GC24-5091-5
File No. S370-36

Systems

OS/VS1 Programmer's Reference Digest

Release 6

IBM

Preface

This publication contains quick reference information for the experienced programmer and systems support personnel. For the most part, definitions, restrictions, and limitations have been omitted to provide the most rapid access to the information in this publication. If the reference to information included here is not sufficient, refer to the publication list on the first page of each section; then refer to the applicable System Reference Library publication.

For information concerning the Mass Storage System and allied devices, see *Mass Storage System (MSS) Principles of Operations*, GA32-0029.

This publication does not contain information about system control blocks. Refer to *OS/VS1 System Data Areas*, SY28-0605, to find this information. Some information useful in debugging the system is contained in this publication. For additional information, refer to *OS/VS1 Debugging Guide*, GC24-5093.

Sixth Edition (November 1975)

This edition, together with Technical Newsletter GN24-5525, applies to Release 6 of OS/VS1 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the *IBM System/370 Bibliography*, GC20-0001 for the editions that are applicable and current.

Summary of Amendments

For a list of changes, see page 3, and the TNL (GN24-5525) cover letter.

Information on the 3203-4 and ECPS (Extended Control-Program Support) is for planning purposes only until the availability of the 3203-4 and S/370 Models 135-3, 138, 145-3, and 148.

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.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Programming Publications, Dept. G60, P.O. Box 6, Endicott, New York 13760. Comments become the property of IBM.

IBM Technical Newsletter

This TNL No. GN24-5525
Date September 15, 1976
Base Publ. No. GC24-5091-5
File No. S370-36 (OS/VS1 Release 6)
Previous TNLs None

OS/VS1 Programmer's Reference Digest

This Technical Newsletter, a part of Release 6 of OS/VS1, provides replacement pages for your publication. These replacement pages remain in effect for subsequent OS/VS1 releases unless specifically altered. Pages to be inserted and/or removed are:

Cover through 6	4-5, 4-6	8-55, 8-56
7, 8 (Removed)	4-9, 4-10	8-69 through 8-82 (Removed)
1-7, 1-8	4-13, 4-14	Section 9
2-3 through 2-18	5-7 through 5-20	Index 5, 6
2-31 through 2-34	6-7, 6-8	11, 12
2-41, 2-42 (Added)	8-1 through 8-8	15 through 18
3-5, 3-6	8-45 through 8-48	
3-11 through 3-26		

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

This Technical Newsletter contains editorial changes and additions for Release 6, including:

UCB Sense Bytes for Channel-to Channel Adapter and 3203-4
Trace Table entry for External Interrupts
PGOUT macro outline
OPEN macro changes for Open/Extend support
New console reply format
3800 Printer BURST Parameter
3790 RES Workstation Command Changes
Simplified WRITER Command
New UCS Operands for the ICAPRTBL Utility
Extended Control - Program Support

SMP Reference Summary information has been removed from Section 8; refer to *OS/VS System Modification Program (SMP), System Programmer's Guide,* GC28-0673 for this information.

Information on the 3203-4 and ECPS (Extended Control-Program Support) is for planning purposes only until the availability of the 3203-4 and S/370 Models 135-3, 138, 145-3, and 148.

Note: *Please insert this page in your publication to provide a record of changes.*

IBM Corporation, Programming Publications, Dept. G60, P.O. Box 6, Endicott, N.Y. 13760

 Printed in U.S.A.

Summary of Amendments for GC24-5091-5 as updated by GN24-5525 VS1 Release 6

Sections 2, 3, 4, 5, and 8 contain information included for the first time. This material includes:

SECTION 2

UCB Sense Bytes for Channel-to-Channel Adapter and 3203-4
New Trace Table entry for External Interrupts
Virtual Storage Organization
New Wait State Code for Extended Control-Program Support (ECPS)

SECTION 3

PGOUT Macro outline

SECTION 4

OPEN Macro changes for Open/Extend support
3800 Printer BURST parameter

SECTION 5

3790 RES Workstation Command Changes
Simplified WRITER Command
New Console Reply Format

SECTION 8

The addition of the DEVT, FCB, and UCS operands for the ICAPRTBL utility.

Information on the 3203-4 and ECPS (Extended Control-Program Support) is for planning purposes only until the availability of the 3203-4 and S/370 Models 135-3, 138, 145-3, and 148.

Summary of Amendments for GC24-5091-5 VS1 Release 5

Sections 1, 2, 4, 7, and 8 contain information included for the first time. This material includes:

SECTION 1

Fractional-value Decimal/Hexadecimal conversion table
Command Codes for 3350 DASD
Virtual-to-Real Address Translation

SECTION 2

UCB Sense Bytes for 3350 DASD, 3800 Printing Subsystem
Device Statistics Table for 3540

SECTION 4

Device Capacities for 3350
3540 Standard Labels

SECTION 7

VTAM Network Operator Commands

SECTION 8

3800 Printer high density dump parameters
IEBIMAGE, a 3800 data-set utility

The 3350 Direct Access Storage information and 3800 Printing Subsystem information contained in this publicaion is for planning purposes only until the products become available.

Summary of Amendments for GC24-5091-4 VS1 Release 4

Sections 2 and 8 contain information included for the first time. This material includes:

SECTION 2

New-format Trace Table
UCB sense bytes for 3540
VS1 Storage Subpools

SECTION 8

SMP Reference Summary

In addition, all sections contain substantial additions, corrections, and improvements. Review them in their entirety for new and modified information.

Contents

Section 7: BTAM/TCAM/VTAM

Section 8: Utilities and Service Aids

This section contains a variety of general information needed to write assembler language programs.

Source Publications

Additional information about the System/370 and valid instructions is contained in *IBM System/370 Principles of Operation*, GA22-7000.

Additional information about the System Assembler is in *OS/VS, DOS/VS, and VM/370 Assembler Language*, GC33-4010.

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	ENQ		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

Code Translation Table (cont'd)

Dec.	Hex	Instruction (RX)	Graphics and Controls BCDIC	EBCDIC(1)		ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
64	40	STH		Sp	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	‡	ı	ı	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	Δ	¬	¬	–	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	¥	_	_	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	√	"	"	DEL	8 4 2 1	7-8	0111 1111

Code Translation Table (cont'd)

Dec.	Hex	Instruction and Format	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
128	80	SSM -S					12-0-1-8	1000 0000
129	81			a	a		12-0-1	1000 0001
130	82	LPSW -S		b	b		12-0-2	1000 0010
131	83	Diagnose		c	c		12-0-3	1000 0011
132	84	WRD -SI		d	d		12-0-4	1000 0100
133	85	RDD -SI		e	e		12-0-5	1000 0101
134	86	BXH		f	f		12-0-6	1000 0110
135	87	BXLE		g	g		12-0-7	1000 0111
136	88	SRL		h	h		12-0-8	1000 1000
137	89	SLL		i	i		12-0-9	1000 1001
138	8A	SRA					12-0-2-8	1000 1010
139	8B	SLA -RS			{		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 -SI		j	j		12-11-1	1001 0001
146	92	MVI -SI		k	k		12-11-2	1001 0010
147	93	TS -S		l	l		12-11-3	1001 0011
148	94	NI		m	m		12-11-4	1001 0100
149	95	CLI		n	n		12-11-5	1001 0101
150	96	OI -SI		o	o		12-11-6	1001 0110
151	97	XI		p	p		12-11-7	1001 0111
152	98	LM -RS		q	q		12-11-8	1001 1000
153	99			r	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 -S			¤		12-11-4-8	1001 1100
157	9D	TIO, CLRIO -S			)		12-11-5-8	1001 1101
158	9E	HIO, HDV -S			±		12-11-6-8	1001 1110
159	9F	TCH -S			■		12-11-7-8	1001 1111
160	A0						11-0-1-8	1010 0000
161	A1			~	°		11-0-1	1010 0001
162	A2			s	s		11-0-2	1010 0010
163	A3			t	t		11-0-3	1010 0011
164	A4			u	u		11-0-4	1010 0100
165	A5			v	v		11-0-5	1010 0101
166	A6			w	w		11-0-6	1010 0110
167	A7			x	x		11-0-7	1010 0111
168	A8			y	y		11-0-8	1010 1000
169	A9			z	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 -SI			⌐		11-0-4-8	1010 1100
173	AD	STOSM -SI			[		11-0-5-8	1010 1101
174	AE	SIGP -RS			≥		11-0-6-8	1010 1110
175	AF	MC -SI			●		11-0-7-8	1010 1111
176	B0				0		12-11-0-1-8	1011 0000
177	B1	LRA -RX			1		12-11-0-1	1011 0001
178	B2	See below			2		12-11-0-2	1011 0010
179	B3				3		12-11-0-3	1011 0011
180	B4				4		12-11-0-4	1011 0100
181	B5				5		12-11-0-5	1011 0101
182	B6	STCTL -RS			6		12-11-0-6	1011 0110
183	B7	LCTL -RS			7		12-11-0-7	1011 0111
184	B8				8		12-11-0-8	1011 1000
185	B9				9		12-11-0-9	1011 1001
186	BA	CS -RS					12-11-0-2-8	1011 1010
187	BB	CDS -RS			⌟		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 -RS			≠		12-11-0-6-8	1011 1110
191	BF	ICM -RS			—		12-11-0-7-8	1011 1111

Code Translation Table (cont'd)

Dec.	Hex	Instruction (SS)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
192	C0		?	{		B A 8 2	12-0	1100 0000
193	C1		A	A	A	B A 1	12-1	1100 0001
194	C2		B	B	B	B A 2	12-2	1100 0010
195	C3		C	C	C	B A 2 1	12-3	1100 0011
196	C4		D	D	D	B A 4	12-4	1100 0100
197	C5		E	E	E	B A 4 1	12-5	1100 0101
198	C6		F	F	F	B A 4 2	12-6	1100 0110
199	C7		G	G	G	B A 4 2 1	12-7	1100 0111
200	C8		H	H	H	B A 8	12-8	1100 1000
201	C9		I	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	J	B 1	11-1	1101 0001
210	D2	MVC	K	K	K	B 2	11-2	1101 0010
211	D3	MVZ	L	L	L	B 2 1	11-3	1101 0011
212	D4	NC	M	M	M	B 4	11-4	1101 0100
213	D5	CLC	N	N	N	B 4 1	11-5	1101 0101
214	D6	OC	O	O	O	B 4 2	11-6	1101 0110
215	D7	XC	P	P	P	B 4 2 1	11-7	1101 0111
216	D8		Q	Q	Q	B 8	11-8	1101 1000
217	D9		R	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	S	A 2	0-2	1110 0010
227	E3		T	T	T	A 2 1	0-3	1110 0011
228	E4		U	U	U	A 4	0-4	1110 0100
229	E5		V	V	V	A 4 1	0-5	1110 0101
230	E6		W	W	W	A 4 2	0-6	1110 0110
231	E7		X	X	X	A 4 2 1	0-7	1110 0111
232	E8		Y	Y	Y	A 8	0-8	1110 1000
233	E9		Z	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	0	8 2	0	1111 0000
241	F1	MVO	1	1	1	1	1	1111 0001
242	F2	PACK	2	2	2	2	2	1111 0010
243	F3	UNPK	3	3	3	2 1	3	1111 0011
244	F4		4	4	4	4	4	1111 0100
245	F5		5	5	5	4 1	5	1111 0101
246	F6		6	6	6	4 2	6	1111 0110
247	F7		7	7	7	4 2 1	7	1111 0111
248	F8	ZAP	8	8	8	8	8	1111 1000
249	F9	CP	9	9	9	8 1	9	1111 1001
250	FA	AP					12-11-0-2-8-9	1111 1010
251	FB	SP					12-11-0-3-8-9	1111 1011

Code Translation Table (cont'd) - Machine Instruction Formats

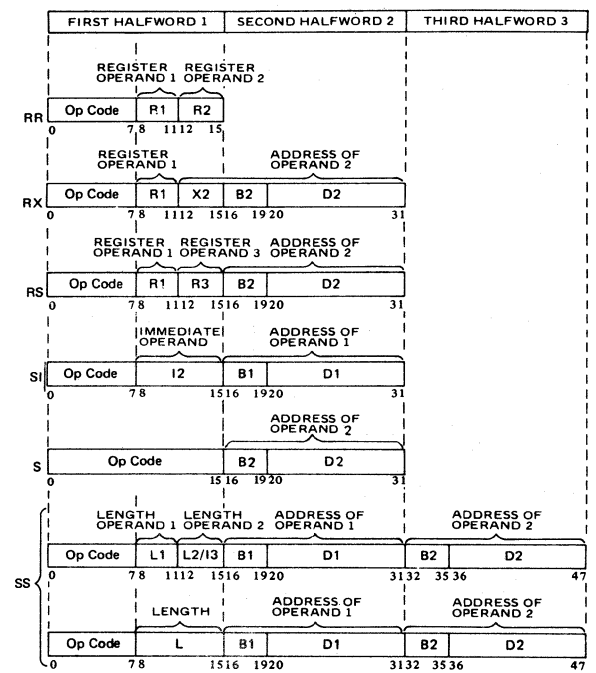
Dec.	Hex	Instruction (SS)	Graphics and Controls BCDIC EBCDIC(1) ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
252	FC	MP			12-11-0-4-8-9	1111 1100
253	FD	DP			12-11-0-5-8-9	1111 1101
254	FE				12-11-0-6-8-9	1111 1110
255	FF				12-11-0-7-8-9	1111 1111

1. Two columns of EBCDIC graphics are shown. The first gives standard bit pattern assignments. The second shows the T-11 and TN text printing chains (120 graphics).
2. Add C (check bit) for odd or even parity as needed, except as noted.
3. For even parity use CA.

Op code (S format)

B202 - STIDP	B207 - STCKC	B20D - PTLB
B203 - STIDC	B208 - SPT	B210 - SPX
B204 - SCK	B209 - STPT	B211 - STPX
B205 - STCK	B20A - SPKA	B212 - STAP
B206 - SCKC	B20B - IPK	B213 - RRB

Machine Instruction Formats



Control Registers

CR	Bits	Name of field	Associated with	Init.
0	0	Block - multiplex'g control	Block - multiplex'g	0
	1	SSM suppression control	SSM instruction	0
	2	TOD clock sync control	Multiprocessing	0
	8 - 9	Page size control	Dynamic addr. transl.	0
	10	Unassigned (must be zero)		0
	11 - 12	Segment size control		0
	16	Malfunction alert mask	Multiprocessing	0
	17	Emergency signal mask		0
	18	External call mask		0
	19	TOD clock sync check mask		0
	20	Clock comparator mask	Clock comparator	0
	21	CPU timer mask	CPU timer	0
	24	Interval timer mask	Interval timer	1
	25	Interrupt key mask	Interrupt key	1
	26	External signal mask	External signal	1
1	0 - 7	Segment table length	Dynamic addr. transl.	0
	8-31	Segment table address		0
2	0 - 31	Channel masks	Channels	1
8	16 - 31	Monitor masks	Monitoring	0
9	0	Successful branching event mask	Program - event record'g	0
	1	Instruction fetching event mask		0
	2	Storage alteration event mask		0
	3	GR alteration event mask		
	16 - 31	PER general register masks		0
10	8 - 31	PER starting address	Program - event record'g	0
11	8 - 31	PER ending address	Program - event record'g	0
14	0	Check - stop control	Machine - check handling	1
	1	Synch. MCEL control		1
	2	I/O extended logout control	I/O extended logout	0
	4	Recovery report mask	Machine - check handling	0
	5	Degradation report mask		0
	6	Ext. damage report mask		1
	7	Warning mask		0
	8	Asynch. MCEL control		0
	9	Asynch. fixed log control		0
15	8 - 28	MCEL address	Machine - check handling	512

Condition Codes

Condition Code Setting	0	1	2	3
Mask Bit Value	8	4	2	1
General Instructions				
Add, Add Halfword	zero	< zero	> zero	overflow
Add Logical	zero, no carry	not zero, no carry	zero, carry	not zero, carry
AND	zero	not zero	—	—
Compare, Compare Halfword	equal	1st op low	1st op high	—
Compare and Swap/Double	equal	not equal	—	—
Compare Logical	equal	1st op low	1st op high	—
Exclusive OR	zero	not zero	—	—
Insert Characters under Mask	all zero	1st bit one	1st bit zero	—
Load and Test	zero	< zero	> zero	—
Load Complement	zero	< zero	> zero	overflow
Load Negative	zero	< zero	—	—
Load Positive	zero	—	> zero	overflow
Move Long	count equal	count low	count high	overlap
OR	zero	not zero	—	—
Shift Left Double/Single	zero	< zero	> zero	overflow
Shift Right Double/Single	zero	< zero	> zero	—
Store Clock	set	not set	error	not oper
Subtract, Subtract Halfword	zero	< zero	> zero	overflow
Subtract Logical	—	not zero, no carry	zero, carry	not zero, carry
Test and Set	zero	one	—	—
Test under Mask	zero	mixed	—	ones
Translate and Test	zero	incomplete	complete	—
Decimal Instructions				
Add Decimal	zero	< zero	> zero	overflow
Compare Decimal	equal	1st op low	1st op high	—
Edit, Edit and Mark	zero	< zero	> zero	—
Shift and Round Decimal	zero	< zero	> zero	overflow
Subtract Decimal	zero	< zero	> zero	overflow
Zero and Add	zero	< zero	> zero	overflow
Floating-Point Instructions				
Add Normalized	zero	< zero	> zero	—
Add Unnormalized	zero	< zero	> zero	—
Compare	equal	1st op low	1st op high	—
Load and Test	zero	< zero	> zero	—
Load Complement	zero	< zero	> zero	—
Load Negative	zero	< zero	—	—
Load Positive	zero	—	> zero	—
Subtract Normalized	zero	< zero	> zero	—
Subtract Unnormalized	zero	< zero	> zero	—
Input/Output Instructions				
Clear I/O	no oper in progress	CSW stored	chan busy	not oper
Halt Device	interruption pending	CSW stored	channel working	not oper
Halt I/O	interruption pending	CSW stored	burst op stopped	not oper
Start I/O, SIOF	successful	CSW stored	busy	not oper
Store Channel ID	ID stored	CSW stored	busy	not oper
Test Channel	available	interruption pending	burst mode	not oper
Test I/O	available	CSW stored	busy	not oper
System Control Instructions				
Load Real Address	translation available	ST entry invalid	PT entry invalid	length violation
Reset Reference Bit	R=0, C=0	R=0, C=1	R=1, C=0	R=1, C=1
Set Clock	set	secure	—	not oper
Signal Processor	accepted	stat stored	busy	not oper

Program Interruption Codes - CNOP Alignment - Edit EDMK Pattern Characters

PROGRAM INTERRUPTION CODES

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	64	0040	Monitor event
10	000A	Decimal overflow	128	0080	Program event (code may be combined with another code)
11	000B	Decimal divide			

CNOP ALIGNMENT

Double Word							
Word				Word			
Half Word		Half Word		Half Word		Half Word	
Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte
0,4		2,4		0,4		2,4	
0,8		2,8		4,8		6,8	

EDIT AND EDMK PATTERN CHARACTERS (in hex)

20 - digit selector	40 - blank	5C - asterisk
21 - start of significance	4B - period	6B - comma
22 - field separator	5B - dollar sign	C3D9 - CR

Fixed Storage Locations

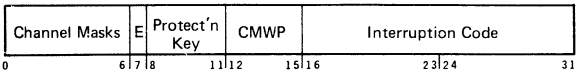
Area, dec.	Hex addr	Purpose
0-7	0	Initial program loading PSW, restart new PSW
8-15	8	Initial program loading CCW1, restart old PSW
16-23	10	Initial program loading CCW2
24-31	18	External old PSW
32-39	20	Supervisor Call old PSW
40-47	28	Program old PSW
48-55	30	Machine-check old PSW
56-63	38	Input/output old PSW
64-71	40	Channel status word
72-75	48	Channel address word
80-83	50	Interval timer
88-95	58	External new PSW
96-103	60	Supervisor Call new PSW
104-111	68	Program new PSW
112-119	70	Machine-check new PSW
120-127	78	Input/output new PSW
132-133	84	CPU address assoc'd with external interruption , or unchanged
132-133	84	CPU address assoc'd with external interruption, or zero (EC mode only)
134-135	86	External interruption code (EC mode only)
136-139	88	SVC interruption [0-12 zeros, 13-14 ILC, 15:0, 16-31 code] (EC mode only)
140-143	8C	Program interrupt [0-12 zeros, 13-14 ILC, 15:0, 16-31 code] (EC mode only)
144-147	90	Translation exception address [0-7 zeros, 8-31 address] (EC mode only)
148-149	94	Monitor class [0-7 zeros, 8-15 class number]
150-151	96	PER interruption code [0-3 code, 4-15 zeros] (EC mode only)
152-155	98	PER address [0-7 zeros, 8-31 address] (EC mode only)
156-159	9C	Monitor code [0-7 zeros, 8-31 monitor code]
168-171	AB	Channel ID [0-3 type, 4-15 model, 16-31 max. IOEL length]
172-175	AC	I/O extended logout (IOEL) address [0-7 unused, 8-31 addr]
176-179	B0	Limited channel logout (see diagram)
185-187	B9	I/O address [0-7 zeros, 8-23 address] (EC mode only)
216-223	D8	CPU timer save area
224-231	E0	Clock comparator save area
232-239	E8	Machine-check interruption code
248-251	F8	Failing processor storage address [0-7 zeros, 8-31 addr]
252-255	FC	Region code*
256-351	100	Machine-check fixed logout area*
352-383	160	Machine-check floating-point register save area
384-447	180	Machine-check general register save area
448-511	1C0	Machine-check control register save area
512-†	200	Machine-check CPU extended logout area (size varies)

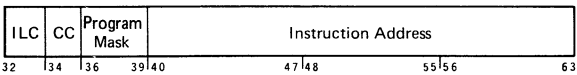
* Functions and use of fields may vary among models. See system library manuals for specific model.

