSRL.

SLib — Identifies that the label definition and/or allocation has to be made for a source statement library. A label definition is made only if PRIVate is specified as well; in this case, the file name used is: IJSYSSL.

PLib — Identifies that a procedure library is to be restored and that its allocations are as specified in EXTent (see below).

Restrictions: DEFine PLib may be specified only in relation to function control statement

- INSTall

SYStem — Identifies that the library is to be restored as system library.

Restrictions: DEFine SYStem may be specified only in relation to function control statements:

- INSTall
- RESTore Dtape

PRIVate — Identifies that (if applicable) the library is to be restored as private library. Further, a label definition entry is made in the partition label area.

Restrictions: PRIVate may not be specified with PLib.

PLACEholder — Identifies that, when restoring, space should be left for an (empty) library.

Restrictions: PLACEholder must not be specified with CLib.

DEFine PLACEholder may be specified only in relation to the function control statement:

- INSTall

APPLICATION/EXPLANATION (. . . . Cont'd)

Default for SYStem/PRIVate/PLACEholder. If neither SYStem nor PRIVate, nor PLACEholder is specified, the default taken depends on which function control statement the DEFINE statement is detail control statement to.
The default is:
SYStem for INSTall
PRIVate for UPGrade

EXTent — Identifies that extent information (if a label definition entry is made), and the library allocation (when restoring is involved) is to be derived from the specified values in start-track : tracks (see below).

start-track — For CKD devices, specifies the track number relative to zero, where a private library is to begin; the library must begin on a cylinder boundary, except in the case of a private core image library.

For FBA devices, designates the number of the first FBA block of a private library.

Restrictions: start-track must be specified for libraries or type PRIVate; it must not be specified for libraries of type SYStem.

start-track is any decimal number not exceeding 2.147.483.647.

tracks — Specifies, for a private library, what the extent is to be in the label definition; and for any type of library (SYStem or PRIVate), how many tracks/blocks are to be allocated when restoring.

tracks is any decimal number not exceeding 2.147.483.647.

For CKD devices, the number specified is in number of tracks and may be any integral number of cylinders. If less than one cylinder is specified, the minimum allocated is one cylinder; if an odd number of tracks is specified, an integral number of cylinders is allocated.

DIRectory — Identifies that space, as specified in directory-size (see below), is to be allocated for the library directory.

directory-size — For CKD devices, specifies how many tracks must be allocated for the library directory; for FBA devices, how many FBA blocks.
directory-size is a decimal number not exceeding 2.147.483.647.

IDentifier — Idenfies that the libray identifier given in dataset-name (see below) is to be entered in the VTOC.

Dataset-name — Specifies the library identification that is to be entered in the VTOC.

dataset-name is a string, enclosed in quotes, of one to 44 alphameric characters.

Defaults: If dataset-name is not specified, MSHP takes the following defaults:

- For CLib PRIVate : 'DOS.SYSCLB.FILE'
- For RLib PRIVate : 'DOS.SYSRLB.FILE'
- For SLib PRIVate : 'DOS.SYSSLB.FILE'

If DEFine SYStem has been specified and dataset-name is omitted, MSHP takes the default: DOS.SYSRES.FILE

Restriction: If DEFine SYStem RLib/SLib/PLib and dataset-name has been specified, the dataset-name specification is ignored. If CLib has been specified, the VTOC entry for the CLIb is that of the SYSRES file.

History — Identifies that a label definition is to be made for a history file.

AUXiliary — Identifies that the entry in the partition's user label area for the history file is to be made under the file-name:
IJSYS02

SYStem — Identifies that the entry in the partition's user label area for the history file is to be made under the file-name:
IJSYSHP

DLIBbuild — Identifies that entries for the file-names IJSYSHF and IJSYS02 are to be made in the partition's user label area.

Both files have the same extent and identifier as specified or defaulted in EXTent and IDentifier.