† Location may be changed by programming (bits 8-28 of CR15 specify address).

PSW (BC and EC modes)

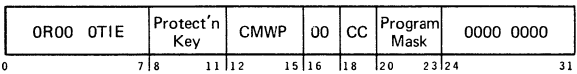
PROGRAM STATUS WORD (BC Mode)

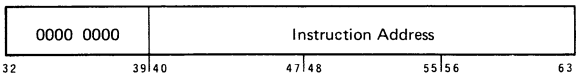




0-5 Channel 0 to 5 masks
6 Mask for channel 6 and up
7 (E) External mask
12 (C=0) Basic control mode
13 (M) Machine-check mask
14 (W=1) Wait state
15 (P=1) Problem state
32-33 (ILC) Instruction length code
34-35 (CC) Condition code
36 Fixed-point overflow mask
37 Decimal overflow mask
38 Exponent underflow mask
39 Significance mask

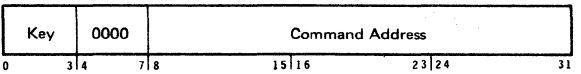
PROGRAM STATUS WORD (EC Mode)



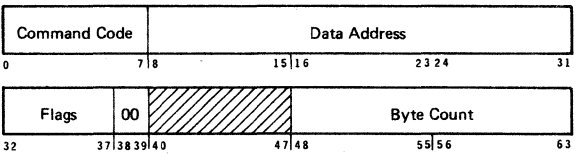


1 (R) Program event recording mask
5 (T=1) Translation mode
6 (I) Input/output mask
7 (E) External mask
12 (C=1) Extended control mode
13 (M) Machine-check mask
14 (W=1) Wait state
15 (P=1) Problem state
18-19 (CC) Condition code
20 Fixed-point overflow mask
21 Decimal overflow mask
22 Exponent underflow mask
23 Significance mask

CHANNEL ADDRESS WORD (hex 48)

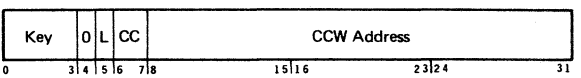


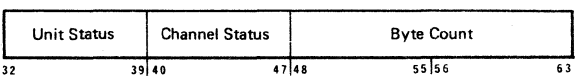
CHANNEL COMMAND WORD



CD—bit 32 (80) causes use of address portion of next CCW.
CC—bit 33 (40) causes use of command code and data address of next CCW.
SLI—bit 34 (20) causes suppression of possible incorrect length indication.
Skip—bit 35 (10) suppresses transfer of information to main storage.
PCI—bit 36 (08) causes a channel program controlled interruption.
IDA—bit 37 (04) causes bits 8-31 of CCW to specify location of first IDAW.

CHANNEL STATUS WORD (hex 40)





5 Logout pending
6-7 Deferred condition code
32 (8000) Attention
33 (4000) Status modifier
34 (2000) Control unit end
35 (1000) Busy
36 (0800) Channel end
37 (0400) Device end
38 (0200) Unit check
39 (0100) Unit exception
40 (0080) Program control interruption
41 (0040) Incorrect length
42 (0020) Program check
43 (0010) Protection check
44 (0008) Channel data check
45 (0004) Channel control check
46 (0002) Interface control check
47 (0001) Chaining check
48-63 Residual byte count for the last CCW used

Limited Channel Logout - Machine-check Interruption Code

LIMITED CHANNEL LOGOUT (hex B0)

0	SCU id	Detect	Source	000	Field validity flags	TT	00	A	Seq.
0 1 3	4	7	8 12	13 15	16 23	24 26		28	29 31

Detect field
- 4 CPU
- 5 Channel
- 6 Storage control unit
- 7 Storage unit

Source field
- 8 CPU
- 9 Channel
- 10 Storage control unit
- 11 Storage unit
- 12 Control unit

16-23 Field validity flags
- 16 Interface address
- 17-18 Reserved (00)
- 19 Sequence code
- 20 Unit status
- 21 Command address and key
- 22 Channel address
- 23 Device address

24-25 (TT) Type of termination
Code 00 Interface disconnect
01 Stop, stack, or normal
10 Selective reset
11 System reset

28 (A) I/O error alert

29-31 Sequence code

Machine - check Interruption Code

MACHINE-CHECK INTERRUPTION CODE (hex E8)

MC conditions	000	00	Time	Stg. error	0	Validity indicators
0 8	9	13	14	16 18	19	20 31

0000	0000	0000	00	Val.	MCEL length
32	39	40	45	46 48	55 56 63

0 System damage
1 Instr. proc'g damage
2 System recovery
3 Timer damage
4 Timing facil. damage
5 External damage
6 Not assigned (0)
7 Degradation
8 Warning
14 Backed-up
15 Delayed
16 Uncorrected
17 Corrected
18 Key uncorrected
20 PSW bits 12-15
21 PSW masks and key
22 Prog. mask and CC
23 Instruction address
24 Failing stg. address
25 Region code
27 Floating-pt registers
28 General registers
29 Control registers
30 CPU ext'd logout
31 Storage logical
46 CPU timer
47 Clock comparator

I/O Command Codes

Standard Command Code Assignments (CCW bits 0-7)

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

CONSOLE PRINTERS

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

3504, 3505 CARD READER/3525 CARD PUNCH (GA21-9124)

Command	Binary	Hex
Sense	0000 0100	04
Feed, Select Stacker	SS10 F011	
Read Only*	11D0 F010	
Diagnostic Read	1101 0010	D2
Read, Feed, Select Stacker*	SSD0 F010	
Write RCE Format*	0001 0001	11
3504, 3505 only		
Write OMR Format†	0011 0001	31
3525 only		
Write, Feed, Select Stacker	SSD0 0001	
Print Line*	LLLL L101	

Bit Meanings

SS	Stacker
00	1
01	2
10	2
F	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

PRINTERS: 3211/3811 (GA24-3543), 3203/IPA, 1403/2821* (GA24-3312)

	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 without folding	FB
Fold†	43
Unfold†	23
Load UCSB and Fold (exc. 3211)	F3
UCS Gate Load (1403 only)	EB
Load FCB†	63
Block Data Check	73
Allow Data Check	7B
Read PLB†	02
Read UCSB†	0A
Read FCB†	12
Diag. Check Read (exc. 3203)	06
Diagnostic Write†	05
Raise Cover†	6B
Diagnostic Gate†	07
Diagnostic Read (1403 only)	02

*1403/IPA diagnostics are model-dependent; UCS special feature on 1403 †3211 only

I/O Command Codes (cont'd)

3420/3803, 3410/3411 MAGNETIC TAPE
See GA32-0020, -0021, -0022 for function of specific models and special features required.

		Density	Parity	DC	Trans	Cmd
Write	01	Mode Set 1 (7-track)				
Read Forward	02		odd	on	off	13
Read Backward	0C			off	off	33
Sense	04	200			on	3B
Sense Reserve*†	F4		even	off	off	23
Sense Release*†	D4				on	2B
Request Track-in-Error	1B			on	off	53
Loop Write-to-Read†	8B		odd	off	off	73
See Diagnose†	4B	556			on	7B
Rewind	07		even	off	off	63
Rewind Unload	0F				on	6B
Erase Gap	17			on	off	93
Write Tape Mark	1F		odd	off	off	B3
Backspace Block	27	800			on	BB
Backspace File	2F		even	off	off	A3
Forward Space Block	37				on	AB
Forward Space File	3F	Mode Set 2 (9-track)				
Data Security Erase†	97	800 bpi				CB
Diagnostic Mode Set†	0B	1600 bpi				C3
		6250 bpi†				D3

*Two-channel switch required †3420 only

DIRECT ACCESS STORAGE DEVICES:
3330-3340 SERIES (GA26-1592, -1617, -1619, -1620);
3350 (GA26-1638);
2305/2835 (GA26-1589); 2314, 2319(GA26-3599, -1606)

Command		MT Off	MT On*	Count
Control	Orient (c)	2B		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, f)	23		1
	Restore (executes as a no-op)	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 (including mask bytes) in search argument
	Key and Data High (d)	4D	CD	
	Key and Data Eq. or Hi (d)	6D	ED	
Continue Scan	Search Equal (d)	25	A5	
	Search High (d)	45	C5	
	Search High or Equal (d)	65	E5	
	Set Compare (d)	35	B5	
	Set Compare (d)	75	F5	
	No Compare (d)	55	D5	

* Code same as MT Off except as listed.
a. Except 2314, 2319
b. 3330-3340 Series only; manual reset on 3340.
c. 2304/2835 only.
d. 2314, 2319 only.
e. String switch or 2-channel switch feature required; standard on 2314 and 2844.
f. Special feature required on 3340.

I/O Command Codes (cont'd)

DIRECT ACCESS STORAGE DEVICES: (cont'd)
3330-3340 SERIES (GA26-1592, -1617, -1619, 1620);
3350 (GA26-1638)
2305/2835 (GA26-1589); 2314, 2319 (GA26-3599, -1606)

Command		MT Off	MT On*	Count
Read	Home Address	1A	9A	5
	Count	12	92	8
	Record 0	16	96	
	Data	06	86	Number of bytes to be transferred
	Key and Data	0E	8E	
	Count, Key and Data	1E	9E	
	IPL	02		
	Sector (a, f)	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 (exc. 7 on 3340)
	Record 0	15		8+KL+DL of 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. Except 2314, 2319.
b. 3330-3340, 3350 Series only; manual reset on 3340/3344.
c. 2304/2835 only.
d. 2314, 2319 only.
e. String switch or 2-channel switch feature required; standard on 2314 and 2844.
f. Special feature required on 3340.

3540 Diskette I/O Unit

Command	Binary	Hex
Seek	0000 0111	07
Read IPL	0000 0010	02
Feed	0001 0111	17
Write Control	0010 0001	21
Define Operations	0010 1111	2F
Read Data	0000 0110	06
Sense	0000 0100	04
Test I/O	0000 0000	00
No-Op	0000 0011	03
Write Data	0000 0101	05

Channel Command Code - ANSI Control Characters

Device	Command for CCW	8-Bit Code 0	1	2	3	4	5	6	7	Hex	Dec
1052	Read Inquiry BCD	0	0	0	0	1	0	1	0	0A	10
	Read Reader 2 BCD	0	0	0	0	0	0	1	0	02	02
	Write BDC, Auto Carriage Return	0	0	0	0	1	0	0	1	09	09
	Write BDC, No Auto Carriage Retum	0	0	0	0	0	0	0	1	01	01
	No Op	0	0	0	0	0	0	1	1	03	03
	Sense	0	0	0	0	0	1	0	0	04	04
	Alarm	0	0	0	0	1	0	1	1	0B	11
2540	Read, Feed, Select Stacker SS — Type AA	S	S	D	0	0	0	1	0		
	Read — Type AB	1	1	D	0	0	0	1	0		
	Read, Feed (1400 compatability mode only)	1	1	D	1	0	0	1	0		
	Feed, Select Stacker SS — Type BA	S	S	1	0	0	0	1	1		
	PFR Punch, Feed, Select Stacker SS — Type BA	S	S	D	0	1	0	0	1		
	Punch, Feed, Select Stacker SS — Type BB	S	S	D	0	0	0	0	1		
2400 Tape*	Read Backward (Overrides Data Converter On)	0	0	0	0	1	1	0	0	0C	12
	Sense	0	0	0	0	0	1	0	0	04	04
	Write	0	0	0	0	0	0	0	1	01	01
	Read	0	0	0	0	0	0	1	0	02	02
	Control	0	0	C	C	C	1	1	1		
		D	D	M	M	M	0	1	1		
		1	1	N	N	N	0	1	1		

SS	Stacker
00	R1
01	R2
10	RP3

D	Data Mode
0	EBCDIC
1	Column Binary

N	N	N	
0	0	0	1600 bpi P.E.**
0	0	1	800 bpi NRZ1

C	C	C	Control Codes	Hex	Dec
0	0	0	REW	7	7
0	0	1	RUN	0F	15
0	1	0	ERG	17	23
0	1	1	WTM	1F	31
1	0	0	BSR	27	39
1	0	1	BSF	2F	47
1	1	0	FSR	37	55
1	1	1	FSF	3F	63

D	D	7 Track Density
0	0	200
0	1	556
1	0	800**
1	1	***

7 Track

M	M	M	(Mode Modifiers)	Set Density	Set Odd Parity	Set Even Parity	Data Converter On	Data Converter Off	Translator On	Translator Off	Request TIE (Track in Error)
0	0	0	No Op								
0	0	1	Not Used								
0	1	0	Reset Condition	X	X		X			X	
0	1	1	Nine-track only								X
1	0	0		X		X		X		X	
1	0	1		X		X		X	X		
1	1	0	Reset Condition	X	X			X		X	
1	1	1		X	X			X	X		

* 9 track op. forces 800 BPI and odd parity; also, it overrides 7 track but does not reset 7 track. Load/Sys Reset forces 7 track to 800 BPI, odd parity, data converter on, translator off.

** Reset condition

*** Set 9 Track mode, Models 4-6

ANSI Control Characters

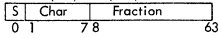
Code	Action Before Printing a Line
b	Space one line (blank code)
0	Space two lines
-	Space three lines
+	Suppress space
1	Skip to channel 1
2	Skip to channel 2
3	Skip to channel 3
4	Skip to channel 4
5	Skip to channel 5
6	Skip to channel 6
7	Skip to channel 7
8	Skip to channel 8
9	Skip to channel 9
A	Skip to channel 10
B	Skip to channel 11
C	Skip to channel 12

Code	Action After Punching a Card
V	Select punch pocket 1
W	Select punch pocket 2

Channel Command Codes (cont'd)

Device	Type Command	Command Name	Hex Code
3800 Printing Subsystem	Load	Load Forms Control Buffer	63
		Load Translate Table	83
		Load Character Module (WCGM)	53
		Load Copy Number	23
		Load Forms Overlay Control	43
		Load Graphic Character Modification	25
		Load Copy Modification	35
	Write	Write with No Space	01
		Write and Space 1 Line	09
		Write and Space 2 Lines	11
		Write and Space 3 Lines	19
		Write and Skip to Channel 1	89
		Write and Skip to Channel 2	91
		Write and Skip to Channel 3	99
		Write and Skip to Channel 4	A1
		Write and Skip to Channel 5	A9
		Write and Skip to Channel 6	B1
		Write and Skip to Channel 7	B9
		Write and Skip to Channel 8	C1
		Write and Skip to Channel 9	C9
		Write and Skip to Channel 10	D1
		Write and Skip to Channel 11	D9
		Write and Skip to Channel 12	E1
	Forms	Space 1 Line Immediately	0B
		Space 2 Lines Immediately	13
		Space 3 Lines Immediately	1B
		Skip to Channel 1 Immediately	8B
		Skip to Channel 2 Immediately	93
		Skip to Channel 3 Immediately	9B
		Skip to Channel 4 Immediately	A3
		Skip to Channel 5 Immediately	AB
		Skip to Channel 6 Immediately	B3
		Skip to Channel 7 Immediately	BB
		Skip to Channel 8 Immediately	C3
		Skip to Channel 9 Immediately	CB
		Skip to Channel 10 Immediately	D3
		Skip to Channel 11 Immediately	DB
		Skip to Channel 12 Immediately	E3
	Sense	Test I/O	00
		Sense I/O	04
		Sense Intermediate Buffer	14
		Sense Error Log	24
	Control	No Operation	03
		Block Data Check	73
		Allow Data Check	7B
		Initialize Printer	37
		Select Translate Table 0	47
		Select Translate Table 1	57
		Select Translate Table 2	67
		Select Translate Table 3	77
		Clear Printer	87
		End of Transmission	07
		Mark Form (End of Job)	17