APPLICATION/EXPLANATION (. . . . Cont'd)

EXTent	Identifies that extent information and the library allocation is to be derived from the specified values in start-track : tracks.
start-track	For CKD devices, specifies with which track relative to zero the extent for the history file is to begin; for FBA devices with which FBA block number. start-track is any decimal number not exceeding 2.147.483.647.
tracks	Specifies, for the history file, the size of the extent. tracks is a decimal number not exceeding 2.147.483.647. For CKD devices, the number specified is in number of tracks; for FBA devices, the number means FBA blocks.
SPlit	Identifies that the history file is to be maintained as a split-cylinder file; identifies the value given in split-track as the last track to be allocated to the file. Restrictions: SPlit must not be specified for a history file on FBA devices.
split-track	Specifies which track is the last one in each cylinder to be allocated to the history file defined. (The first cylinder occupied by the file ist the one in which the "start-track" lies, and the last cylinder is determined by the number of tracks specified.) split-track is a decimal integer not exceeding 19.
Unit	Identifies that a history file is to reside on the logical unit (other than SYSREC) as specified in SYSnnn. Restrictions: Specification of Unit does not allow for maintaining the system's parameter history file on a logical unit other than SYSREC.
SYSnnn	Specifies the logical unit on which the defined history file is to reside, and assigns the unit to a physical device (CKD or FBA). SYS is followed immediately by three digits (for nnn) representing the physical device address. Default: If not specified, MSHP takes the following defaults: For a SYStem history file: SYSREC For a AUXiliary history file: SYS002 For a DLIBbuild history file: SYS002
IDentifier	Identifies that the history identifier given in dataset-name is to be entered in the VTOC.
'dataset-name'	Specifies the history file identification that is to be entered in the VTOC. dataset-name is a string, enclosed in quotes, of one to 44 alphameric characters. Defaults: If dataset-name is not specified, MSHP takes the following defaults: For History AUXiliary: 'DOS.AUXILIARY.HISTORY.FILE' For History SYStem: 'DOS.SYSTEM.HISTORY.FILE' For History DLIBbuild: 'DOS.DLIB.HISTORY.FILE' Restriction: If you use IBM-supplied standard labels or if your own standard label set contains DLBL and EXTENT for IJSYSHF, do not use DEFINE HISTORY SYSTEM in any MSHP job accessing the system history file.
Labelarea	Identifies that a labelarea is to be allocated when installing a system on a FBA device. Restrictions: DEFine Labelarea may only be specified in conjunction with the function control statement INSTall. Default: If omitted, defaults apply as assumed by the RESTORE utility program.
EXTent	Identifies that the size of the labelarea that will be allocated is to be derived from the value given in blocks (see below).
blocks	Specifies how many FBA blocks are to be allocated for the label area when installing a system. blocks is a decimal number of minimum 12 and not exceeding 2.147.483.647.

Detail Control Statement	Keyword Operands
DELete	from-line :to-line ;

APPLICATION/EXPLANATION

The DELETE statement indicates the lines to be deleted from a source statement library member when archiving a local fix.

from-line — Specifies the line-number, in columns 73 through 78 in the de-edited (E-served) deck, where deletion begins. The from-line is the first line to be deleted.

from-line is an integer of one to six digits.

Default: If omitted, 'from-line' is assumed to be equal to 'to-line'. This means, that only the line designated by to-line (see below) is deleted.

to-line — Identifies the last line of the lines to be deleted.

to-line is an integer of one to six digits.

Restrictions: to-line must numerically be equal to or greater than the value given in from-line.

Detail Control Statement	Keyword Operands
INsert	after-line ;

APPLICATION/EXPLANATION

The INSERT statement identifies where, in a source statement library member, additions are to be made when archiving a local fix.

after-line — Specifies the line number in the de-edited (E-served) deck in columns 73 through 78, after which the source input (following the INsert statement up to the next /$) is to be inserted.

after-line is an integer of one to six digits.

Detail Control Statement	Keyword Operands
PTF	ptf-number [IRRevokable / REVokable] [NODLIBbuild / DLIBbuild] [NOLINK] [COMment = 'comment'] ,

APPLICATION/EXPLANATION

The PTF statement indicates (to SELect) which PTFs are to be selected from the cumulative PTF file. One PTF statement is required for each PTF that is to be selected; they need not to be in any specific sequence.

ptf-numb er — Identifies the PTF that is to be selected from the cumulative PTF file.

ptf-number is a string of 6 or 7 characters, the first one or two are alphabetic, the remaining five are digits.
For example:
N12345 (Nnnnnn is used for "old" format PTFs)
UD12345 (UDnnnnn is used for "new" format PTFs)

IRRevokable — Specifies that, when applying the selected PTF, no backout PTF is to be generated. That is, the PTF cannot be revoked.

Restrictions: See Hierarchy of Irrevokable/Revokable, below.

APPLICATION/EXPLANATION (. . . . Cont'd)

REVokable	Specifies that, when applying the selected PTF, a backout PTF is to be generated. That is, the PTF can be revoked. Restrictions: See "Hierarchy of Irrevokable/Revokable", below. REVokable cannot be specified with the DLIBbuild option specified.
NODLIBbuild	If explicitly specified, suppresses DLIBbuild specification in the SELect or APply statement for the selected PTF.
DLIBbuild	If specified, has the same effect as if specified in the APply statement or the selected PTF. It overrides any defaulted or explicitly specified NODLIBbuild in the APply or SELect statement. For the function of DLIBbuild in APply, see the APPLY statement. Restrictions: DLIBbuild and REVokable are mutually exclusive options.
NOLINK	If specified, has the same effect as if specified in the APPLY statement of the selected PTF; even if in the SELECT statement no indication is given that Linkedits are to be suppressed (the option APply being coded). For the functions of NOLINK, see NOLINK in the APPLY statement.
COMment	Identifies that the comment as specified in comment (see below) is to be inserted in the history file.
'comment'	Specifies that a comment relating to the selected PTF has to be inserted in the history file when the PTF is applied. comment is a string, enclosed in quotes, of one to 57 characters (not counting the quotes).

Hierarchy of Irrevokable/Revokable

Revokability for a PTF may be specified in the SELect statement, the APply statement and in the PTF statement. The following hierarchy holds:

An explicit specification of IRRevokable or REVokable in the PTF statement overrides any other revokability specification (explicit or defaulted).

If irrevokable/revokable is not explicitly specified in the PTF statement, then an explicit specification in the SELect statement becomes the controlling one.

If irrevokable/revokable is not explicitly specified in the PTF or int the SELect statement, then the revokability in the APply statement (explicit or defaulted) becomes effective.

Release 34 format PTFs are always IRRevokable, unless REVokable is specified in the PTF or the SELect statement.

Detail Control Function	Keyword Operands
REJect	component — PTFs = ptf-number-list — ;

APPLICATION/EXPLANATION

The REJect statement indicates which PTFs are explicitly to be excluded if upgrading a system.

Restrictions: The total number of PTFs that are to be rejected per UPGRADE (and specified in one or more REJECT statements) together with the total number of components that are to be excepted must not exceed 113.

component	Specifies the component, for which PTFs (specified in PTFs = ptf-number-list, see below) are to be rejected. component is a string of 11 characters according to the example: 5745-SC-JCL.
PTFs	Identifies that PTFs as specified in ptf-number (see below) are to be excluded from application.
ptf-number-list	Specifies the numbers of the PTFs to be excluded. ptf-number is a string of 6 or 7 characters, the first one or two are alphabetic, the remaining five are digits. For example: N12345 UD12345

Detail Control Statement	Keyword Operands
REPlace ——— from-line : to-line ———;	

APPLICATION/EXPLANATION

The REPlace statement is used, when archiving a local fix. The statement defines where replacement of lines in a source statement library member must begin and end, and initiates the replacing of the source text. (The replacing data are to be terminated by an input line containing /$ in columns 1 and 2.)

from-line	Specifies in the d-edited (E-served) deck the line-number in columns 73 through 78, which is the first line to be deleted and to be replaced. from-line is an integer of one to six digits. If less than six digits are coded, leading zeros are supplied.
to-line	Specifies that beginning with "from-line", all lines in the de-edited (E-served) deck are to be deleted up to and including to-line. to-line is the line-number contained in columns 73 through 78 of the data to be modified. Any input lines contained between the REPlace statement and the next /$ line, are inserted. to-line is an integer of one to six digits. If less than six digits are coded, leading zeros are supplied.

PDZAP

This program allows you to make changes to programs cataloged in a system or private core image library. It provides a printout of the changes on SYSLST. This printout should be kept on the installation to keep track of the changes made.

SYSTEM REQUIREMENTS

PDZAP can be executed in any partition. Since PDZAP accesses a core image library, other programs running currently should not use the phases PDZAP is operating on the same library. SYSLOG must be assigned to the operator console. When card input is used, SYSIPT must be assigned to a card reader (hopper 1 on 5425/2560). SYSLST should be assigned to a printer.