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Add	A	5A	RX	R1, D2(X2,B2)	Add opr 2 to opr 1 (Sto) (Reg)	Addr Specif Fxpt Oflo	0 Sum = 0 1 Sum < 0 2 Sum > 0 3 Overflow
Add	AR	1A	RR	R1, R2	Add opr 2 to opr 1 (GPR) (Reg)	Fxpt Oflo	0 Sum = 0 1 Sum < 0 2 Sum > 0 3 Overflow
Add Decimal	AP	FA	SS	D1(L1,B1), D2(L2,B2)	Add dec opr 2 to opr 1 (Sto) (Sto) (Right to left byte by byte). (Opr 1 and 2 must be in packed) (Fields can overlap if low-order bytes coincide) (If opr 1 and opr 2 refer to same field, the field is doubled)	Addr Data Dec Oflo Protect Opera	0 Sum = 0 1 Sum < 0 2 Sum > 0 3 Overflow
Add Halfword	AH	4A	RX	R1, D2(X2,B2)	Add opr 2 to opr 1 (Sto) (Reg) (High-order 16 bits expanded) opr 2	Addr Fxpt Oflo Specif	0 Sum = 0 1 Sum < 0 2 Sum > 0 3 Overflow
Add Logical	AL	5E	RX	R1, D2(X2,B2)	Add log opr 2 to opr 1 (Sto) (Reg)	Addr Specif	0 Sum = 0 1 Sum ≠ 0 2 Sum = 0 3 Sum ≠ 0
Add Logical	ALR	1E	RR	R1, R2	Add log opr 2 to opr 1 (Reg) (Reg)	None	0 Sum = 0 1 Sum ≠ 0 2 Sum = 0 3 Sum ≠ 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Add Normalized (Extended)	AXR	36	RR	R1, R2	FP Add opr 2 to opr 1 (FPR pair) (FPR pair) Extended sum is put in opr 1 (FPR pair) Each operand consists of two FPR Only FPR 0 and FPR 4 may be specified for opr 1 or opr 2.	Specif Exp Oflo Exp Uflo Signif Opera	0 Fract = 0 1 Result < 0 2 Result > 0
Add Normalized (Long)	AD	6A	RX	R1, D2(X2, B2)	FP Add opr 2 to opr 1 (Sto) (FPR)  	Addr Specif Signif Exp Oflo Exp Uflo Opera	0 Fract = 0 1 Result < 0 2 Result > 0
Add Normalized (Long)	ADR	2A	RR	R1, R2	FP Add opr 2 to opr 1 (FPR) (FPR)	Specif Opera Signif Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0
Add Normalized (Short)	AE	7A	RX	R1, D2(X2, B2)	FP Add opr 2 to opr 1 (Sto) (FPR) (Low-order halves of FPR ignored and unchanged)	Addr Specif Signif Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0
Add Normalized (Short)	AER	3A	RR	R1, R2	FP Add opr 2 to opr 1 (FPR) (FPR) (Low-order halves of FPR ignored and unchanged)	Specif Signif Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Add Unnormalized (Long)	AW	6E	RX	R1, D2(X2,B2)	FP Add opr 2 to opr 1 (Sto) (FPR)	Addr Specif Signif Exp Oflo Opera	0 Fract = 0 1 Result < 0 2 Result > 0
Add Unnormalized (Long)	AWR	2E	RR	R1,R2	FP Add opr 2 to opr 1 (FPR) (FPR)	Specif Signif Exp Oflo Opera	0 Fract = 0 1 Result < 0 2 Result > 0
Add Unnormalized (Short)	AU	7E	RX	R1, D2(X2,B2)	FP Add opr 2 to opr 1 (Sto) (FPR) (Low-order halves of FPR ignored and unchanged)	Addr Specif Signif Exp Oflo Opera	0 Fract = 0 1 Result < 0 2 Result > 0
Add Unnormalized (Short)	AUR	3E	RR	R1, R2	FP Add opr 2 to opr 1 (FPR) (FPR) (Low-order halves of FPR ignored and unchanged)	Specif Signif Exp Oflo Opera	0 Fract = 0 1 Result < 0 2 Result > 0
AND	N	54	RX	R1, D2(X2,B2)	Place the product of both opr's into opr 1	Addr Specif	0 Result = 0 1 Result ≠ 0
AND	NC	D4	SS	D1(L,B1), D2(B2)	Place the product of both opr's into opr 1 (Left to right byte by byte) (Max number of bytes ANDed: 256)	Addr Protect	0 Result = 0 1 Result ≠ 0
AND	NR	14	RR	R1, R2	Place the product of both opr's into opr 1	None	0 Result = 0 1 Result ≠ 0
AND	NI	94	SI	D1(B1), I2	AND the 1 byte from the instruction stream (8-15) to opr 1	Addr Protect	0 Result = 0 1 Result ≠ 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Branch and Link	BAL	45	RX	R1, D2(X2, B2)	Store ILC, CC prog mask, and 24 bits of inst adr in opr 1. Branch to adr of opr 2	None	Unchanged
Branch and Link	BALR	05	RR	R1, R2	Store ILC, CC prog mask, and 24 bits of inst adr in opr 1. Branch to adr of opr 2 (If opr 2 = 0, store, no branch)	None	Unchanged
Branch on Condition	BC	47	RX	M1, D2(X2, B2)	Compare opr 1 with cond code (Mask) 8-11 (Mask = 7) Branch on non-zero cond code (Mask = 15) Uncond branch (Mask = 8) Cond code 00 (Mask = 4) Cond code 01 (Mask = 2) Cond code 10 (Mask = 1) Cond code 11 (NOP if cond not met)	None	Unchanged
Branch on Condition	BCR	07	RR	M1, R2	Compare opr 1 with cond code Branch to opr 2 adr if cond met (If opr 2 = 0) NOP	None	Unchanged
Branch on Count	BCT	46	RX	R1, D2(X2, B2)	Reduce opr 1 by 1 and branch to opr 2 adr (If opr 1 = 1) Reduce, no branch	None	Unchanged
Branch on Count	BCTR	06	RR	R1, R2	Reduce opr 1 by 1 and branch to opr 2 adr (If opr 1 = 1) Reduce, no branch (If opr 2 = 0) Reduce, no branch	None	Unchanged
Branch on Equal	BE	47(BC 8)	RX, Ext.	D2(X2, B2)	Branch if mask = cond code	None	Unchanged
Branch on Equal	BER	07(BCR 8)	RR, Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on High	BH	47(BC 2)	RX, Ext.	D2(X2, B2)	Branch if mask = cond code	None	Unchanged
Branch on High	BHR	07(BCR 2)	RR, Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Index High	BXH	86	RS	R1, R3, D2(B2)	Add opr 3 to opr 1 Sum compared to opr 3 if opr 3 adr is odd Sum compared to opr 3 + 1 if opr 3 addr is even Branch to opr 2 adr if sum > comparand	None	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Branch on Index Low or Equal	BXLE	87	RS	R1, R3, D2(B2)	Same as Branch On Index High Branch to opr 2 adr if sum <or = opr 3+1	None	Unchanged
Branch on Low	BL	47(BC 4)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Low	BLR	07(BCR4)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch if Mixed	BM	47(BC 4)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch if Mixed	BMR	07(BCR 4)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Minus	BM	47(BC 4)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Minus	BMR	07(BCR 4)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not Equal	BNE	47(BC 7)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not Equal	BNER	07(BCR 7)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not High	BNH	47(BC 13)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not High	BNHR	07(BCR 13)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not Low	BNL	47(BC 11)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not Low	BNLR	07(BCR 11)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not Minus	BNM	47(BC 11)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not Minus	BNMR	07(BCR 11)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not Ones	BNO	47(BC 14)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not Ones	BNOR	07(BCR 14)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not Plus	BNP	47(BC 13)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not Plus	BNPR	07(BCR 13)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Not Zeros	BNZ	47(BC 7)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Not Zeros	BNZR	07(BCR 7)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged

Ext = Extended Mnemonic

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Branch if Ones	BO	47(BC 1)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch if Ones	BOR	07(BCR 1)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Overflow	BO	47(BC 1)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Overflow	BOR	07(BCR 1)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Plus	BP	47(BC 2)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Plus	BPR	07(BCR 2)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch if Zeros	BZ	47(BC 8)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch if Zeros	BZR	07(BCR8)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch on Zero	BZ	47(BC 8)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch on Zero	BZR	07(BCR 8)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Branch Unconditional	B	47(BC 15)	RX,Ext.	D2 (X2,B2)	Branch if mask = cond code	None	Unchanged
Branch Unconditional	BR	07(BC 15)	RR,Ext.	R2	Branch if mask = cond code	None	Unchanged
Clear I/O	CLRIO	9D01	S	D2 (B2)	Terminate execution of current I/O op at addressed dev.	Priv	0 opr's = 1 CSW stored 2 channel or subchannel busy 3 not oprtnal
Compare	C	59	RX	R1, D2(X2,B2)	Compare opr 1 algebraically to opr 2 (Reg)	Addr Specif	0 opr's = 1 1st < 2 1st >
Compare	CR	19	RR	R1, R2	Compare opr 1 algebraically to opr 2	None	0 opr's = 1 1st < 2 1st >

Ext = Extended Mnemonic

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Compare and Swap	CS	BA	RS	R1,R3,D2(B2)	Compare opr 1 to opr 2. Store opr 3 in opr 2 if =, store opr 2 in opr 1 if ≠.	Addr Specif Protect Opera	0 opr's = 1 1st = 2nd; 2nd replaced by 3rd
Compare Decimal	CP	F9	SS	D1 (L1,B1), D2(L2,B2)	Compare opr 1 to opr 2 (binary right to left) byte by byte (Opr's must be packed) (Fields can overlap if low-order bytes coincide) (The shorter opr is extended with high-order zeros)	Addr Data Opera	0 opr's = 1 1st < 2 1st >
Compare Double and Swap	CDS	BB	RS	R1, R3, D2(B2)	Compare opr 1 to opr 2. Store opr 3 in opr 2 if =, store opr 2 in opr 1 if ≠.	Addr Specif Protect Opera	0 opr's = 1 1st = 2nd 2nd replaced by 3rd
Compare Halfword	CH	49	RX	R1, D2(X2,B2)	Compare opr 1 algebraically to opr 2 (Hi-order 16 bits expanded) opr 2	Addr Specif	0 opr's = 1 1st < 2 1st >
Compare Logical	CL	55	RX	R1, D2(X2,B2)	Compare opr 1 to opr 2 (binary left to right) (Terminates if/when ≠ found)	Addr Specif	0 opr's = 1 1st < 2 1st >
Compare Logical	CLC	D5	SS	D1 (L,B1), D2(B2)	Compare opr 1 to opr 2 (binary left to right) (Terminated if/when ≠ found) (opr length max 256 bytes)	Addr Specif	0 opr's = 1 1st < 2 1st >
Compare Logical Immediate	CLI	95	SI	D1 (B1), I2	Compare opr 1 to opr 2 (Imm) (Sto) (binary left to right) (Terminates if/when ≠ found)	Addr	0 opr's = 1 1st < 2 1st >
Compare Logical	CLR	15	RR	R1, R2	Compare opr 1 to opr 2 (binary left to right) (Terminates if/when = found)	Addr	0 opr's = 1 1st < 2 1st >
Compare Logical Characters Under Mask	CLM	BD	RS	R1, M3,D2(B2)	Compare opr 2 to opr 1 under control of mask (binary left to right)	Addr Protect Opera	0 Selected by bytes = or mask = 0 1 Selected field 1st opr is low 2 Selected Field 1st opr is high

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Compare Logical Long	CLCL	0F	RR	R1, R2	Compare opr 1 to opr 2 (opr 1 and 2 indicate even/odd reg. pair)	Addr Specif Opera Protect	0 opr's = 1 1st < 2 1st > 3 --
Compare (Long)	CD	69	RX	R1, D2(X2, B2)	Compare opr 1 algebraically to opr 2 (Equalize and subtract)	Addr Specif Opera	0 opr's = 1 1st < 2 1st >
Compare (Long)	CDR	29	RR	R1, R2	Compare opr 1 algebraically to opr 2 (FPR) (Equalize and subtract)	Specif Addr Opera	0 opr's = 1 1st < 2 1st >
Compare (Short)	CE	79	RX	R1, D2(X2, B2)	Compare opr 1 algebraically to opr 2 (FPR) (Sto) (Low-order halves of FPR ignored and unchanged)	Addr Specif Opera	0 opr's = 1 1st < 2 1st >
Compare (Short)	CER	39	RR	R1, R2	Compare opr 1 algebraically to opr 2 (FPR) (FPR) (Low-order halves of FPR ignored and unchanged)	Specif Opera	0 opr's = 1 1st < 2 1st >
Convert to Binary	CVB	4F	RX	R1, D2(X2, B2)	Convert opr 2 (packed decimal) (Doubleword bounds) to binary and put in opr 1 location	Addr Specif Data Fxpt Div	Unchanged
Convert to Decimal	CVD	4E	RX	R1, D2(X2, B2)	Convert opr 1 (binary) to packed decimal (doubleword bounds) and put in opr 2	Addr Specif Protect	Unchanged
Diagnose	----	83		See IBM System/370 Principles of Operation, GA22-7000	See IBM System/370 Principles of Operation, GA22-7000	Priv Oper Model dependent	Unpredictable

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Divide	D	5D	RX	R1, D2 (X2,B2)	Divide opr 1 by opr 2 (even and odd regs) (Sto) Opr 1 becomes remainder and quotient	Addr Specif Fxpt Div	Unchanged
Divide	DR	1D	RR	R1, R2	Divide opr 1 by opr 2 Dividend: even and odd pair regs Opr 1 becomes remainder and quotient (full word only)	Specif Fxpt Div	Unchanged
Divide Decimal	DP	FD	SS	D1(L1,B1), D2(L2,B2)	Divide opr 1 by opr 2 Opr 1 becomes quotient and remainder (left justified) Dividend: at least 1 leading zero, max size 31 digits and sign Divisor: max size 15 digits and sign, numerically larger than dividend Both opr's packed format Remainder size = divisor size (Fields can overlap if low-order bytes coincide.)	Addr Protect Specif Data Dec Div Opera	Unchanged
Divide (Long)	DD	6D	RX	R1, D2(X2,B2)	FP Divide opr 1 by opr 2 (FPR) (Sto) Opr 1 becomes quotient (prenormalized)	Addr Specif Exp Oflo FP Div Opera Exp Uflo	Unchanged
Divide (Long)	DDR	2D	RR	R1, R2	FP Divide opr 1 by opr 2 Prenormalize (FPR) (FPR) (Dividend) (Divisor) Oper 1 becomes quotient	Specif Opera Exp Oflo Exp Uflo FP Div	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Divide (Short)	DE	7D	RX	R1, D2(X2,B2)	FP Divide opr 1 by opr 2 Prenormalize (Dividend) (Divisor) Opr 1 becomes quotient (Low-order halves of FPR ignored and unchanged)	Addr Specif Exp Oflo Exp Uflo FP Div Opera	Unchanged
Divide (Short)	DER	3D	RR	R1, R2	FP Divide opr 1 by 2 Prenormalize (FPR) (FPR) (Dividend) (Divisor) Opr 1 becomes quotient (Low-order halves of FPR ignored and unchanged)	Specif Exp Oflo FP Div Exp Uflo Opera	Unchanged
Edit	ED	DE	SS	D1(L,B1), D2(B2)	Opr 1 = pattern, opr 2 = source Opr 2 is changed from packed to zoned and edited under control of opr 1. Opr's processed left to right (Fill char is 1st char in pattern field unless it is a digit/select/significance-start char.) (Opr 1 terminates operation) See IBM System/370 Principles of Operation, GA22-7000	Addr Data Opera Protect	Source 0 field = 0 1 field < 0 2 field > 0
Edit and Mark	EDMK	DF	SS	D1(L,B1), D2(B2)	Same as Edit (Adr of 1st significant result digit recorded in GPR 1)	Opera Addr Data Protect	Source 0 field = 0 1 field < 0 2 field > 0
Exclusive OR	X	57	RX	R1, D2(X2,B2)	Exclusive-OR opr 2 and opr 1 and the modulo-two sum placed in opr 1	Addr Specif	0 Result = 0 1 Result ≠ 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Exclusive OR	XC	D7	SS	D1(L,B1), D2(B2)	Exclusive-OR opr 2 and opr 1 and modulo-two sum placed in opr 1.	Addr Protect	0 Result = 0 1 Result ≠ 0
Exclusive OR	XR	17	RR	R1, R2	Exclusive-OR opr 2 and opr 1 and modulo-two sum placed in opr 1.		0 Result = 0 1 Result ≠ 0
Exclusive OR Immediate	XI	97	SI	D1(B1), I2	Exclusive-OR opr 2 and opr 1 and modulo-two sum placed in opr 1.	Addr Protect	0 Result = 0 1 Result ≠ 1
Execute	EX	44	RX	R1, D2(X2,B2)	The instruction addressed by opr 2 is modified by opr 1 and executed.	Addr Exec Specif	May be set by this instruction
Halve, Long	HDR	24	RR	R1, R2	Opr 2 is divided by 2 and placed in opr 1.	Specif Opera	Unchanged
Halve, Short	HER	34	RR	R1, R2	Opr 2 is divided by 2 and placed in opr 1.	Specif Opera	Unchanged
Halt Device	HDV	9E01	S	D1(B1)	Execution of current I/O op at addressed dev is terminated (full op cd - 1001 1110 xxxx xxx1).	Priv	0 Subchan busy with another dev or int pending 1 CSW stored 2 Chan working with another device
Halt I/O	HIO	9E00	S	D1(B1)	Execution of current I/O op at addresses dev, subchan, and chan term (full op cd - 1001 1110 xxxx xxx0).	Priv	0 Chan or sub-chan not working 1 CSW stored 2 Burst oper terminated 3 Not operational
Insert Character	IC	43	RX	R1, D2(X2,B2)	Byte at opr 2 is inserted in bits 24-31 of reg at opr 1.	Addr	Unchanged
Insert Characters Under Mask	ICM	BF	RS	R1, M3, D2(B2)	1 to 4 bytes at opr 2 are inserted in reg at opr 1 under control of mask.	Addr Protect Opera	0 Selected bits or mask = 0 1 Leftmost bit of spec byte = 1 2 Leftmost bit of spec byte = 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Insert PSW Key	IPK	B208	S		Protection key of current PSW inserted into reg 2 bit pos 24-27. Bits 28-31 set to 0.	Priv	Unchanged
Insert Storage Key	ISK	09	RR	R1, R2	Opr 2, 8-20 fetches 7-bit sto key byte. 7-bit sto key is placed in opr 1, 24-30. Bits 0-23 unchanged, 31 set to zero. (opr 2, 0-7 and 21-27 ignored, 28-31 must = 0)	Priv Addr Specif Opera	Unchanged
Load	L	58	RX	R1, D2(X2,B2)	Load opr 2 into opr 1.	Addr Specif	Unchanged
Load	LR	18	RR	R1, R2	Opr 2 into opr 1.	None	Unchanged
Load Address	LA	41	RX	R1, D2(X2,B2)	Opr 2, 12-31 to opr 1, 8-31. Opr 1, 0-7 set to zero (no storage reference made)	None	Unchanged
Load and Test	LTR	12	RR	R1, R2	Opr 2 into opr 1 (When opr 1 and opr 2 specify same reg result is test without data transfer.)	None	0 Result = 0 1 Result < 0 2 Result > 0
Load and Test (Long)	LTDR	22	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR) (When opr 1 and opr 2 specify same reg result is test without data transfer.)	Specif Opera	0 Result fraction = 0 1 Result < 0 2 Result > 0
Load and Test (Short)	LTER	32	RR	R1, R2	Opr 2 inio opr 1 (FPR) (FPR) (Low-order half of opr 1 unchanged) (When opr 1 and opr 2 specify same reg result is test without data transfer.)	Specif Opera	0 Result Fraction = 0 1 Result < 0 2 Result > 0
Load Complement	LCR	13	RR	R1, R2	2's complement of opr 2 into opr 1 (overflow when max negative number is complemented)	Fxpt Oflo	0 Result = Expt Uflo 1 Result < 0 2 Result > 0 3 Overflow