EXECUTING PDZAP

The PDZAP program can be executed from the operator console or from a card reader.

HOW TO EXECUTE PDZAP FROM THE OPERATOR CONSOLE

Teh following is an example in which the program PROG is used as a phase to be modified.

Call in the program
// EXEC PDZAP

The system will respond:
4C86I PDZAP EXECUTION BEGINS
4C99A Enter your name. Reply to this message by typing your name.
4C85A ENTER PHASENAME (XCIL= xxxxxxxx)

Reply to this message in one of the following ways:
a) if PROG is in the system core image library enter SCIL=PROG, or simply PROG, as SCIL is the default.
b) if PROG is in the private core image library enter PCIL=PROG.

When the phase is found, the following messages are issued:
4C87I LOAD ADDRESS=xxxxxx
4C88D ADDRESS XXXXXX, OFFSET+XXXXXX, SCAN=XX..XX,REF=XXXXXX

Reply to this message in one of the following ways:
option 1 specify the hexadecimal address of the data you want displayed: 08072A
option 2 specify the offset to the reference point of the data you want displayed (the initial reference point is the load address): +6D4
option 3 specify a character string to be searched: SCAN='LABPROG'
option 4 specify a hexadecimal string to be searched: SCAN=D3C1C2D7D9D6C7
option 5 set a reference point: REF=08071C.

If an address is invalid or a string cannot be found, an error message will be issued.

To options 3 and 4 the system will first respond with:
4C94I SCAN ADDRESS xxxxxx

Options 1-4 will result in the display of up to 16 bytes of data in the format
4C89D
xxxx..xx cccc..cc

where xxxx..xx is the hexadecimal representation and cccc.cc is the corresponding character representation.

The data printed is contained in a single library block. If less than 16 bytes are displayed, it is either the end of the program or the end of the library block.

If only a display is wanted, press END/ENTER.

To modify the displayed data, type in replacement data for the number of bytes to be changed. For instance, if the data displayed is:
4C89D
05B012107464100000147908428O7F6 6

to change the first four bytes to NOP, type in:
07000700

The system will respond with message 4C88A again and you can use one of the five options mentioned above to display or modify another portion of the same phase, or to set another reference point.

HOW TO EXECUTE PDZAP FROM THE OPERATOR CONSOLE (. . . . Cont'd)

If you are finished with this phase, but want to access another phase, use

- option 6 terminate processing with this phase by typing END PHASE

Now the program repeats message 4C85A, which allows you to specify the name of another phase.

If you want to terminate the execution of PDZAP, use

- option 7 end PDZAP operation by typing: END

HOW TO EXECUTE PDZAP FROM A CARD READER

Executing PDZAP from SYSIPT is the same as from SYSLOG, with the following exceptions:
// UPSI 1 must be specified to indicate card input
data must be verified before it can be changed.

Example:

// UPSI 1	indicate card input
// EXEC PDZAP	call the program
NAME=your name	specify your name
SCIL=PROG	specify the phase to be accessed
+6D4	specifies the position of the data to be displayed (option 2)
VER=05B0, 1211 or	specifies the data to be verified; if the data is
VER=05B01211	not found, no update will take place
REP=07000700 or	specifies the data which is to replace the date
REP=0700,0700	just verified
END	terminates the run

The format of the VER and REP data can be:

- a hexadecimal string (full bytes, or an even number of digits)
- a set of 2-byte entries, separated by commas
- a character string, preceded by a quote (VER='LABPROG).

OUTPUT OF PDZAP ON SYSLST

On SYSLST, which must be a line printer, the program prints the following in the sequence as shown:
PDZAP.
Date and time of the change.
Your name as specified.
Name of the phase that has been changed.
Load address of that phase.
Address of the changed data.
Old data in hexadecimal notation.
New data in hexadecimal notation.

Note: Full details about possible responses to PDZAP messages are given in DOS/VSE messages.

DOSVSDMP AND STAND-ALONE DUMP

DOSVSDMP, A PROGRAM OF DOS/VSE, CAN BE USED TO CREATE A STAND-ALONE DUMP IN CARDS, ON A DISKETTE, ON MAGNETIC TAPE, OR ON DISK. THE PROGRAM CAN ALSO BE USED TO PRINT THE OUTPUT OF:

- A STAND-ALONE DUMP.
- A DUMP TAKEN IN RESPONSE TO A DUMP COMMAND IF THE OUTPUT WAS WRITTEN ONTO MAGNETIC TAPE OR DISK.
- AN EXECUTION OF THE SDAID PROGRAM IF ITS OUTPUT WAS WRITTEN ONTO MAGNETIC TAPE.

1. RECORD ANY ERROR OR STATUS INDICATORS ON THE SYSTEM CONSOLE.
2. USING ALTER/DISPLAY, RECORD THE CURRENT PSW, GENERAL REGISTERS, AND FLOATING POINT REGISTERS.
3. PERFORM THE STORE STATUS (43XX PROCESSORS SAVE MACHINE) PROCEDURE FOR YOUR SYSTEM.
4. IPL YOUR DUMP PROGRAM FROM CARDREADER, TAPE, DISKETTE, OR DISK.
5. IF MESSAGES 4C43I DOSVS DMP COMPETE 0000XX TRACKS USED, RESTART DOS/VSE AND PREPARE TO PRINT THE DUMP AS SOON AS THE SYSTEM HAS BEEN RESTARTED.

CREATING THE STAND-ALONE DUMP PROGRAM

THE PROGRAM, ONCE IT RECEIVED CONTROL, PROMTS YOU FOR FURTHER CONTROL INFORMATION, AND YOU SELECT THE DESIRED OPTION BY AN APPROPRIATE RESPONSE TO THE PROGRAM'S PROMT AS SHOWN.

PROMT MESSAGE

XX 4C50D SELECT YOUR OPTION BY THE CORRESPONDING NUMBER

1. CREATE DOSDMPF	2 PRINT DUMP	3 PRINT SDAID TAPE
4 PRINT FILE	5 INFORMATION	6 EOJ (DEFAULT)
7 CREATE DOSDMPG	8 CLR DOSDMPF	9 CLR DOSDMPG

EXAMPLE FOR JOB CONTROL

TAPE DUMP:

1. SELECT A NON-LABELED TAPE TO BE USED AS THE DUMP VOLUME.
2. EXECUTE DOSVSDMP:
   ```
   // JOB
   // ASSGN SYS006, 280
   // EXEC DOSVSDMP
   /&
   ```
3. WHEN MESSAGE 4C50D IS ISSUED, REPLY: 1
   (1 CREATE DOSDUMPF) (6 EOJ DEFAULT)
4. MESSAGE 4C79 I INDICATES THAT CREATION OF THE DUMP PROGRAM TAPE IS COMPLETED.
   REFER TO DOS/VSE MESSAGES FOR AN EXPLANATION OF THE ABOVE MESSAGES.

DISK DUMP:

1. SELECT A DISK PACK TO BE USED AS THE DUMP VOLUME.
   NOTE THAT THE IPL TEXT ON THE PACK WILL BE REPLACED BY THE DUMP IPL TEXT. THE SELECTED PACK CANNOT BE A SYSTEM RESIDENCE VOLUME.
2. BEFORE YOU CAN EXECUTE DOSVSDMP, THE FILE WHICH IS TO HOLD THE DUMP OUTPUT MUST BE ALLOCATED ON THE SELECTED VOLUME. THIS FILE MUST BE NAMED DOSDMPF, AND BEGIN AND END ON A CYLINDER BOUNDARY. THE SIZE OF THE FILE IS CALCULATED USING THE FOLLOWING FORMULA:

   FOR CKD DEVICE:
   $N = 2 + (V+R) / T$

   FOR AN FBA DEVICE USING BLOCKS OF 512 BYTES IN LENGTH:
   $N = 4 (7 + (V+R) (1 + 1/256))$

EXAMPLE FOR JOB CONTROL (. . . . Cont'd)

Where

N = IS THE NUMBER OF REQUIRED TRACKS FOR A CKD DEVICE AND THE NUMBER OF BLOCKS FOR AN FBA DEVICE.

V = IS THE NUMBER OF PAGES IN THE VIRTUAL ADDRESS AREA.

R = IS THE NUMBER OF PAGES IN THE REAL ADDRESS AREA.

T = 3 FOR A VOLUME ON A 2314 (2319) OR 3340
6 FOR A VOLUME ON A 3330
8 FOR A VOLUME ON A 3350

THE RESULT N MUST BE ROUNDED TO THE NEXT HIGHER NUMBER OF TRACKS EQUIVALENT TO THE NEXT WHOLE NUMBER OF CYLINDERS.