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Load Complement (Short)	LCER	33	RR	R1, R2	Opr 2 into opr i (FPR) (FPR) (Opr 1 sign inverted, low-order half unchanged) (Opr 2 unchanged)	Specif Opera	0 Result Fract = 0 1 Result < 0 2 Result > 0
Load Complement (Long)	LCDR	23	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR) (Opr 1 sign inverted, low-order half unchanged) (Opr 2 unchanged) (Low-order half of opr 1 unchanged)	Specif Opera	0 Result Fract = 0 1 Result < 0 2 Result > 0
Load Control	LCTL	B7	RS	R1, R3, D2(B2)	Cntl regs from opr 1 to opr 3 loaded with info starting at opr 2.	Addr Specif Priv Protect Opera	Unchanged
Load Halfword	LH	48	RX	R1, D2(X2,B2)	Opr 2 halfword expanded to fullword with sign bits, placed in opr 1 (High-order expanded)	Addr Specif	Unchanged
Load (Long)	LD	68	RX	R1, D2(X2,B2)	Opr 2 into opr 1 (Sto) (FPR)	Addr Specif Opera	Unchanged
Load (Long)	LDR	28	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR)	Specif Opera	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Load Multiple	LM	98	RS	R1, R3, D2(B2)	Opr 2 into GPRs in ascending order Starting reg specified by R1, ending reg specified by R3 (Reg wrap-around possible)	Addr Specif	Unchanged
Load Negative	LNR	11	RR	R1, R2	2's complement of opr 2 into opr 1 (Reg) (Reg) (If opr 2 contains a (-) number or zero, the number is unchanged)	None	0 Result = 0 1 Result < 0
Load Negative (Long)	LNDR	21	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR) Opr 1 sign bit is 1 (negative) Opr 2 unchanged	Specif Opera	0 Result Fract = 0 1 Result < 0
Load Negative (Short)	LNER	31	RR	R1, R2	Opr 2 into opr 1 Opr 1 sign bit is 1 (negative) Opr 2 unchanged (Low-order half of opr 1 unchanged)	Specif Opera	0 Result Fract = 0 1 Result < 0
Load Positive	LPR	10	RR	R1, R2	Opr 2 into opr 1 (Negative numbers are complemented) (Overflow occurs when the max negative number is complemented)	Fxpt Oflo	0 Result = 0 2 Result > 0 3 Overflow
Load Positive (Long)	LPDR	20	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR) Opr 1 sign bit made a zero (positive) Opr 2 unchanged	Specif Opera	0 Result Fract = 0 1 Result < 0 2 Result > 0
Load Positive (Short)	LPER	30	RR	R1, R2	Opr 2 into opr 1 Opr 1 sign bit made a zero (positive) Opr 2 unchanged (Low-order half of opr 1 unchanged)	Specif Opera	0 Result Fract = 0 1 Result < 0 2 Result > 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Load PSW	LPSW	82	SI	D1 (B1)	Opr 1 into PSW (Opr 1 low-order 3 bit adr must = 0) (Instruction used to enter the problem or wait state)	Priv Addr Specif	Set according to new PSW bits 34 and 35
Load (Short)	LE	78	RX	R1, D2(X2,B2)	Opr 2 into opr 1 (Sto) (FPR) (Low-order half of opr 1 unchanged)	Addr Specif Opera	Unchanged
Load (Short)	LER	38	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR) (Low-order half of opr 1 unchanged)	Specif Opera	Unchanged
Load Real Address	LRA	B1	RX	R1, D2(X2,B2)	Real adr corresponding to opr 2 logical adr placed in opr 1.	Priv Addr Specif Opera	0 Translation available 1 Seg tbl entry invalid 2 Page tbl entry invalid 3 Seg or page tbl length violation
Load Rounded (Extended to Long)	LRDR	25	RR	R1, R2	Opr 2 is rounded from extended to long format and put in opr 1 (FPR pair) (FPR) Only FPR 0 and FPR 4 may be specified for opr 2.	Specif Exp Oflo Opera	Unchanged
Load Rounded (Long to Short)	LRER	35	RR	R1, R2	Opr 2 is rounded from long to short format and put into opr 1 (FPR) (FPR) Add an absolute 1 to opr 2, bit 32; carry will ripple left. Lower half of result FPR will remain unchanged.	Specif Exp Oflo Opera	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Monitor Call	MC	AF	SI	D1 (B1),I2	Causes program interrupt if monitor-mask bit in cont. reg 8 = appropriate monitor class specified in positions 12-15 of I2. Real storage locations 148 and 156 will zero, loc 149=I2, and loc. 157-159=D1 + contents to B1.	Monitor Specif	Unchanged
Move Characters	MVC	D2	SS	D1(L,B1),D2(B2)	Opr 2 to opr 1 (Left to right byte by byte) (Max number of bytes moved: 256) (No restriction on overlapping fields)	Addr Protect	Unchanged
Move Immediate	MVI	92	SI	D1(B1),I2	Move the 1 byte from the instruction stream (8-15) to opr 1.	Addr Protect	Unchanged
Move Long	MVCL	0E	RR	R1, R2	Move char from area spec in opr 2 to area spec in opr 1. Opr 2 is even/odd reg pair where R2 is "from adr", R2+1 bits 0-7 is padding char, and R2+1 bits 8-31 is length. Opr 1 is even/odd reg. pair where R1 is "to" addr, R1+1 bits 8-31 is length.	Addr Specif	0 Opr cnts = 1 Opr 1 cnt< opr 2 cnt 2 Opr 1 cnt> opr 2 cnt 3 No move due to destructive overlap.

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Move Numerics	MVN	D1	SS	D1(L,B1), D2(B2)	The 4 low-order bits of opr 2 bytes into the 4 low-order bits of opr 1 bytes. (Left to right byte by byte) (Max number of bytes moved: 256) (High-order bits of each byte of both opr's unchanged.) (No restriction on overlapping fields.)	Addr Protect	Unchanged
Move with Offset	MVO	F1	SS	D1(L1,B1), D2(L2,B2)	Opr 2 to the left of and adjacent to the low-order 4 bits of opr 1. (Right to left byte by byte) (Data can be packed, unpacked, or binary format) (No restriction on overlapping fields) (Processing terminated by high-order bit in opr 1) (If opr 2 field shorter than opr 1, insert leading zeros in opr 2.)	Addr Protect	Unchanged
Move Zones	MVZ	D3	SS	D1(L,B1), D2(B2)	The 4 high-order bits of opr 2 bytes into the 4 high-order bits of opr 1 bytes (Left to right byte by byte) (Max number of bytes moved: 256) (Low-order bits of each byte of both opr's unchanged.) (No restriction on overlapping fields)	Addr Protect	Unchanged
Multiply	M	5C	RX	R1, D2(X2,B2)	Multiply opr 1 by opr 2 Product: even and odd pair regs Opr 1 becomes the product. (Opr 1 must specify an even-numbered reg) (Sign bit extended to 1st significant product digit)	Addr Specif	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Multiply	MR	1C	RR	R1, R2	Multiply opr 1 by opr 2 Product: even and odd pair of regs Opr 1 becomes the product. (Opr 1 must specify an even-numbered reg) (Sign bit extended to 1st significant product digit)	Specif	Unchanged
Multiply (Extended)	MXR	26	RR	R1, R2	Multiply extended opr 1 by extended opr 2 (FPR pair) (FPR pair) Extended product is put in opr 1 (FPR pair) (Only FPR 0 and FPR 4 may be specified for either opr 1 or opr 2) (Low-order characteristic is made 14 < high-order characteristic except when the result would be > 0, then the low-order characteristic is made 128 > its correct value; sign of low-order characteristic remains the same as high-order characteristic)	Specif Exp Oflo Exp Uflo Opera	Unchanged
Multiply Decimal	MP	FC	SS	D1(L1,B1), D2(L2,B2)	Multiply opr 1 by opr 2 Multiplier: 8 bytes max size and shorter than the multiplicand. Multiplicand: must have high-order zeros equal to or greater than the size of the multiplier. (Both opr's in packed format) (Right to left byte by byte) Product: must contain at least 1 high-order zero.	Addr Specif Data Protect Opera	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Multiply Halfword	MH	4C	RX	R1, D2(X2,B2)	Multiply opr 1 by opr 2 (Opr 2 is expanded to a 32-bit integer) (Only the low-order 32 bits of the product, opr 1, are retained)	Addr Specif	Unchanged
Multiply (Long)	MD	6C	RX	R1, D2(X2,B2)	Multiply opr 1 by opr 2 (FPR) (Sto) Product: prenormalizes the opr's and post-normalizes the intermediate product. (If all fraction digits (15) = zero; the product, sign and char are made zero.) (The intermediate product fraction is truncated before left-shifting.)	Addr Specif Exp Oflo Exp Uflo Opera	Unchanged
Multiply (Long)	MDR	2C	RR	R1, R2	Multiply opr 1 by opr 2 (FPR) (FPR) Product: prenormalizes the opr's and post-normalizes the intermediate product. (If all fraction digits (15) = 0; the product sign and char are made zero.) (The intermediate product fraction is truncated before left-shifting.)	Specif Exp Oflo Exp Uflo Opera	Unchanged
Multiply (Long to Extended)	MXD	67	RX	R1, D2(X2,B2)	Multiply long opr 1 by long opr 2. (FPR) (Sto) Extended product is put in FPR pair specified by opr 1 (Only FPR 0 and FPR 4 may be specified for opr 1) (Signs of FPR pair are the same) (Can only use doubleword boundary in storage) (Continued)	Addr Specif Exp Oflo Exp Uflo Protect Opera	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Multiply (Long to Extended) (Cont'd)	MXD	67	RX	R1, D2(X2,B2)	(Low-order characteristic is made 14< high-order characteristic except when the result would be > 0, then the low-order characteristic is made 128 > its correct value; sign of low-order characteristic remains the same as high-order characteristics)		
Multiply (Long to Extended)	MXDR	27	RR	R1, R2	Multiply long opr 1 by long opr 2. (FPR) (FPR) Extended product is put in FPR pair specified by opr 1 (Only FPR 0 and FPR 4 may be specified for opr 1) (Signs of FPR pair are the same) (Low-order characteristic is made 14< high-order characteristic except when the result would be > 0, then the low-order characteristic is made 128> its correct value; sign of low-order characteristic remains the same as the high-order characteristic)	Specif Exp Oflo Exp Uflo Opera	Unchanged
Multiply (Short)	ME	7C	RX	R1, D2(X2,B2)	Multiply opr 1 by opr 2 (FPR) (Sto) Product: prenormalizes the opr's and post-normalizes the intermediate product. (If all fraction digits (14) = 0; the product sign and char are made zero.) (The intermediate product fraction is truncated before left-shifting.) (The 2 low-order fraction digits of the product always = zero.)	Addr Specif Exp Oflo Exp Uflo Opera	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Multiply (Short)	MER	3C	RR	R1, R2	Multiply opr 1 by opr 2 (FPR) (FPR) Product: prenormalizes the opr's and post-normalizes the intermediate product. (If all fraction digits (14) = 0; the product sign and char are made zero.) (The intermediate product fraction is truncated before left-shifting.)	Specif Exp Oflo Exp Uflo Opera	Unchanged
No Operation	NOP	47(BC 0)	RX, Ext.	D2(X2,B2)	Comp mask with cond code	None	Unchanged
No Operation	NOPR	07(BCR 0)	RR, Ext.	R2	Comp mask with cond code	None	Unchanged
OR Logical	O	56	RX	R1, D2(X2,B2)	The ORed sum of both opr's into opr 1	Addr Specif	0 Result = 0 1 Result ≠ 0
OR Logical	OC	D6	SS	D1(L,B1), D2(B2)	The ORed sum of both opr's into opr 1 (Left to right byte by byte) (Max number of bytes ORed: 256)	Addr Protect	0 Result = 0 1 Result ≠ 0
OR Logical	OR	16	RR	R1, R2	The ORed sum of both opr's into opr 1	None	0 Result = 0 1 Result ≠ 0
OR Logical Immediate	OI	96	SI	D1(B1), I2	OR the 1 byte from the instruction stream (8-15) to opr 1	Addr Protect	0 Result = 0 1 Result ≠ 0
Pack	PACK	F2	SS	D1(L1,B1), D2(L2,B2)	Change opr 2 from zoned to packed format and place into opr 1. (Right to left byte by byte) (No restriction on overlapping fields) (Opr 2 may be extended with hi-order zeros)	Addr Protect	Unchanged
Purge Translation Lookaside Buffer	PTLB	B20D	S	---	Invalidate current info in TLB.	Priv Opera	Unchanged

Ext. = Extended Mnemonic

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Read Direct	RDD	85	SI	D1(B1), I2	The 1 byte from the instruction stream (8-15) is placed on the signal-out, in a form of 8 timing pulses, along with a 9th pulse at the read-out line. The 8 bit lines at the direct-in lines are stored in 0 or 1.	Priv Addr Protect Opera	Unchanged
Reset Reference Bit	RRB	B213	S	D1(B1)	Set refence-bit=0 for 2048 byte block referenced by opr 1. CC indicates setting of ref and change bits prior to exec of this instruction.	Priv Opera	0 Ref = 0 Chg = 0 1 Ref = 0 Chg = 1 2 Ref = 1 Chg = 0 3 Ref = 1 Chg = 1
Set Clock	SCK	B204	S	D1(B1)	Replace curr val of TOD clock with eight bytes starting at opr 1.	Addr Specif Priv Protect Opera	0 Clock val set 1 Clock val secure 2 -- 3 Clock not oper
Set Clock Comparator	SCKC	B206	S	D1(B1)	Dblwd at opr 1 replaces curr value of clock comparator	Addr Priv Specif Protect Opera	Unchanged
Set CPU Timer	SPT	B208	S	D1(B1)	Dblwd at opr 1 replaces curr value of CPU timer.	Addr Priv Specif Protect Opera	Unchanged
Set Prefix	SPX	B210	S	D2(B2)	Prefix reg contents replaced by contents of bit pos 8-19 of word located by opr 2 address.	Specif Opera Priv	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Set Program Mask	SPM	04	RR	R1	Opr 1 (2-7) replaces the cond code and program mask bits of the current PSW (34-39) (Bits 0, 1 and 8-31 of opr 1 are ignored and unchanged.)	None	Set by bits 2 and 3
Set PSW Key From Address	SPKA	B20A	S	D1(B1)	Protection key of PSW replaced by bits 24-27 of the opr address.	Opera Priv	Unchanged
Set Storage Key	SSK	08	RR	R1, R2	Opr 1 (24-30) replaces the storage key specified by opr 2 (Opr 1 bits 0-23 and 31 are ignored) (Opr 2 bits 0-7 and 21-27 are ignored) (Bits 28-31 must be zero)	Addr Priv Specif Opera	Unchanged
Set System Mask	SSM	80	S	D1(B1)	Opr 1 (1 byte) replaces the system mask bits of the current PSW (0-7).	Priv Addr	Unchanged
Shift and Round Decimal	SRP	F0	SS	D1(L1,B1), D2(B2), I3	Shift opr 1 as specified by opr 2. If shift is right, round by factor in opr 3.	Protect Opera Addr Data Dec Oflo	0 Result = 0 1 Result < 0 2 Result > 0 3 Result Oflo
Shift Left Double Algebraic	SLDA	8F	RS	R1, D2(B2)	Opr 1 (even and odd regs) is shifted left the number of times equal to opr 2 (low-order 6 bits).	Specif Fxpt Oflo	0 Result = 0 1 Result < 0 2 Result > 0 3 Overflow
Shift Left Double Logical	SLDL	8D	RS	R1, D2(B2)	Opr 1 (even and odd regs) is shifted left the number of times equal to opr 2 (low-order 6 bits). (Hi-order bit participates in the shift)	Specif	Unchanged
Shift Left Single Algebraic	SLA	8B	RS	R1, D2(B2)	Opr 1 is shifted left the number of times equal to opr 2 (low-order 6 bits).	Fxpt Oflo	0 Result = 0 1 Result < 0 2 Result > 0 3 Overflow

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Shift Left Single Logical	SLL	89	RS	R1, D2(B2)	Opr 1 is shifted left the number of times equal to opr 2 (low-order 6 bits). (Hi-order bit participates in the shift)	None	Unchanged
Shift Right Double Algebraic	SRDA	8E	RS	R1, D2(B2)	Opr 1 (even and odd regs) is shifted right the number of times equal to opr 2 (Low-order 6 bits).	Specif	0 Result = 0 1 Result < 0 2 Result > 0
Shift Right Double Logical	SRDL	8C	RS	R1, D2(B2)	Opr 1 (even and odd regs) is shifted right the number of times equal to opr 2 (low-order 6 bits). (Hi-order bit participates in the shift)	Specif	Unchanged
Shift Right Single Algebraic	SRA	8A	RS	R1, D2(B2)	Opr 1 is shifted right the number of times equal to opr 2 (low-order 6 bits). (Shifting (+) numbers: vacated bits are replaced with zeros.) (Shifting (–) numbers: vacated bits are replaced with ones.)	None	0 Result = 0 1 Result < 0 2 Result > 0
Shift Right Single Logical	SRL	88	RS	R1, D2(B2)	Opr 1 is shifted right the number of times equal to opr 2 (low-order 6 bits). (Vacated bits are replaced with zeros) (Hi-order bit participates in the shift)	None	Unchanged
Signal Processor	SIGP	AE	RS	R1, R3, D2(B2)	An eight-bit order code (bits 24–31 of the second-operand address) is transmitted to the CPU designated by the processor address (bits 16–31) in the third operand.	Opera Priv	0 = Order code accepted 1 = Status stored 2 = Channel or subchannel busy 3 = Channel not operational
Start I/O	SIO	9C00	S	D1(B1)	Opr 1 (16–31) identifies the selected chan, ctl unit and I/O device to perform write, read, read bkwd, control or sense oper. The CAW at loc 48 is fetched, which locates the first CCW. The SIO is initiated providing the addressed chan, ctl unit and I/O device are available without pending interrupt errors. Exceptional conditions pending (Full op cd – 1001 1100 xxxx xxx0)	Priv	0 = I/O oper initiated and chan proceeding with operation. 1 = CSW stored 2 = Chan or subchannel busy 3 = Not operational

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Start I/O Fast Release	SIOF	9C01	S	D1(B1)	This instruction takes advantage of the block-multiplex channel, but is otherwise identical to SIO. (Full op cd - 1001 1100 xxxx xxx1).	Priv	Same as SIO
Store	ST	50	RX	R1, D2(X2,B2)	Opr 1 is stored into opr 2.	Addr Specif Protect	Unchanged
Store Channel ID	STIDC	B203	S	D1(B1)	Store opr 1 at loc 168 in main storage.	Priv Opera	0 ID stored 1 CSW stored 2 Chan activity ID not stored 3 Not oper.
Store Character	STC	42	RX	R1, D2(X2,B2)	Opr 1 (24-31) replaces the character at opr 2's address.	Addr Protect	Unchanged
Store Characters Under Mask	STCM	BE	RS	R1, M3, D2(B2)	Bytes selected from opr 1 under control of mask are stored at opr 2.	Addr Opera Protect	Unchanged
Store Clock	STCK	B205	S	D1(B1)	Current val of TOD clock stored in 8 bytes at opr 1.	Addr Protect Opera	0 Clock in set state 1 Clk in not-set state 2 Clk in error 3 Clk not oper or in stopped state
Store Clock Comparator	STCKC	B207	S	D1(B1)	Curr contents of clock comparator stored at opr 1.	Addr Priv Specif Protect Opera	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Store Control	STCTL	B6	RS	R1, R3, D2(B2)	Control regs from opr 1 to opr 3 stored at opr 2.	Priv Addr Specif Protect Opera	Unchanged
Store CPU Address	STAP	B212	S	D2(B2)	CPU address stored at halfword location designated by second-operand address.	Specif Opera Priv	Unchanged
Store CPU ID	STIDP	B202	S	D1(B1)	CPU info stored in 8 bytes at opr1.	Priv Addr Specif Protect Opera	Unchanged
Store CPU Timer	STPT	B209	S	D1(B1)	Curr contents of CPU timer stored in dblwd at opr 1.	Priv Addr Specif Protect Opera	Unchanged
Store Halfword	STH	40	RX	R1, D2(X2,B2)	Opr 1 (16 low-order bits) is stored at opr 2's location. (Hi-order bits, opr 1, ignored and unchanged)	Addr Specif Protect	Unchanged
Store (Long)	STD	60	RX	R1, D2(X2,B2)	FP opr 1 to opr 2's location.	Addr Protect Specif Opera	Unchanged
Store Multiple	STM	90	RS	R1, R3, D2(B2)	R1 thru R3 are stored at opr 2's location in ascending order. Starting reg specified by R1, ending reg specified by R3. (Reg wrap-around possible)	Addr Specif Protect	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Store Prefix	STPX	B211	S	D2(B2)	Prefix register contents are stored at word location designated by second operand address.	Specif Opera Priv	Unchanged
Store (Short)	STE	70	RX	R1, D2(X2,B2)	FP opr 1 is stored at opr 2's location (Low-order half of FPR ignored and unchanged)	Opera Addr Specif Protect	Unchanged
Store Then AND System Mask	STNSM	AC	SI	D1(B1), I2	Bits 0-7 current PSW stored at opr 1, then these bits ANDed with opr 2 and replaced in current PSW.	Addr Priv Protect Opera	Unchanged
Store Then OR System Mask	STOSM	AD	SI	D1(B1), I2	Bits 0-7 of current PSW stored at opr 1, then these bits ORed with opr 2 and replaced in current PSW.	Addr Priv Protect Opera	Unchanged
Subtract	S	5B	RX	R1, D2(X2)	Subtract opr 2 from opr 1 and place the difference into opr 1.	Addr Fxpt Oflo Specif	0 Dif = 0 1 Dif < 0 2 Dif > 0 3 Overflow
Subtract	SR	1B	RR	R1, R2	Subtract opr 2 from opr 1; difference placed into opr 1.	Fxpt Oflo	0 Dif = 0 1 Dif < 0 2 Dif > 0 3 Overflow