EXAMPLE:

3. EXECUTE DOSVSDMP
```
// JOB
// ASSGN SYS006, 132
// DLBL DOSDMPF, 'FILENAME'
// EXTENT SYS006, BALANCE OF EXTENT INFORMATION
// EXEC DOSVSDMP
/&
```
4. WHEN MESSAGE 4C50D IS ISSUED, REPLY: 1,6
(1 CREATE DOSMPF) (6 EOJ DEFAULT)

5. MESSAGE 4C79 I INDICATES THAT CREATION OF THE DUMP PROGRAM DISK IS COMPLETED.
REFER TO DOS/VSE MESSAGES FOR AN EXPLANATION OF THE ABOVE MESSAGES.

DISKETTE OR IN CARDS

ASSIGN SYS006 TO A DISKETTE OR A CARD UNIT. THIS CAUSES PROGRAM DOSVSDMP TO PROMPT YOU FOR THE DESIRED OUTPUT MODE AS SHOWN.

PROMT MESSAGE

```
XX4C51D   SELECT YOUR OPTION BY NUMBER
1 cuu     PRINTER ADDRESS (DEFAULT IS 1=00E)
2 INTERRUPT          3 EOJ
```

EXAMPLE JOB CONTROL: DISKETTE AND CARDS

DISKETTE DUMP:

1. EXECUTE DOSVSDMP:
```
// JOB
// ASSGN SYS006, 04F
// DLBL DOSDMPF, 'DATA', 77/001, DU
// EXTENT SYS006
// EXEC DOSVSDMP
/&
```

CARD DUMP:

1. EXECUTE DOSVSDMP:
```
// JOB
// ASSGN SYS006, 00D
// EXEC DOSVSDMP
/&
```

EXECUTING THE STAND-ALONE DUMP PROGRAM

THE PROGRAM IS ACTIVATED VIA AN IPL OF THE TAPE, DISK, DISKETTE OR CARD-READER CONTAINING THE DUMP PROGRAM. THE IPL PROCEDURE LOADS THE PROGRAM INTO AND OVERLAYS STORAGE LOCATIONS X'A00' TO X'AFF'. ONCE LOADED, THE PROGRAM PROCEEDS TO WRITE OUT STORAGE IN VIRTUAL PAGE ORDER. FOLLOWING THE VIRTUAL STORAGE DUMP, OR AS A RESULT OF AN ERROR IN ATTEMPTING TO DUMP VIRTUAL STORAGE, A DUMP OF REAL STORAGE IN REAL PAGE ORDER IS TAKEN. UPON COMPLETION OF THE DUMP MESSAGE DOSVSDMP COMPLETE IS ISSUED AND THE SYSTEM IS PLACED IN THE WAIT STATE WITH A COMPLETION CODE IN BITS 48 TO 63 OF THE CURRENT PSW.

PRINTING THE STAND-ALONE DUMP OUTPUT

FOR PRINTING THE STAND-ALONE DUMP OUTPUT, THE PROGRAM REQUIRES AT LEAST 40 K OF STORAGE, NOT INCLUDING THE SIZE OF THE SUPERVISOR.

EXAMPLE FOR JOB CONTROL: PRINTING FROM DISK

1. EXECUTE DOSVSDMP:
```
// JOB
// ASSGN SYS006,DEVICE
// DLBL DOSDMPF, 'FILENAME'
// EXTENT SYS006, BALANCE OF EXTENT INFORMATION
// EXEC DOSVSDMP
/&
```
2. WHEN MESSAGE XX4C50D IS ISSUED, REPLY: 2
   (2 PRINT DUMP)
3. WHEN MESSAGE XX4C52D IS ISSUED, REPLY:
   SELECT YOUR DUMP(S) BY NUMBER.