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Subtract Decimal	SP	FB	SS	D1(L1,B1), D2(L2,B2)	Subtract dec opr 2 from opr 1; difference stored into opr 1. (Right to left byte by byte) (Both opr's must be in packed format) (Fields can overlap if low-order bytes coincide)	Opera Addr Data Dec Oflo Protect	0 Dif = 0 1 Dif < 0 2 Dif > 0 3 Overflow
Subtract Halfword	SH	4B	RX	R1, D2(X2,B2)	Opr 2 halfword expanded to fullword and subtracted from opr 1; difference placed into opr 1.	Addr Fxpt Oflo Specif	0 Dif = 0 1 Dif < 0 2 Dif > 0 3 Overflow
Subtract Logical	SL	5F	RX	R1, D2(X2,B2)	Subtract opr 2 from opr 1; difference placed into opr 1.	Addr Specif	0 -- 1 Dif ≠ 0 No Carry 2 Dif = 0 Carry 3 Dif ≠ 0 Carry
Subtract Logical	SLR	1F	RR	R1, R2	Subtract opr 2 from opr 1; difference placed into opr 1.	None	0 -- 1 Dif ≠ 0 No Carry 2 Dif = 0 Carry 3 Dif ≠ 0 Carry
Subtract Normalized (Extended)	SXR	37	RR	R1, R2	FP subtract extended opr 2 from extended opr 1. (FPR pair) (FPR pair) Extended difference is put in opr 1 (FPR pair) (Sign of extended opr 2 is inverted before the addition) (Only FPR 0 and FPR 4 may be specified for either opr 1 or opr 2) (Continued)	Specif Exp Oflo Exp Uflo Signif	0 Fract = 0 1 Fract < 0 2 Fract > 0 3 --

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Subtract Normalized (Extended) (Cont'd)	SXR	37	RR	R1, R2	(High-order and low-order signs of a FPR pair are always the same in extended precision) (Low-order characteristic is made 14 < high-order characteristic except when the result would be > 0, then the low-order characteristic is made 128 > its correct value; sign of low-order characteristic remains the same as high-order characteristic)		
Subtract Normalized (Long)	SD	6B	RX	R1, D2(X2,B2)	FP Subtract opr 2 from opr 1 and the difference placed into opr 1. (The sign of opr 2 is inverted before the addition.)	Addr Specif Signif Exp Oflo Exp Uflo	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo
Subtract Normalized (Long)	SDR	2B	RR	R1, R2	FP Subtract opr 2 from opr 1 (FPR) (FPR) (The sign of opr 2 is inverted before the addition.)	Specif Signif Exp Oflo Exp Uflo	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo
Subtract Normalized (Short)	SE	7B	RX	R1, D2(X2,B2)	FP Subtract opr 2 from opr 1 (The sign of opr 2 is inverted before the addition.) (Low-order halves of FPR ignored and unchanged).	Addr Specif Signif Exp Oflo Exp Uflo	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Subtract Normalized (Short)	SER	3B	RR	R1, R2	Subtract opr 2 from opr 1 (The sign of opr 2 is inverted before the addition.) (Low-order halves of FPRs ignored and unchanged)	Specif Signif Exp Oflo Exp Uflo	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo
Subtract Unnormalized (Long)	SW	6F	RX	R1, D2(X2,B2)	FP Subtract opr 2 from opr 1 (Sto) (FPR) (The sign of opr 2 is inverted before the addition.)	Addr Specif Signif Exp Oflo Opera	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo
Subtract Unnormalized (Long)	SWR	2F	RR	R1, R2	FP Subtract opr 2 from opr 1 (FPR) (FPR) (The sign of opr 2 is inverted before the addition.)	Specif Signif Exp Oflo Opera	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo
Subtract Unnormalized (Short)	SU	7F	RX	R1, D2(X2,B2)	FP Subtract opr 2 from opr 1 (Sto) (FPR) (Low-order half of FPR ignored and unchanged) (The sign of opr 2 is inverted before the addition.)	Addr Specif Signif Exp Oflo Opera	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo
Subtract Unnormalized (Short)	SUR	3F	RR	R1, R2	FP Subtract opr 2 from opr 1 (FPR) (FPR) (Low-order halves of FPRs ignored and unchanged) (The sign of opr 2 is inverted before the addition.)	Specif Signif Exp Oflo Opera	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 Exp Oflo

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Supervisor Call	SVC	OA	RR	I	Immediate bits (8-15) placed in loc. 138 and PSW swap performed. (16-23) are made zero. (Old PSW at loc 32). (New PSW from loc 96).	None	Unchanged
Test and Set	TS	93	SI	D1(B1)	Hi-order bit of 1st byte of opr adr sets cond code. Entire byte then set to 1's	Addr Protect	0 Hi-order bit = 0 1 Hi-order bit = 1 2 -- 3 --
Test Channel	TCH	9F	S	D1(B1)	Opr 1 (16-23) identifies the tested channel. (Bits 24-31 are ignored.) (Instruction checks the channel's status and sets appropriate cond code.)	Priv	0 Chan Avl 1 Int Pending 2 Chan in Burst Mode 3 Chan not Operational
Test I/O	TIO	9D	S	D1(B1)	Opr 1 (16-31) identifies the tested channel, control unit, and I/O device. Used to clear a pending interrupt. (CSW stored at loc 64): Subchannel contains a pending interrupt. I/O device contains a pending interrupt. Control unit or I/O device is executing a previous operation or a pending channel-end/control unit-end for another I/O device. Channel or I/O device equipment error or device not ready.	Priv	0 Available 1 CSW Stored 2 Channel or Subchan Busy 3 Not Operational

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Test Under Mask	TM	91	SI	D1(B1), I2	Immediate bits (8-15) used as a mask to compare against opr 1. Mask bit 1: storage bit tested. Mask bit 0: storage bit ignored.	Addr	0 Selected bits all zero (mask is all zero) 1 Selected bits mixed 0's and 1's 3 Selected bits all 1's
Translate	TR	DC	SS	D1(L,B1), D2(B2)	Opr 1 (argument byte) is added to initial adr of opr 2; this generated adr is the loc of the function byte. The function byte replaces the original argement byte. (The operation proceeds left to right, byte-by-byte, and terminates when opr 1 field is exhausted. All data is valid.)	Addr Protect	Unchanged
Translate and Test	TRT	DD	SS	D1(L,B1), D2(B2)	Opr 1 (argument byte) is added to initial adr of opr 2; this generated adr is the loc of the function byte. When the function byte is zero, the next function byte is inspected. When the function byte is nonzero, the argument byte adr is placed in GPR 1, bits 8-31, and the function byte is placed in GPR 2, bits 24-31 (all high-order bits are unchanged), and the operation is terminated. (The operation proceeds left-to-right, byte-by-byte, and terminates when opr 1 field is exhausted.)	Addr	0 All function bytes 0 1 Non-0 function byte met 2 Last function byte non-0 3 Not used

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Unpack	UNPK	F3	SS	D1(L1,B1), D2(L2,B2)	Change opr 2 from packed to zoned format and place into opr 1. (Right to left byte by byte) (No restrictions on overlapping fields) (Opr 2 may be extended with hi-order zeros.)	Addr Protect	Unchanged
Write Direct	WRD	84	SI	D1(B1), I2	The 1 byte from the instruction stream (8-15) is placed on the timing signal out, in a form of 8 timing pulses, along with a 9th pulse at the write-out line. The 8 bit lines at the direct-out lines are brought up by opr 1.	Priv Addr Opera	Unchanged
Zero and Add	ZAP	F8	SS	D1(L1,B1), D2(L2,B2)	Opr 1 cleared and opr 2 placed in opr 1 (Low-order opr's may coincide) (Opr 2 must be in packed format) (Opr 1 field must be large enough for all opr 2 significant digits) (Opr 2 extended with zeros to fill opr 1.)	Addr Data Dec Oflo Protect Opera	0 Result = 0 1 Result < 0 2 Result > 0 3 Overflow

Operation	Name Entry	Operand Entry	Is used to:
ACTR	A sequence symbol or blank	A SETA expression	Limit the number of AGO and AIF operations executed: prevent incessant looping.
AGO	A sequence symbol or blank	A sequence symbol	Unconditionally alter the sequence in which statements are processed.
AIF	A sequence symbol or blank	A logical expression enclosed in parentheses, immediately followed by a sequence symbol	Conditionally alter the sequence in which statements are processed.
ANOP	A sequence symbol or blank	Must not be present	Act as the target of AGO and AIF instructions.
CCW	Any symbol or blank	Four operands, separated by commas	Define and generate an 8-byte Channel Command word having doubleword alignment.
CNOP	Any symbol or blank	Two absolute expressions, separated by a comma	Align the location on a specified halfword boundary.
COM	Any symbol or blank	Must not be present	Reserve a common area of storage referred to by independent assemblies that are linked and loaded together or execution.
COPY	Must not be present	One ordinary symbol	Obtain and copy source code from a PDS member into the program being assembled.
CSECT	Any symbol or blank	Must not be present	Identify the beginning or continuation of a control section (see DSECT instruction).
CXD	Any symbol or blank	Must not be present	Allocate a fullword that will contain the sum of the lengths of all external dummy sections when the program is executed.
DC	Any symbol or blank	One or more operands, separated by commas	Define data constants in storage (see DS instruction).
DROP	A sequence symbol or blank	One to sixteen absolute expressions, separated by commas; or blank	Inform the assembler that specified registers are no longer to be used as base registers (see USING instruction).
DS	Any symbol or blank	One or more operands, separated by commas	Reserve areas of storage without assembling their contents (see DC instruction).
DSECT	Any symbol or blank	Must not be present	Initiate or continue a dummy section; describe an area of storage without reserving it (see CSECT instruction).
DXD	Any symbol	One or more operands, separated by commas	Identify and define an external dummy section.

Operation	Name Entry	Operand Entry	Is used to:
EJECT	A sequence symbol or blank	Must not be present	Start a new page in the assembly listing; specify the sectioning of the assembly listing.[1]
END	A sequence symbol or blank	A relocatable expression or blank	Terminate the assembly of a source module.
ENTRY	A sequence symbol or blank	One or more relocatable symbols separated by commas	Identify symbols that are defined in the same source module, but are referred to in another source module (see EXTRN, WXTRN instructions).
EQU	An ordinary symbol or a variable symbol	One to three operands, separated by commas	Assign values to symbols.
EXTRN	A sequence symbol or blank	One or more relocatable symbols, separated by commas	Identify symbols that are referred to in the same source module but are defined in another source module (see ENTRY, WXTRN instructions).
GBLA GBLB GBLC	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas.[2]	Define a global Arithmetic, Binary, or Character SET symbol.
ICTL	Must not be present	One to three decimal values, separated by commas	Alter the position of Begin, End, and Continuation columns in the source module.
ISEQ	Must not be present	Two decimal values, separated by commas	Sequence-check the source module statements.
LCLA LCLB LCLC	Must not be present	One or more variable symbols, that are to be used as SET symbols, separated by commas.[2]	Define a local Arithmetic, Binary, or Character SET symbol.
LTORG	Any symbol or blank	Not required	Position a literal pool at other than the end of the first control section to ensure addressability of literals in a large control section.
MACRO[3]	Must not be present	Not required	Indicate the beginning of a macro definition.
MEND[3]	A sequence symbol or blank	Not required	Indicate the end of a macro definition.
MEXIT[3]	A sequence symbol or blank	Not required	Indicate an exit from a macro definition.
MNOTE	A sequence symbol or blank	A severity code (optional), comma, characters enclosed in apostrophes	Display an error severity code, generate a message.

Operation	Name Entry	Operand Entry	Is used to:
OPSYN	An ordinary symbol	A mnemonic operation, a macro or assembler operation, a machine instruction operation, or a blank	Define a symbol to represent an operation code, or delete its properties as an operation code.
ORG	Any symbol or blank	A relocatable expression	Change the location counter to redefine portions of a control section, especially, constant tables.
POP	A sequence symbol or blank	One or more operands, separated by commas	Restore the PRINT or USING status saved by the most recent PUSH instruction.
PRINT	A sequence symbol or blank	One to three operands	Control the amount of detail printed in the assembly listing.
PUNCH	A sequence symbol or blank	One to eighty characters, enclosed in apostrophes	Punch one card with the data specified in the operand, substituting values for variable symbols (see REPRO instruction).
PUSH	A sequence symbol or blank	One or more operands, separated by a comma	Save the current PRINT or USING status (see POP instruction).
REPRO	A sequence symbol or blank	Not required	Punch one card with the characters specified in the statement that follows (see PUNCH instruction).
SETA SETB SETC	A SETA symbol A SETB symbol A SETC symbol	An arithmetic expression, a logical expression, or a character expression	Assign a value to an Arithmetic, Binary, or Character SET symbol.
SPACE	A sequence symbol or blank	A decimal self-defining term or blank	Insert blank lines into the source module assembly listing to separate sections of code.[1]
START	Any symbol or blank	A self-defining term or blank	Initialize the location counter for, and name the first control section of the module.
TITLE	A variable symbol, and/or character string, or sequence symbol, or blank	One to 100 characters, enclosed in apostrophes	Produce headings on the assembly listing pages, punch identifying characters into the object deck.[1]
USING	A sequence symbol or blank	An absolute or relocatable expression followed by 1 to 16 absolute expressions, separated by commas	Identify registers that may be used by the assembler as base registers (see DROP instruction).

Operation	Name Entry	Operand Entry	Is used to:
WXTRN	A sequence symbol or blank	One or more relocatable symbols, separated by commas	Identify symbols referred to in the same source module, but defined in another source module in the same load module (see EXTRN instruction).

[1] The statement itself does not appear in the assembly listing.
[2] SET symbols can be defined as subscripted SET symbols.
[3] Can be used only as part of a macro definition.

System Assembler Statements

Instruction	Name Entry	Operand Entry
Model Statements	An ordinary symbol, a variable symbol, a sequence symbol, a combination of variable symbols and other characters that is equivalent to a symbol, or blank	Any combination of characters (including variable symbols)
Prototype Statement[1]	A symbolic parameter or blank	Zero or more operands that are symbolic parameters, separated by commas
Macro-Instruction Statement[2]	An ordinary symbol, a variable symbol, a sequence symbol, a combination of variable symbols and other characters that is equivalent to a symbol,[2] or blank	Zero or more positional operands and/or zero or more keyword operands separated by commas[2]
Assembler Language Statement	An ordinary symbol, a variable symbol, a sequence symbol, a combination of variable symbols and other characters that is equivalent to a symbol, or blank	Any combination of characters (including variable symbols)

[1] Can only be used as part of a macro definition.

[2] Variable symbols appearing in a macro instruction are replaced by their values before the macro instruction is processed.

System Assembler Constants

TYPE	IMPLICIT LENGTH (BYTES)	ALIGN-MENT	LENGTH MODI-FIER RANGE	SPECIFIED BY	NUMBER OF CON-STANTS PER OPERAND	RANGE FOR EX-PONENTS	RANGE FOR SCALE	TRUN-CATION/ PADDING SIDE
C	as needed	byte	.1 to 256 (1)	characters	one			right
X	as needed	byte	.1 to 256 (1)	hexadecimal digits	multi-ple			left
B	as needed	byte	.1 to 256	binary digits	multi-ple			left
F	4	word	.1 to 8	decimal digits	multi-ple	-85 to +75	-187 to +346	left (3)
H	2	half word	.1 to 8	decimal digits	multi-ple	-85 to +75	-187 +346	left (3)
E	4	word	.1 to 8	decimal digits	multi-ple	-85 to +75	0-14	right (3)
D	8	double word	.1 to 8	decimal digits	multi-ple	-85 to +75	0-14	right (3)
L	16	double word	.1 to 16	decimal digits	multi-ple	-85 to +75	0-28	right (3)
P	as needed	byte	.1 to 16	decimal digits	multi-ple			left
Z	as needed	byte	.1 to 16	decimal digits	multi-ple			left
A	4	word	.1 to 4 (2)	any expression	multi-ple			left
Q	4	word	1-4	symbol nam-ing a DXD or DSECT	multi-ple			left
V	4	word	3, 4	relocatable symbol	multi-ple			left
S	2	half word	2 only	one absolute or relocat-able ex-pression or two absolute expressions: exp (exp)	multi-ple			
Y	2	half word	.1 to 2 (2)	any expression	multi-ple			left

(1) In a DS assembler instruction C and X type constants can have length specification to 65535.
(2) Bit length specification permitted with absolute expressions only. Relocatable A-type constants, 3 or 4 bytes only; relocatable Y-type constants, 2 bytes only.
(3) Errors will be flagged if significant bits are truncated or if the value specified cannot be contained in the implicit length of the constant.

	Variable Symbols													Attributes						
		Global SET Symbols			Local SET Symbols			System Variable Symbols												
Requirement	Symbolic Parameter	SETA	SETB	SETC	SETA	SETB	SETC	&SYSNDX	&SYSECT	&SYSLIST	&SYSPARM	&SYSDATE	&SYSTIME	Type	Length	Scaling	Integer	Count	Number	Sequence Symbol
MACRO																				
Prototype Statement	Name Operand																			
GBLA		Operand																		
GBLB			Operand																	
GBLC				Operand																
LCLA					Operand															
LCLB						Operand														
LCLC							Operand													
Model Statement	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Operand	Operand							Name
SETA	Operand[2]	Name Operand	Operand[3]	Operand[9]	Name Operand	Operand[3]	Operand[9]	Operand		Operand[2]	Operand[9]				Operand	Operand	Operand	Operand	Operand	
SETB	Operand[6]	Operand[6]	Name Operand	Operand[6]	Operand[6]	Name Operand	Operand[6]	Operand[6]	Operand[4]	Operand[6]	Operand[6]			Operand[4]	Operand[5]	Operand[5]	Operand[5]	Operand[5]	Operand[5]	
SETC	Operand	Operand[7]	Operand[8]	Name Operand	Operand[7]	Operand[8]	Name Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand						
AIF	Operand[6]	Operand[6]	Operand	Operand[6]	Operand[6]	Operand	Operand[6]	Operand[6]	Operand[4]	Operand[6]	Operand[6]			Operand[4]	Operand[5]	Operand[5]	Operand[5]	Operand[5]	Operand[5]	Name Operand
AGO																				Name Operand
ACTR	Operand[2]	Operand	Operand[3]	Operand[2]	Operand	Operand[3]	Operand[2]	Operand		Operand[2]	Operand[2]				Operand	Operand	Operand	Operand	Operand	
ANOP																				Name
MEXIT																				Name
MNOTE	Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand	Operand							Name
MEND																				Name
OUTER Macro		Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand				Name Operand	Operand	Operand							Name
Inner Macro	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Name Operand	Operand	Operand							Name
Assembler Language Statement		Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand	Name Operation Operand													Name

1. Variable symbols in macro instructions are replaced by their values before processing.
2. Only if value is self-defining term.
3. Converted to arithmetic +1 or +0.
4. Only in character relations.
5. Only in arithmetic relations.
6. Only in arithmetic or character relations.
7. Converted to unsigned number.
8. Converted to character 1 or 0.
9. Only if one to one decimal digits (from 0 through 2, 147, 483, 647).