PROMPT MESSAGE

```
XX 4C50D  SELECT YOUR OPTION BY THE CORRESPONDING NUMBER
1 CREATE DOSDMPF      2 PRINT DUMP        3 PRINT SDAID TAPE
4 PRINT FILE          5 INFORMATION       6 EOJ (DEFAULT)
7 CREATE DOSDMPG      8 CLR DOSDMPF       9 CLR DOSDMPG

2 <----------OPERATOR'S RESPONSE
```

PROMPT MESSAGE

```
XX 4C52D  SELECT YOUR DUMP(S) BY NUMBER
1 SUPVR   2 VIRT (DEFLT)   3 REAL          4 FORMATTED
5 SVA     6 BG OR F1 ETC   7 ADDR-ADDR     8 END SELECT
9 EOJ     EXAMPLE: 5,6 BG,6 F1,7 0-37FF,8

1, 6 F2, 4, 8 <----------OPERATOR'S RESPONSE
```

BRIEF EXPLANATION OF POSSIBLE DUMP OPTIONS:

1—PRINT THE CONTENTS OF THE SUPERVISOR.
2—PRINT THE CONTENTS OF ALL OF VIRTUAL STORAGE (THIS IS THE DEFAULT IF NO DUMP OUTPUT OPTIONS ARE SPECIFIED; THAT IS, IF YOU RESPOND BY SIMPLY PRESSING END/ENTER.
3—PRINT THE CONTENTS OF PROCESSOR (REAL) STORAGE (APPLIES ONLY TO DOS/VSE IN 370 MODE).
4—FORMAT AND PRINT DOS/VSE CONTROL BLOCKS.
5—PRINT THE CONTENTS OF THE SVA.
6—PRINT THE CONTENTS OF THE SPECIFIED PARTITION(S).
7—PRINT THE CONTENTS OF ONE OR MORE (UP TO EIGHT) AREAS OF VIRTUAL STORAGE AS DEFINED BY PAIRS OF ADDRESSES SPECIFIED IN RESPONSE IN SUCCESSIVE PROGRAM PROMPTS. THESE ADDRESSES MUST BE SPECIFIED IN HEXADECIMAL NOTATION. IF YOU SPECIFY, FOR EXAMPLE

```
7 2001F-20 300
```

THE PROGRAM PRINTS THE CONTENTS OF ONE PAGE OF VIRTUAL STORAGE FROM 200 00 to 207FF) BECAUSE PRINTING FOR AN ADDRESS PAIR ALWAYS BEGINS AT THE NEXT LOWER 2K BOUNDARY OF THE ADDRESS SPECIFIED FIRST AND ENDS WITH THE NEXT HIGHER 2K BOUNDARY OF THE ADDRESS SPECIFIED LAST. IF YOU SPECIFY 2K BOUNDARIES, PRINTING STARTS AND ENDS ON THE SPECIFIED BOUNDARIES.

8—INDICATES TO DOSVSDMP THAT YOU HAVE FINISHED SELECTING DUMP OUTPUT OPTIONS.
9—TERMINATE EXECUTION OF DOSVSDMP IMMEDIATELY.

PROMPTS AND RESPONSES FOR PRINTING FROM TAPE OR DISK.
FOR DETAILS REFER TO DOS/VSE SERVICEABILITY AIDS AND DEBUGGING PROCEDURES.

INDEX

C (. . . . Cont'd)

D (. . . . Cont'd)

E

F

G

H

I

J

L

P (. . . . Cont'd)

Q

R

S

S (. . . . Cont'd)

T

U

V

W

X

Y

Z

THIS PAGE HAS BEEN ADDED FOR YOUR OWN NOTES

THIS PAGE HAS BEEN ADDED FOR YOUR OWN NOTES

THIS PAGE HAS BEEN ADDED FOR YOUR OWN NOTES

THIS PAGE HAS BEEN ADDED FOR YOUR OWN NOTES

THIS PAGE HAS BEEN ADDED FOR YOUR OWN NOTES

SY33-8571-7

DOS/VSE Handbook S 370 / S 4300-40 Printed in USA SY33-8571-7

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

DOS/VSE Handbook
SY33-8571-7

READER'S COMMENT FORM

This sheet is for comments and suggestions about this manual. We would appreciate *your* views, favorable or unfavorable, in order to aid us in improving *this* publication. This form will be sent directly to the author's department. Please include your name and address if you wish a reply. Contact your IBM branch office for answers to technical questions about the system or when requesting additional publications. Thank you.

Your comments* and suggestions:

* We would especially appreciate your comments on any of the following topics:

Clarity of the text	Accuracy	Index	Illustrations	Appearance	Paper
Organization of the text	Cross-references	Tables	Examples	Printing	Binding

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 812 BP
1133 Westchester Avenue
White Plains, New York 10604