System Assembler Conditional Assembly Expressions

Expression	Arithmetic Expressions	Character Expressions	Logical Expressions
Can contain	• Self-defining terms • Length, scaling, integer, count, and number attributes	• Any combination of characters enclosed in apostrophes	• A 0 or a 1 • SETB symbols
	• SETA and SETB symbols	• Any variable symbol enclosed in apostrophes	• Arithmetic relations[1]
	• SETC symbols whose values are a decimal self-defining term	• A concatenation of variable symbols and other characters enclosed in apostrophes	• Character relations[2] • Arithmetic value
	• &SYSPARM if its value is a decimal self-defining term		
	• Symbolic parameters if the corresponding operand is a decimal self-defining term	• A type attribute reference	
	• &SYSLIST (n) if the corresponding operand is a decimal self-defining term		
	• &SYSLIST (n,m) if the corresponding operand is a decimal self-defining term		
	• &SYSNDX		
Operations are	+, - (unary and binary), *, and /; parentheses permitted	concatenation, with a period (.)	AND, OR, and NOT parentheses permitted
Range of values	-2^{31} to $+2^{31}-1$	0 through 255 characters	0 (false) or 1 (true)
May be used in	• SETA operands • Arithmetic relations[1] • Subscripted SET symbols • SYSLIST subscript (s) • Substring notation • Sublist notation	• SETC operands • Character relations[2]	• SETB operands • AIF operands

[1] An arithmetic relation consists of two arithmetic expressions related by the operators GT, LT, EQ, NE, GE, or LE.

[2] A character relation consists of two character expressions related by the operator GT, LT, EQ, NE, GE, or LE. Type attribute notation and Substring notation may also be used in character relations. The maximum size of the character expressions that can be compared is 255 characters. If the two character expressions are of unequal size, the smaller one will always compare less than the larger.

System Assembler Attributes

Attribute	Notation	Can be used with:	Can be used only if type attribute is:	Can be used in:
Type	T'	Ordinary Symbols defined in open code; symbolic parameters inside macro definitions; SET symbols, &SYSPARM, &SYSDATE, &SYSTIME, inside or outside macro definitions; &SYSLIST (m), &SYSLIST (m,n), &SYSECT, &SYSNDX inside macro definitions	(May always be used)	1. SETC operand fields 2. Character relations
Length	L'	Ordinary Symbols defined in open code; symbolic parameters inside macro definitions; &SYSLIST (m), and &SYSLIST (n,n) inside macro definitions	Any letter except M,N,O,T and U	Arithmetic expressions
Scaling	S'	Ordinary Symbols defined in open code; symbolic parameters inside macro definitions; &SYSLIST (m), and &SYSLIST (m,n) inside macro definitions	H,F,G,D,E,L,K,P, and Z	Arithmetic expressions
Integer	I'	Ordinary Symbols defined in open code; symbolic parameters inside macro definitions; &SYSLIST (m), and &SYSLIST (m,n) inside macro definitions	H,F,G,D,E,L,K,P, and Z	Arithmetic expressions
Count	K'	Symbolic parameters inside macro definitions; SET symbols; all system variable symbols	Any letter	Arithmetic expressions
Number	N'	Symbolic parameters, &SYSLIST (m), and &SYSLIST (m,n) inside macro definitions	Any letter	Arithmetic expressions

System Assembler Variable Symbols

Variable Symbol	Declared by:	Initialized, or set to:	Value changed by:	May be used in:
Symbolic[1] parameter	Prototype statement	Corresponding macro instruction operand	(Constant throughout definition)	• Arithmetic expressions if operand is decimal self-defining term • Character expressions
SETA	LCLA or GBLA instruction	0	SETA instruction	• Arithmetic expressions • Character expressions
SETB	LCLB or GBLB instruction	0	SETB instruction	• Arithmetic expressions • Character expressions • Logical expressions
SETC	LCLC or GBLC instruction	String of length 0 (null)	SETC instruction	• Arithmetic expressions if value is decimal self-defining term • Character expressions
&SYSNDX[1]	The assembler	Macro instruction index	(Constant throughout definition; unique for each macro instruction)	• Arithmetic expressions • Character expressions
&SYSECT[1]	The assembler	Control section in which macro instruction appears	(Constant throughout definition; set by CSECT, DSECT, START, and COM)	• Character expressions

ystem Assembler Variable Symbols (cont'd)

Variable Symbol	Declared by:	Initialized, or set to:	Value changed by:	May be used in:
&SYSLIST[1]	The assembler	Not applicable	Not applicable	• N'&SYSLIST in arithmetic expressions
&SYSLIST (n)[1] &SYSLIST (n,M)[1]	The assembler	Corresponding macro instruction operand	(Constant throughout definition)	• Arithmetic expressions if operand is decimal self-defining term • Character expressions
&SYSPARM	PARM field	User defined or null	Constant throughout assembly	• Arithmetic expression if value is decimal self-defining term • Character expression
&SYSTIME	The assembler	System time	Constant throughout assembly	• Character expression
&SYSDATE	The assembler	System date	Constant throughout assembly	• Character expression

[1]Can be used only in macro definitions.

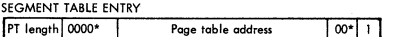
Dynamic Address Translation-Virtual-to-Real Address Translation

DYNAMIC ADDRESS TRANSLATION

VIRTUAL (LOGICAL) ADDRESS FORMAT

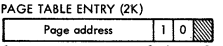
Segment Size	Page Size	Bits	Segment Index	Page Index	Byte Index
64K	2K	0 - 7	8 - 15	16 - 20	21 - 31
1M	2K	are ignored	8 - 11	12 - 20	21 - 31

SEGMENT TABLE ENTRY



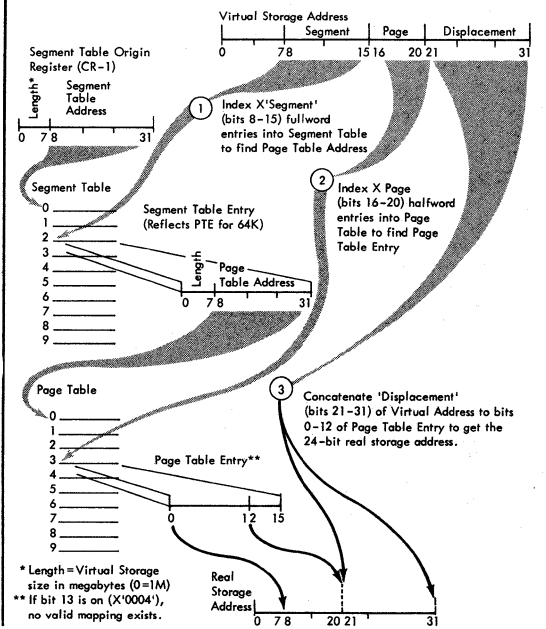
* Normally zeros; ignored on some models. 31 (1) Segment - invalid bit.

PAGE TABLE ENTRY (2K)



13 (1) Page - invalid bit

Virtual-to-Real Address Translation



* Length = Virtual Storage size in megabytes (0 = 1M)

** If bit 13 is on (X'0004'), no valid mapping exists.

Hexadecimal Addition, Multiplication, Subtraction Tables

Hexadecimal Addition and Subtraction Table

Example: 6 + 2 = 8, 8 - 2 = 6, and 8 - 6 = 2

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
2	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
3	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12
4	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
5	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14
6	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
7	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16
8	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17
9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18
A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A
C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Hexadecimal Multiplication Table

Example: 2 x 4 = 08, F x 2 = 1E

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
2	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
3	03	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
4	04	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	05	0A	0F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	06	0C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	07	0E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	08	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	09	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Decimal and Hexadecimal 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 decimal: (1) locate the largest decimal value in the table that will fit into the decimal number to be converted, and (2) note its hex equivalent and hex column position. (3) find the decimal remainder. Repeat the process on this and subsequent remainders.

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

	Hexadecimal positions to right of 'heximal point'					
	1	2	3	4	5	6
HEX =	DEC	DEC	DEC	DEC	DEC	DEC
0	0	0	0	0	0	0
1	0.0625	0.00390625	$2.44140625 \times 10^{-4}$	$1.52587890 \times 10^{-5}$	$9.53674316 \times 10^{-7}$	$5.96046447 \times 10^{-8}$
2	0.125	0.0078125	$4.88281250 \times 10^{-4}$	$3.05175781 \times 10^{-5}$	$1.90734863 \times 10^{-6}$	$1.19209289 \times 10^{-7}$
3	0.1875	0.01171875	$7.32421875 \times 10^{-4}$	$4.57763671 \times 10^{-5}$	$2.86102294 \times 10^{-6}$	$1.78813934 \times 10^{-7}$
4	0.25	0.015625	$9.76562500 \times 10^{-4}$	$6.10351562 \times 10^{-5}$	$3.81469726 \times 10^{-6}$	$2.38418579 \times 10^{-7}$
5	0.3125	0.01953125	$1.22070312 \times 10^{-3}$	$7.62939453 \times 10^{-5}$	$4.76837158 \times 10^{-6}$	$2.98023223 \times 10^{-7}$
6	0.375	0.0234375	$1.46484375 \times 10^{-3}$	$8.15527343 \times 10^{-5}$	$5.72204589 \times 10^{-6}$	$3.57627868 \times 10^{-7}$
7	0.4375	0.02734375	$1.70898437 \times 10^{-3}$	$1.06811523 \times 10^{-4}$	$6.67572021 \times 10^{-6}$	$4.17232513 \times 10^{-7}$
8	0.5	0.03125	$1.95312500 \times 10^{-3}$	$1.22070312 \times 10^{-4}$	$7.62939453 \times 10^{-6}$	$4.76837158 \times 10^{-7}$
9	0.5625	0.03515625	$2.19726562 \times 10^{-3}$	$1.37329101 \times 10^{-4}$	$8.58306884 \times 10^{-6}$	$5.36441803 \times 10^{-7}$
A	0.625	0.0390625	$2.44140625 \times 10^{-3}$	$1.52587890 \times 10^{-4}$	$9.53674316 \times 10^{-6}$	$5.96046447 \times 10^{-7}$
B	0.6875	0.04296875	$2.68554687 \times 10^{-3}$	$1.67846679 \times 10^{-4}$	$1.04904174 \times 10^{-5}$	$6.55651092 \times 10^{-7}$
C	0.75	0.046875	$2.92968750 \times 10^{-3}$	$1.83105468 \times 10^{-4}$	$1.14440918 \times 10^{-5}$	$7.15255737 \times 10^{-7}$
D	0.8125	0.05078125	$3.17382812 \times 10^{-3}$	$1.98364257 \times 10^{-4}$	$1.23977661 \times 10^{-5}$	$7.74860382 \times 10^{-7}$
E	0.875	0.0546875	$3.41796875 \times 10^{-3}$	$2.13623046 \times 10^{-4}$	$1.33514404 \times 10^{-5}$	$8.34465026 \times 10^{-7}$
F	0.9375	0.05859375	$3.66210937 \times 10^{-3}$	$2.28881835 \times 10^{-4}$	$1.43051147 \times 10^{-5}$	$8.94069671 \times 10^{-7}$

Powers of 16 Table

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
4 294 967 296	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

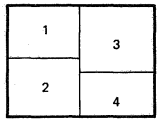
Negative Powers of 16 Table

n	16^n		
0	1.0		
−1	0.0625		
−2	0.0039	0625	
−3	0.0002	4414	0625
−4	1.5258	7890	6250×10^{-5}
−5	9.5367	4316	4062×10^{-7}
−6	5.9604	6447	7539×10^{-8}
−7	3.7252	9029	8461×10^{-9}
−8	2.3283	0643	6538×10^{-10}

Extended Binary Coded Decimal Interchange Code (EBCDIC)

The following 256-position table, outlined by the heavy black lines, shows the graphic characters and control character representations for EBCDIC. The bit-position numbers, bit patterns, hexadecimal representations and card hole patterns for these and other possible EBCDIC characters are also shown.

To find the card hole patterns for most characters, partition the 256-position table into four blocks as follows:



Block 1: Zone punches at top of table; digit punches at left
Block 2: Zone punches at bottom of table; digit punches at left
Block 3: Zone punches at top of table; digit punches at right
Block 4: Zone punches at bottom of table; digit punches at right

Fifteen positions in the table are exceptions to the above arrangement. These positions are indicated by small numbers in the upper right corners of their boxes in the table. The card hole patterns for these positions are given at the bottom of the table. Bit-position numbers, bit patterns, and hexadecimal representations for these positions are found in the usual manner.

Following are some examples of the use of the EBCDIC chart:

Character	Type	Bit Pattern	Hex	Hole Pattern	
				Zone Punches	Digit Punches
PF	Control Character	00 00 0100	04	12 - 9	- 4
%	Special Graphic	01 10 1100	6C	0	- 8 - 4
R	Upper Case	11 01 1001	D9	11	- 9
a	Lower Case	10 00 0001	81	12 - 0	- 1
	Control Character, function not yet assigned	00 11 0000	30	12 - 11 - 0 - 9	- 8 - 1

Bit Positions
01 23 4567

Extended Binary Coded Decimal Interchange Code (EBCDIC) (cont'd)

EBCDIC Codes

Bit Positions 4, 5, 6, 7	Second Hexadecimal Digit	Digit Punches	00				01				10				11				Bit Positions 0,1
			00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2,3
			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit
			12				12	12		12	12	12		12	12				
				11				11	11	11		11	11	11		11			Zone Punches
					0		0		0	0	0		0	0			0		
			9	9	9	9	9	9	9	9									Digit Punches
0000	0	8-1	① NUL	② DLE	③ DS	④	⑤ SP	⑥ &	⑦ -	⑧					⑨	⑩	⑪	⑫ 0	8-1
0001	1	1	SOH	DC1	SOS				⑬ /		a	j			A	J	⑭	1	1
0010	2	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2	2
0011	3	3	ETX	TM							c	l	t		C	L	T	3	3
0100	4	4	PF	RES	BYP	PN					d	m	u		D	M	U	4	4
0101	5	5	HT	NL	LF	RS					e	n	v		E	N	V	5	5
0110	6	6	LC	BS	ETB	UC					f	o	w		F	O	W	6	6
0111	7	7	DEL	IL	ESC	EOT					g	p	x		G	P	X	7	7
1000	8	8		CAN							h	q	y		H	Q	Y	8	8
1001	9	8-1		EM							i	r	z		I	R	Z	9	9
1010	A	8-2	SMM	CC	SM		¢	!	⑮ :										8-2
1011	B	8-3	VT	CU1	CU2	CU3		$	,	#									8-3
1100	C	8-4	FF	IFS		DC4	<	*	%	@									8-4
1101	D	8-5	CR	IGS	ENQ	NAK	(	)	_	'									8-5
1110	E	8-6	SO	IRS	ACK		+	;	>	=									8-6
1111	F	8-7	SI	IUS	BEL	SUB	|	¬	?	"									8-7
			12				12				12	12		12	12	12		12	
				11				11				11	11	11		11	11	11	Zone Punches
					0				0		0		0	0	0		0	0	
			9	9	9	9									9	9	9	9	

Card Hole Patterns

① 12-0-9-8-1
② 12-11-9-8-1
③ 11-0-9-8-1
④ 12-11-0-9-8-1
⑤ No Punches
⑥ 12
⑦ 11
⑧ 12-11-0
⑨ 12-0
⑩ 11-0
⑪ 0-8-2
⑫ 0
⑬ 0-1
⑭ 11-0-9-1
⑮ 12-11

Control Character Representations

ACK	Acknowledge	EOT	End of Transmission
BEL	Bell	ESC	Escape
BS	Backspace	ETB	End of Transmission Block
BYP	Bypass	ETX	End of Text
CAN	Cancel	FF	Form Feed
CC	Cursor Control	FS	Field Separator
CR	Carriage Return	HT	Horizontal Tab
CU1	Customer Use 1	IFS	Interchange File Separator
CU2	Customer Use 2	IGS	Interchange Group Separator
CU3	Customer Use 3	IL	Idle
DC1	Device Control 1	IRS	Interchange Record Separator
DC2	Device Control 2	IUS	Interchange Unit Separator
DC4	Device Control 4	LC	Lower Case
DEL	Delete	LF	Line Feed
DLE	Data Link Escape	NAK	Negative Acknowledge
DS	Digit Select	NL	New Line
EM	End of Medium	NUL	Null
ENQ	Enquiry		

PF	Punch Off	SO	Shift Out
PN	Punch On	SOH	Start of Heading
RES	Restore	SOS	Start of Significance
RS	Reader Stop	SP	Space
SI	Shift In	STX	Start of Text
SM	Set Mode	SUB	Substitute
SMM	Start of Manual Message	SYN	Synchronous Idle
		TM	Tape Mark
		UC	Upper Case
		VT	Vertical Tab

Special Graphic Characters

¢	Cent Sign	-	Minus Sign, Hyphen
.	Period, Decimal Point	/	Slash
<	Less-than Sign	,	Comma
(	Left Parenthesis	%	Percent
+	Plus Sign	_	Underscore
|	Logical OR	>	Greater-than Sign
&	Ampersand	?	Question Mark
!	Exclamation Point	:	Colon
$	Dollar Sign	#	Number Sign
*	Asterisk	@	At Sign
)	Right Parenthesis	'	Prime, Apostrophe
;	Semicolon	=	Equal Sign
¬	Logical NOT	"	Quotation Mark

This section contains information that is mostly concerned with VS1 system internals and is useful in debugging.

Source Publications

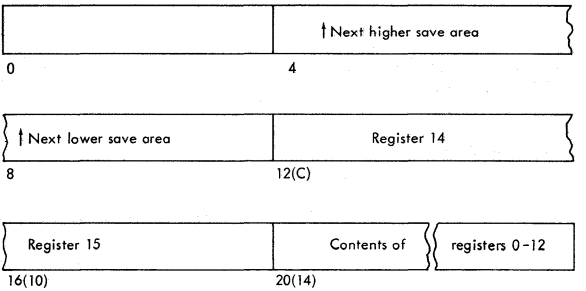
Additional information about linkage registers is in *OS/VS1 Supervisor Services and Macro Instructions*, GC24-5103.

You can obtain additional information about the devices referenced from the publication on the theory of operations or operating procedures. Refer to the *IBM System/370 Bibliography*, GC20-0001 for a list of these publications.

You can obtain additional information about completion codes from *OS/VS Message Library: VS1 System Codes*, GC38-1003.

UCB Sense information for the 3850/3851 is in *Mass Storage System (MSS) Principles of Operation*, GA32-0029.

Save Area Format



Bytes 4-7: Pointer to the next higher level save area or, if this is the highest level save area, zeros.

Bytes 8-11(B): Pointer to the next lower level save area or, if this is the lowest level save area, unused.

Bytes 12-15(C-F): Contents of register 14 (optional)

Bytes 16-19(10-13): Contents of register 15 (optional)

Bytes 20-71(14-3F): Contents of registers 0 to 12

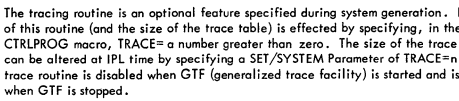
Trace Table

HOW TO USE TRACE TABLE

The tracing routine is an optional feature specified during system generation. Inclusion of this routine (and the size of the trace table) is effected by specifying, in the CTRLPROG macro, TRACE= a number greater than zero. The size of the trace table can be altered at IPL time by specifying a SET/SYSTEM Parameter of TRACE=n. The trace routine is disabled when GTF (generalized trace facility) is started and is enabled when GTF is stopped.

The trace routine places entries, each of which is associated with a certain type of event, into a trace table. When the table is filled, the routine overlays old entries with new entries, beginning at the top of the table (the entry having the lowest address).

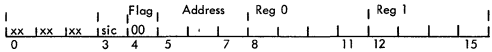
Location X'14' or location X'54' points to pertinent trace table address:



If X'14' is other than X'00', the internal trace table is disabled. (In a stand-alone dump, location X'14' is overlaid. If its content is desired, it should be displayed prior to taking the dump.)

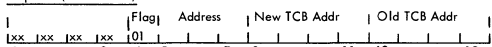
Trace table entries are 16 (decimal) bytes long and represent occurrences of SIO, I/O, SVC, Program Check, and DSP (task switch) interruptions.

SVC Interrupt



Where sic in byte 3 indicates the SVC interrupt code.

Dispatch (Task Switch)



I/O Interrupt

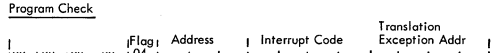
Note that the device address (DDD) spans bytes 3 and 4.

SIO Instruction

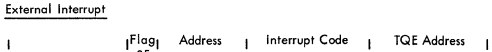
Where cc in byte 0 is the SIO condition code.

Note that the device address (DDD) spans bytes 2 and 3.

Program Check



External Interrupt



Register Usage - How to Find

SYSTEM/370 OPERATING SYSTEM REGISTER USAGE

General Register	Supervisor	IOS	Open/ Close/EOV	Fetch, Link, XCTL, Load
0		@TCB	Work/Par	
1		@RQE	Work/Par	
2		@IOB	DCB/ACB	@Linkor's RB
3	@CVT	@DEB	Base	@CVT
4	@TCB	@DCB	@Work Area	@TCB
5	@RB	Base	@Par List	@SVRB
6	@SVC	Unit Addr	@WTG	@Work
7		@UCB	@Curr Par	Base
8		Base	@Curr WTG	Return
9		Char	@TIOT	Branch
10		Work	@UCB	@Linkee's RB
11		Work		@Work
12		Work/Link		@Linkee's Name
13		Log Ch Wd		
14	@T1 Exit	Link		
15		Appn Base		

Note: Reg 2 does not always point to the DCB/ACB for OPEN/CLOSE/EOV. (Work register for DADSM and CATALOG.)

Symbol	Save Sequence	User
IORGSAV	2-9	IO FLIH
PDSAV	10-1	IO FLIH and Ext FLIH
PISAV	10-9	PC FLIH
SVCSAV	0-15	SVC FLIH
IEAEXSAV	2-9	EXT FLIH

HOW TO FIND

IORGSAV: Location 7D (address portion of I/O new PSW) contains the address of I/O FLIH. The two-byte ADCON of IORGSAV is located 10 bytes from the entry point.

PDSAV IORGSAV + 20 (hex).

PISAV: IORGSAV + 40 (hex).

IEAEXSAV: IORGSAV.

SVCSAV: IORGSAV + 90 (hex).

DSPPSW: (Dispatch Resume PSW) SVCSAV + 40 (hex).

TCBSAV: (Registers saved and restored for dispatch in TCB) Saved Reg. 10 through Reg. 9 in TCB + 30 (hex).

Linkage Register Conventions

Linkage Reg	Register Conventions Use
0	Passes parameters to the control program or the called program. Parameter type depends on macro type.
1	Passes parameters or the address of a parameter list to the control program, or passes parameters to the called program. Parameter type depends on macro type.
2-12	Work registers: not changed by the control program.
13	Passes the address of the register save area provided by the calling program.
14	Passes the return address to the calling program or the control program.
15	Contains the entry-point address, the address of a parameter list as the result of using certain macros, or the return code.

UCB Sense Information

UCB SENSE BYTES

The following abbreviations are used in the chart on the next few pages:

ACC	accelerated
ADV	advance
ALU	arithmetic logic unit
AMT	amount
ATT	attention
BK	back
BKWD	backward
BLK	block
BOC	branch on condition
BR	branch
BUF	buffer
CARR	carriage
CH	channel
CNT	count
CNTR	counter
CNTRL	control
CNTRLR	controller
CNVT	convert
COMP	compare
COMPT	compatible
CORR	correction
CRC	cyclic redundancy check
CU	control unit
DEN	density
DESER	deserialize
DET	detected
DIAG	diagnostic
DISC	disconnect
DOC	document
EC	engineering change
ENG	engineering
ENV	envelope; environmental
EOD	end of data
EOT	end of tape
EQ	equipment
ERR	error
FCB	forms control buffer
FRU	field replaceable unit
FUNC	function
FUT	future
GEN	generation
HDWR	hardware
IBG	interblock gap
IC	instruction counter
ID	identification
INSTR	instruction
INT	intervention
INTRPT	interrupt
INTF	interface
INVLD	invalid
KYBD	keyboard
LRC	longitudinal redundancy check
LRCR	longitudinal redundancy check register
LSR	local storage register
LWR	loop write to read
MAL	malfunction
MP	multiprocessor
MTE	multi-track error
NON-XST	non-existent
OP	operation
OPRATNL	operational
OPR	operator
PRTY	parity
PCU	primary control unit
PE	phase encoded
PERM	permanent
PLB	print line buffer
POSN	position
POSNG	positioning
PROT	protection
RCP	recognition and control processor
RCVY	recovery
RD	read
REC	record
REG	register
REQ	required
REQTD	requested
RESVD	reserved
ROS	read-only storage
RPQ	request for price quotation
RPS	rotational position sensing
RWD	rewind
RWDNG	rewinding
R/W	read/write
SCU	secondary control unit
SEL	selected
SELCTN	selection
SEP	separator
SER	serialize
SPUR	spurious
STKD	stacked
STKR	stacker
SUP	suppression
SW	switch
SYNC	synchronous
TGR	trigger
TI	tape indicate
TM	tape mask
TRK	track
TRNG	turning
TU	tape unit
UCSAR	universal character set address register
UCSB	universal character set buffer
VEL	velocity
VRC	vertical redundancy check
WRT	write
WTM	write tape mark

UCB Sense Information (cont'd)

BYTE 0

BIT / DEVICE	0	1	2	3	4	5	6	7
1052	CMD REJ	INT REQ	BUS OUT	EQ CHK				
1287	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	NON RCVY	KYBD CORR
1288	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	NON RCVY	
1403 1443	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK / TYPE BAR	STR PRTY ERR / TYPE BAR		CH 9
1442, 2501, 2520, 2596	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		
1419/1275 PCU	CMD REJ	INT REQ	BUS OUT		DATA CHK	OVER-RUN	AUTO SELECT	
1419/1275 SCU	CMD REJ	INT REQ	BUS OUT CHK		DATA CHK	OVER-RUN	AUTO SELECT	
2250	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	BUFFER RUN-NING	
2260	CMD REJ	INT REQ	BUS OUT	EQ CHK				
2305	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		
2314, 2319	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	TRK COND CHK	SEEK CHK
2400 series	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD COUNT ZERO	DATA CNVT CHK
2495	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	SHOULD NOT OCCUR	POSN CHK	SHOULD NOT OCCUR
2540	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		UN-USUAL CMD	
2671, 2822	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK			
3203 Model 4	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	BUFFER PARITY CHK	LOAD CHK	CH 9
3210, 3215	CMD REJ	INT REQ		EQ CHK				
3211	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	BUFFER PARITY CHK	LOAD CHK	CH 9
3330/3333 Models 1 and 11; 3350	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		
3850, 3851	Consult Mass Storage System (MSS) Principles of Operation, GA32-0029, for UCB Sense Information for 3850/3851.							

UCB Sense Information (cont'd)

BYTE 0

BIT / DEVICE	0	1	2	3	4	5	6	7
3340	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	TRK COND CHK	SEEK CHK
3410, 3411	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD COUNT ZERO	
3420, 3803	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WORD COUNT ZERO	DATA CNVT CHK
3505, 3525	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		ABNRML FORMAT RESET	PERM ERR (BYPASS KEY)
3540	CMD REJ	INT REQ	BUS OUT CHK	EQ CHK	DATA CHK			
3800	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		LOAD CHK	CH 9
3886	CMD REJ	INT REQ	BUS OUT	EQ CHK			NON INIT	RCP ERR
3890	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		NON INIT	RUN-NING
CHANNEL-TO-CHANNEL ADAPTER		INT REQ	BUFFER DATA CHK	EQ CHK	SELCTN CHK	CTRL SEQ CHK	STATUS GEN CHK	INTF DISC

BYTE 1

BIT / DEVICE	0	1	2	3	4	5	6	7
1287	TAPE MODE	LATE STKR SELECT	NO DOC FOUND		INVLD OP			
1288		END OF PAGE	NO DOC FOUND		INVLD OP			
1419/1275 SCU	FLD 6 VALID	FLD 7 VALID	DOC UNDER W HD	AMT FLD VALID	PRO CTL FLD VALID	ACCT# FLD VALID	TRANSIT FLD VALID	SER# FLD VALID
2250	LIGHT PEN DETECT	END ORDER SEQ	CHAR MODE					
2260								
2305	PERM ERR	INVLD TRK FORMAT	END OF CYL		NO REC FOUND	FILE PROT		OPER-ATION INC
2314, 2319	DATA CHK IN COUNT	TRK OVER-FLOW	END OF CYL	INVLD SEQ	NO REC FOUND	FILE PROT	SERVICE OVER-RUN	OVER-FLOW INL

UCB Sense Information (cont'd)

BYTE 1

BIT / DEVICE	0	1	2	3	4	5	6	7
2400 series	NOISE	00-NON-XST TU 01-NOT READY 10-RDY and NO RWD 11-RDY and RWDING		7 TRK	AT LOAD POINT	WRT STATUS	FILE PROT	TAPE IND
3203 Model 4	CMD RETRY	PRINT CHK		LINE POSN	FORMS CHK	CMD SUP	CNTRLR CHK	
3211	CMD RETRY	PRINT CHK	PRINT QUALITY	LINE POSN	FORMS CHK	CMD SUP	ME-CHAN-ICAL MOTION	
3330/3333, Models 1 and 11	PERM ERR	INVLD TRK FORMAT	END OF CYL	STATE VAR PRES	NO REC FOUND	FILE PROT	WRITE INHIBIT	OPER-ATION INL
3340, 3350	PERM ERR	INVLD TRK FORMAT	END OF CYL		NO REC FOUND	FILE PROT	WRITE INHIBIT	OPER-ATION INC
3410, 3411	NOISE	TU STAT A	TU STAT B		AT LOAD POINT	WRT STAT	FILE PROT	NOT CAPA-BLE
3420, 3803	NOISE	TU STAT A	TU STAT B	7 TRK	AT LOAD POINT	WRT STAT	FILE PROT	NOT CAPA-BLE
3505, 3525	PERM ERR	AUTO RETRY	MOTION MAL	RETRY AFTER INT REQ COMP				
3540	PERM ERR	AUTO RETRY	MOTION MAL	RETRY AFTER INT REQ ~~COMP~~	SPECIAL RECORD TRANS-FERRED			
3800	BIT MEANINGS DEPENDENT UPON BYTE 0. SEE TABLE OF "3800 SENSE BYTES 0-2".							
3886		MARK CHECK	INVLD FORMAT		INCOMP SCAN		NON RECOV-ERY	OUT-BOARD

BYTE 2

BIT / DEVICE	0	1	2	3	4	5	6	7
2250		BUFFER ADDRESS REGISTER BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9
2260		BUFFER ADDRESS REGISTER BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9
2305	BUF LOG FULL	COR-RECT-ABLE						
2314, 2319	UNSAFE		SER/ DESER	TAG LINE	ALU CHK	UNSEL STATUS		
2400 series	BITS 0-7 INDICATE A TRACK IS IN ERROR						6 and 7 INDICATE NO ERROR OR MULTI-ERROR	
3203 Model 4	CARR FAILED TO MOVE	CARR MOTION CHK				FORMS CHK		TRAIN VEL
3211	CARR FAILED TO MOVE	CARR SEQ	CARR STOP	PLATEN FAILED	PLATEN FAILED	FORMS JAM	RIBBON MOTION	TRAIN OVER-LOAD

UCB Sense Information (cont'd)

BYTE 2

BIT / DEVICE	0	1	2	3	4	5	6	7
3330/3333 Models 1 and 11; 3350		CORRECTABLE		ENV DATA PRESENT	COMPATIBILITY MODE			
3340	RPS FEATURE	CORRECTABLE		ENV DATA PRESENT	COMPATIBILITY MODE		MODULE SIZE	MODULE SIZE
3410, 3411	TRACK IN ERROR BITS							
3420, 3803	TRACK IN ERROR BITS							
3505, 3525	USED FOR DIAGNOSTIC PURPOSES ONLY							
3540	USED FOR DIAGNOSTIC PURPOSES ONLY							
3800	BIT MEANINGS DEPENDENT UPON BYTE 0. SEE TABLE OF '3800 SENSE BYTES 0-2'.							

BYTE 3

BIT / DEVICE	0	1	2	3	4	5	6	7
2250, 2260	BUFFER ADDRESS REGISTER							
	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
2305	RESTART COMMAND							
2314	BUSY	ON LINE	UNSAFE	WR CUR GFN	PACK CHNG	END OF CYL	M-MODE SE	SEEK INC
2319	LRC Bit 0	LRC Bit 1	LRC Bit 2	LRC Bit 3				
2400	R/W VRC	LRCR	SKEW	CRC	SKEW REQ	0-1600 1-800	BKWD STATUS	COMPARE
3203 Model 4	UCS PARITY CHK	PLB PARITY CHK		COIL PROT CHK	HAMMER FIRE		SYNC CHK	
3211	USCB PARITY	PLB PARITY	FCB PARITY	COIL PROT CHK	HAMMER FIRE	FIELD ENG	USCAR SYNC CHK	SEP SYNC CHK
3330/3333 Models 1 and 11	RESTART COMMAND							
3340, 3350	RESTART COMMAND							
3410, 3411	R/W VRC	MTE/ LRCR	SKEW	END DATA CHK/CRC	ENV CHK	1600 BPI IN TU	BKWD	
3420, 3803	R/W VRC	MTE/ LRC	SKEW	END DATA CHK/CRC	VRC/ ENV CHK	PHASE ENCODED	BKWD	COMPARE
3505, 3525	USED FOR DIAGNOSTIC PURPOSES ONLY							
3540	TRACK IN ERROR							
3800	PRINTER RDY	PAGE BUFFER EMPTY	BACK DATA CHK	PAPER THREADED FOR SHEET STACKER	SYSTEM RESTART REQ	PHOTO CONDUCTOR ADVANCE ENABLED		

UCB Sense Information (cont'd)

BYTE 4

BIT / DEVICE	0	1	2	3	4	5	6	7
2250, 2260								
2305								
2314					PHYSICAL DRIVE ID			
2319	SEQ IND 0	SEQ IND 1	SEQ IND 2	SEQ IND 3	SEQ IND 4	SEQ IND 5	SEQ IND 6	SEQ IND 7
2400 Series	ECHO ERR	RES TAPE UNIT	READ CLOCK ERR	WRITE CLOCK ERR	DELAY CNTR	SEQ IND C	SEQ IND B	SEQ IND A
3203 Model 4	DEVICE ID	HI-SPEED FAULT				PRINTER SUBSYS-TEM CHK	CNTRLR CHK	STATIS-TICS
3211								
3330/3333 Models 1 and 11	PHYSICAL DRIVE IDENTIFICATION							
3340, 3350	PHYSICAL DRIVE IDENTIFICATION							
3410, 3411	TU POSN CHK	REJ TAPE UNIT	EOT			DIAG TRK CHK	TU CHK	SPARE
3420, 3803	ALU HDWR ERROR	REJ TAPE UNIT	TAPE INDI-CATE	WRITE TRGGR VRC	MICRO-PGM DET ERROR	LWR ERROR	TAPE UNIT CHK	RES RPQ
3540	BINARY ZERO							
3800	DIAGNOSTIC STATUS CODES							

BYTE 5

BIT / DEVICE	0	1	2	3	4	5	6	7
2250, 2260								
2305	CYLINDER ADDRESS							
2314	COMMAND IN PROGRESS WHEN OVERFLOW INCOMPLETE OCCURS							
2319								
2400 Series	COMMAND IN PROGRESS WHEN OVERFLOW INCOMPLETE OCCURS OR ZERO							
3203 Model 4	RIBBON REVER-SAL	RESTORE KEY	RESET LINE COUNTER	SPACE KEY	IDLE STATUS	START KEY	STOP KEY	48 CHAR SET
3211								
3330/3333 Models 1 and 11	CYLINDER ADDRESS							
3340, 3350	CYLINDER ADDRESS							
3410, 3411	NEW SUB-SYSTEM	NEW SUB-SYSTEM	WRT TM CHK	PE ID BURST	PRTY COMP	TACH CHK	FALSE END MARK	RPQ
3420, 3803	NEW SUB-SYSTEM	NEW SUB-SYSTEM	WRT TM CHK	PE ID BURST	START READ CHK	PARTIAL RECORD	XCESSVE PSTAMBL OR TM	RES RPQ
3540	SECTOR IN ERROR							
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

UCB Sense Information (cont'd)

BYTE 6

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	CURRENT HEAD ADDRESS							
3203 Model 4	INVAL FUNC REQ					INVAL I/O CMD	CNTRLR BUS-OUT PARITY	HALT
3330/3333 Model 1	REVERSE	CYL HIGH	DIFFER HIGH	HEAD ADDRESS				
3330/3333 Model 11; 3350		CYL 512	CYL 256	HEAD ADDRESS				
3340	RE-VERSE	CYL HIGH	DIFFER HIGH	HEAD ADDRESS				
3410, 3411		SHRT GAP	DUAL DENSITY	ALT DENSITY	TAPE UNIT MODEL			
3420, 3803	7 TRK	WRT	DUAL	0-1600 1-6250 for Mods 4, 6, 8; 800 BPI for Mods 3, 5, 7	TAPE UNIT MODEL DEFINED			
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 7

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	ENCODED ERROR MESSAGE							
3203 Model 4	PRINT CHK	CAB 2 PARITY	DUCT PARITY	INTRPT OVERRUN	SPUR INTRPT	MISSING BASIC STATUS BIT	FMR PARITY	TIME OUT
3330/3333 Model 1	FORMAT OF REMAINING SENSE BYTES (8-23)				ENCODED ERROR MESSAGE			
3330/3333 Model 11	FORMAT OF REMAINING SENSE BYTES (8-23)				ENCODED ERROR MESSAGE			
3340, 3350	FORMAT OF REMAINING SENSE BYTES (8-23)				ENCODED ERROR MESSAGE			
3410, 3411	LAMP CHK	LEFT COL CHK	RT COL CHK	RESET KEY	DATA SEC ERASE			
3420, 3803	LAMP FAIL	TAPE BOTTOM LEFT	TAPE BOTTOM RIGHT	RESET KEY	DATA SCRTY ERASE	ERASE HEAD FAILED	AIR BRNG PRESS	LOAD FAIL
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 8

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	FAILURE TO MOVE CARR	ACC TOO SLOW	MISSING FB PULSE	EXTRA FB PULSE	STEP CHK 1	STEP CHK 2		CARR CNTRL STOP
3410, 3411		WRT FEED THRU CHK		END VEL CHK	RD BK DATA NOT DET	START VEL CHK		MAR-GINAL VELOC
3240, 3803	IRG DROP IN WRT	FEED THRU CHK	SDR CNTR	EARLY BGN RD BK CHK	EARLY END RD BK CHK	SLOW BGN RD BK CHK	SLOW END RD BK CHK	VELO RETRY/ RESTRT
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

UCB Sense Information (cont'd)

BYTE 9

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	FORMS CHK	OUT OF FORMS	STKR FULL/ JAM	FRONT INTER-LOCK OPEN	ADAPTER COIL PROT	PRINTER INDEP COIL PROT	TRAIN VEL	SYNC CHK 0
3420, 3803	JDR CNTR	VE CHNG ON WRT	SDR COUNTERS					TAPE CTL RESD
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 10

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	FUNC GO RES	MISSING NPL DEV RDY	HAMMER BAR SIGNALS OFF	HAMMER BAR SIGNALS ON	HAMMER BAR SLOW	HAMMER BAR EXCESS BOUNCE	SYNC CHK 1	SYNC CHK 2
3420, 3803	CMD STATUS REJ		CNTRL STATUS REJ	NO BLK ON RCD RD BK CHK	WTM NOT DETECT	TACH START FAIL		VELO-CITY CHK
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

BYTE 11

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	ADAPTER RDY	CARR OP	PRINT OP	INTRPT EXPECT-ED	TIME OUT EXPECT-ED	CHK STOP COND	TRAIN DRIVE OFF	FEED ROLL IDLE
3420, 3803	B BUS PARITY ALU 1		LO ROS/ LO IC PARITY	HI IC BR COND /HI ROS	MCPGM DET HDWR ERR	D BUS PARITY ALU 1		BR COND ALU 1
3800	DIAGNOSTIC ERROR-DEPENDENT DATA							

UCB Sense Information (cont'd)

Byte 12

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	CAC BUSY	ERROR PENDING	GO NOT RDY	ATT PENDING	CHK STOP REQ	CARR CTRL STOP	FUNC REQ STKD	HIGH SPEED CARR
3420, 3803	B BUS PRTY ALU 2		LO ROS/ LO IC ON BR	HI IC BR/HI ROS REG	MCPGM DETECT HDWR ERR	D BUS PARITY ALU 2		BR COND ALU 2
3800	MODULO 256 TRANSFER 2 PPI COUNT							

Byte 13

BIT / DEVICE	0	1	2	3	4	5	6	7
3420, 3803	CONTROL UNIT DENSITY		CONTROL UNIT UNIQUE ID-HIGH ORDER					
3800	MODULO 256 FUSER 8-16 PPI COUNT							

Byte 14

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	PRINT COM-PLETE	NOT READY	TIME STATUS INTRPT	ADAPTER ERROR	OPRATNL	MACH CHK	INTRPT ENABLED	INTRPT REQ'TD
3420, 3803	CONTROL UNIT UNIQUE ID - LOW ORDER							
3800	FUSER PAGE COUNT							

Byte 15

BIT / DEVICE	0	1	2	3	4	5	6	7
3420, 3803	TAPE UNIT UNIQUE ID - HIGH ORDER							
3800	FUSER PAGE COUNT							

UCB Sense Information (cont'd)

Byte 16

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4	ENABLE	COMP	FIRED	HANDLD	DATA CHK	HAMMER FAILURE	HAMMER MISFIRE	PLB PARITY CHK
3420, 3803	TAPE UNIT UNIQUE ID - LOW ORDER							
3800	PAPER COUNT							

Byte 17

BIT / DEVICE	0	1	2	3	4	5	6	7
3420, 3803	2 CHAN SWITCH	CONTROL UNIT WITH DEVICE SWITCH FEATURES			EC LEVEL OF TAPE CONTROL UNIT			
3800	PAPER COUNT							

Byte 18

BIT / DEVICE	0	1	2	3	4	5	6	7
3420, 3803	POWR CHK/ AIRFLO				EC LEVEL OF TAPE UNIT			
3800	SERIAL NUMBER							

Byte 19

BIT / DEVICE	0	1	2	3	4	5	6	7
3203 Model 4		RESTORE KEY	RESET LINE CNTR	SPACE KEY		START/ SINGLE CYCLE KEY		NEAR END OF FORM
3420, 3803	PRIMED FOR DEVICE END TU 7	TU 6	TU 5	TU 4	TU 3	TU 2	TU 1	TU 0
3800	SERIAL NUMBER							

UCB Sense Information (cont'd)

Byte 20

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	PRIMED FOR DEVICE END							
	TU F	TU E	TU D	TU C	TU B	TU A	TU 9	TU 8
3800	PAGE BACKUP COUNT							

Byte 21

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	LOAD BUTTON DEPRESS	LEFT REEL TRNG	RIGHT REEL TRNG	TAPE PRESENT	REELS LOADED	LOAD REWIND	LOAD COM-PLETE	LOAD CHK
3800	PAGE BACKUP COUNT							

Byte 22

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	FRU IDENTIFIERS FOR ALU 1							
3800	PHOTO CONDUCTOR GAP LOCATION OR LOAD CHECK OFFSET							

Byte 23

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	FRU IDENTIFIERS FOR ALU 2							
3800	PHOTO CONDUCTOR GAP LOCATION OR LOAD CHECK OFFSET							

UCB Sense Information (cont'd)

3800

	Bit	Byte 0							
	Bit	0	1	2	3	4	5	6	7
		CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK		LOAD CHK	CH9
Byte 1	0	INVLD CMD	NOT RDY	CMD CODE	HDWR ERR	UNPRNT CHAR		INCORR LENGTH	
	1		OP CHK	DATA BYTE	PERM ERR	(RESVD)		INCORR Multiple of 6 or 8 Lines	
	2		TONER COLLECT-OR FULL		ERR LOG FULL	NO XLATE TABLE		FCB 1/2 INCH ERR	
	3		TONER EMPTY		CANCEL KEY	NO FCB CH CODE		INVLD FCB CH CODES	
	4		DEVELOP-ER RE-PLACE-MENT REQ			MULTIPLE CHARS		FCB VS FORM LENGTH ERR	
	5		END OF FORMS					WCGM NOT LOADED	
	6		OUTPUT FULL					UNASSIGN-ED GRAPHIC CHAR	
	7								
Byte 2	0		FORMS HOLDER INTER-LOCK					INVLD WCGM ID	
	1		TRANS-FER CHK					NO ID FOR WCGM 00	
	2		FUSER CHK					INVLD COPY MODIFY SEQ	
	3		CFS CHK					INVLD FORMS OVERLAY SEQ	
	4		PROCESS CHK					INVLD GRAPHIC MODIFY SEQ	
	5		BURSTER TRIMMER STACKER CHK					WCGM DATA PAR ERR	
	6								
	7		LINE OVER-RUN						

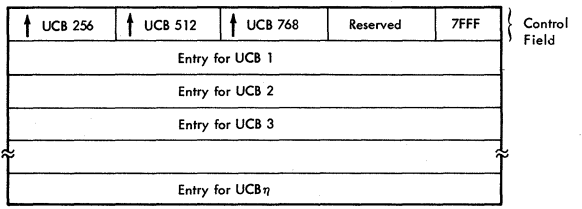
To determine the meaning of bits in bytes 1 and 2, inspect byte 0. Find the column in the table corresponding to the bit in byte 0. The meaning of bits in bytes 1 and 2 will be found in that same column.

Device Statistics Table

The device statistics table contains counters that are used to keep track of the number of times error conditions have occurred on I/O devices. It is used by IBM-supplied error routines and the statistical data records (SDR) of SER. This table is created at system generation and resides in the fixed nucleus. It contains a ten byte control field, used in locating entries to be updated, at the beginning of the table and, with the exceptions noted below, one ten byte entry for each I/O device in a system. The devices UCB points to the entry. The exceptions are:

- 2305 Models 1 and 2 - one common ten-byte entry is pointed to by each UCB in a bank.
- 2314 - A 2314 is considered to be nine devices. Therefore, one common 90-byte entry is pointed to by each 2314 UCB in a bank.
- 2400 tape drives - the entry is 20 bytes if and only if DDR (dynamic device reconfiguration) and a 3400 tape drive are in the system.
- 3330 - one ten-byte dummy entry is pointed to by all 3330 UCBs in a bank.
- 3340 - One ten-byte dummy entry is pointed to by all UCBs in a bank.
- 3410, 3420 tape drives - entry is 20 bytes long; bit 3 in UCBFL5 is on in its UCB.
- 3886 entry is 30 bytes long; bit 3 in UCBFL5 is on in its UCB.

The UCB pointers in the control field are used to determine the section of the table in which the desired entry is located. If the desired entry is in other than section 1, a multiple of 256 (256 for section 2, 512 for section 3, etc.) is added to the STATAB index in the UCB. This is then multiplied by ten and added to the starting address of the statistics table to give the address of the proper entry. When the desired entry is in section 1, the STATAB index itself is multiplied by ten and added to the address of the statistics table. For the 2314, the low-order four bits of the fifth sense byte are also added to the STATAB index to get the correct entry.



Device Statistics Table Entries

2314 Devices

0(0) Temporary Read Failures	Temporary Write Failures	1(1)	Bus-Out Check	2(2) Equipment Check	Overrun	3(3) Track Condition Check	Seek Check
4(4) Unsafe		5(5) Serializer/ Deserializer	Control Unit Tag Line	6(6) Arithmetic Logical Unit		7(7) Missing Address Marker	
8(8) Work Area		9(9) Work Area					

Device Statistics Table (cont'd)

Unit Record Devices

0(0) Temporary Read Failures	Temporary Write Failures	1(1)	Bus-Out Check	2(2) Equipment Check	Overrun	3(3) Device Dependent (Sense Byte 6)	Device Dependent (Sense Byte 7)
4(4)		5(5)		6(6)		7(7)	
8(8) Work Area		9(9) Work Area					

2400 Series and 3400 Magnetic Tape Devices

0(0) Temporary Read Failures	Temporary Write Failures	1(1) Intervention Required	Bus-Out Check	2(2) Equipment Check	Overrun	3 (3) Word Count Zero	Data Converter Check
4(4) Read/ Write Vert. Red. Check	Longitudinal Redundancy Check	5(5) Skew	Cyclic Redundancy Check	6(6) Skew Reg. Vert. Red. Check	Noise	7(7) Read Opposite Recovery	Channel Data Check
8(8) Work Area		9(9) Work Area					

Devices Attached to 2820 Control Units

0(0) Temporary Read Failures	Temporary Write Failures	1(1)	Bus-Out Check	2(2) Equipment Check		3(3) Track Condition Check	
4(4) Track Overrun		5(5)		6(6) No Record Found		7(7)	
8(8) Work Area		9(9) Work Area					

Devices Attached to 2841 Control Units

0(0) Temporary Read Failures	Temporary Write Failures	1(1)	Bus-Out Check	2(2) Equipment Check	Overrun	3(3) Track Conditior Check	Seek Check
4(4) Unsafe		5(5) Serializer/ Deserializer	Control Unit Tag Line	6(6) Arithmetic Logical Unit		7(7) Missing Address Marker	
8(8) Work Area		9(9) Work Area					

Device Statistics Table (cont'd)

3410 Series Magnetic Tape Devices

0(0)		1(1) Noise 1,0		2(2) VRC 3,0		3(3) MTE/LRCR 3,1	
4(4) EDC/CRC 3,3		5(5) Envelope Check 3,4		6(6) Overrun 0,5	Skew 3,2	7(7) Spare 3,7	Spare 4,3
8(8) PE ID CH 5,3 Mask Bit Expansion P		9(9) Track in Error Mask Bits 2, 0-7 0 1 2 3 4 5 6 7		10(A) Write TM Check 5,2	Parity Compare 5,4	11(B) Tach Check 5,5	False End Mark 5,6
12(C) Spare 8,0	Feed-Through Check 8,1	13(D) Spare 8,2	End Velocity Check 8,3	14(E) No Read-back Data 8,4	Start Velocity Check 8,5	15(F) Spare 8,6	Spare 8,7
16(10) Not Used 9,0	Not Used 9,1	17(11) Not Used 9,2	Not Used 9,3	18(12) Backward 3,6		19(13) Bus-Out Check 0,2	Tape Unit Positioning Check 4,0

3420 Series Magnetic Tape Devices

0(0)		1(1) Noise 1,0		2(2) Read Write VRC 3,0		3(3) MTE/LRCR 3,1	
4(4) EDC/CRC 3,3		5(5) Envelope Check/VRC 3,4		6(6) Overrun 0,5	Skew 3,2	7(7) C-Compare 3,7	Write Trigger VRC 4,3
8(8) PE ID CH 5,3 Mask Bit Expansion P		9(9) Track in Error Mask Bits 2, 0-7 0 1 2 3 4 5 6 7		10(A) Write TM Check 5,2	Start Read Check 5,4	11(B) Partial Record 5,5	Excessive Post Amble or TM 5,6
12(C) IBG Drop While Writing 8,0	Feed-Through Check 8,1	13(D) Spare 8,2	Early Begin Readback Check 8,3	14(E) Early End Readback Check 8,4	Slow Begin Readback Check 8,5	15(F) Slow End Readback Check 8,6	Velocity Retry 8,7
16(10) Not Used 9,0	Vel. Change During Write 9,1	17(11) Not Used 9,2	Not Used 9,3	18(12) Backward 3,6		19(13) Bus-Out Check 0,2	ALU Hardware Error 4,0

Device Statistics Table (cont'd)

3540 Input/Output Diskette Unit

0(0) Not Used		1(1) Not Used	Bus-Out Check	2(2) Not Used	No Record Found-CRC Error	3(3) No Record Found-No CRC Error	Fast/Slow Index
4(4) Address Marker Failure	Data Address Marker Incorrect	5(5) Data CRC Error	Control Unit Overrun	6(6) Not Used		7(7) Not Used	Channel Data Check
8(8) Work Area		9(9) Work Area					

3800 Printing Subsystem

0 (0) Temporary Read Failures	Temporary Write Failures	1 (1) Temporary Data Check	Temporary Control Check	2 (2) Temporary Interface Control Check	Refolder Table Check	3 (3)	
4 (4)		5 (5)		6 (6)		7 (7)	
8 (8) Work Area		9 (9) Work Area					

If request is		and data set is	
specific	nonspecific	temporary	nontemporary
X		X	
X			X
	X	X	
	X		X
1)Vol=Ser: 2)Vol=Ref to Another DS in Job Step or to the Catalog Old DS Must Always Use Specific Req	No Vol Serial is Stated or Implied Only for New DS	1)No DSNAME 2)&DSNAME 3)Disp=(New, Delete) 4)Add Card That Refers Back to Any of These	1)Old Data Sets 2)Disp Keep or CATLG

then it can be satisfied with a volume that is:							
Permanently Resident			Reserved			Removable	
Public	Private	Storage	Public	Private	Storage	Public	Private
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X		X	X		X	X	
		X			X	(See Note)	
These Volumes Are Never Available for Dismounting by the System. 1) Nondismountable (2305) 2) IPL Vol 3) LINKLIB, PROCLIB, JOBQE, page, or SPOOL Volumes 4) Via PRESRES			Not Available for Dismounting until an Unload CMD is Issued. 1) Via PRESRES 2) Via a MOUNT CMD			These Volumes Are at the Disposition of the System. 1) All Other DA Volumes	

Note: This type of request is satisfied by a public removable volume that is made private.

ALLOCATION CHARACTERISTICS

A STORAGE volume is:

Designated in PRESRES.
A volume for which the mount command has been given with a USE parameter of STORAGE (i.e., MOUNT 131, USE=STORAGE).

A PRIVATE volume:

Designated in PRESRES.
Requested with the PRIVATE subparameter specified, and the volume is removable. Was requested nonspecifically for a nontemporary data set and the request had to be satisfied with a removable volume.
A volume for which the mount command has been given with a USE parameter of PRIVATE (i.e., MOUNT 131, USE=PRIVATE (default)).

A PUBLIC volume is:

Designated in PRESRES.
A removable volume that has not been made PRIVATE.
A volume for which the mount command has been issued with a USE parameter of PUBLIC (i.e., MOUNT 131, USE=PUBLIC).

Completion Code Summary

NOTE: This summary is sequentially ordered by the 2nd and 3rd digits of the completion code. To quickly find the explanation for a given code, say 03D, first locate the section for _3D.

Group	Completion Code	Operation of Macro Instruction	Explanation
Supervisor Call (SVC nn)	Fnn		Invalid operand, nn, in SVC instruction.
EXCP (SVC 00)	100	I/O operation	Device not operational.
	200	I/O operation	Invalid SCB, IOB, DCB protect key.
	300	I/O operation	Invalid DEB protect key.
	400	I/O operation	Invalid DCB pointers.
	500	I/O operation	Invalid UCB address.
	600	I/O operation	Requested by subsystem (EXCPVR) and JSCB is missing or susbsystem bit in JSCB is not on.
	700	I/O operation	No SQA (system queue area) available for the request or the system lock was set.
	800	I/O operation	Invalid address of control block, appendage, CCW, or CCW data field was found; page fix request too large.
	900	I/O operation	More than 240 CCWs specified for translation.
	A00	I/O operation	CCWs modified in a PCI appendage were changed to cause the CCW translator to need to fix a page. I/O supervisor encountered a page fix request in PCI.
	B00	I/O operation	Overfix threshold exceeded.
	C00	I/O operation	Invalid number of entries in the fix list was found upon return to IOS from the user fix appendage.
Misc.	D00		Error occurred during processing of an SIO extended request.
BDAM/ BISAM/ BSAM	001	CHECK	I/O error encountered and either no SYNAD address specified in DCB, or
QSAM		GET,PUT	terminate option in DCBEROPT of DCB.
WAIT (SVC 01)	101	WAIT	More events than ECBs.
	201	WAIT	Invalid ECB address.
	301	WAIT	ECB wait flag already on.
BSAM/ QSAM/ QISAM/ ISAM	002		Record is greater than 32,768 bytes; exceeds maximum track length or stated blocksize; could not be contained in one extent; too many tracks specified for cylinder overflow; or BDW or RDW (SDW) invalid.
POST (SVC 02)	102	POST	Invalid ECB address.
	202	POST	Invalid RB address in ECB.
	402	POST	Invalid EVENTS table address.
	502	POST	EVENTS table specified in ECB had no available entries to contain posted ECB address.
BSAM/ QSAM	003	EOB for 3525	3525 associated data set I/O macro sequence error.
Task termination (SVC 03)	103	RETURN or branch to return address in register 14	ECB already posted or RB address in ECB invalid.
	A03	RETURN or branch to return address in register 14	Subtasks not yet terminated.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
Task termination (SVC 03) (continued)	C03	RETURN or branch to return address in register 14	TCBDEB points to DEB that is associated with an invalid DCB. WARNING: all data sets not closed.
	D03	RETURN or branch to return address in register 14	ENQ resources not released yet.
BSAM/ QSAM	004	OPEN for 3525/ 3505	Invalid FORMAT card or invalid device specified with OMR; conflicting or invalid DCB parameter; data protection image not found in SYS1.IMAGELIB.
GETMAIN (SVC 04)	604	GETMAIN	Address in A or LA operand is outside task.
	704	GETMAIN	List request not valid for VS1.
	804	GETMAIN or language processor	Request for zero bytes of virtual storage or not enough virtual storage available.
	A04	GETMAIN	Storage management queues have been altered incorrectly. Re-IPL the system.
	B04	GETMAIN	Subpool number greater than 127.
	E04	GETMAIN from program in supervisor mode	Not enough storage available in fixed or pageable SQA.
BSAM	005	READ for 3886	Invalid DECB.
FREEMAIN (SVC 05)	605	FREEMAIN	Address in A or LA operand is outside task.
	705	FREEMAIN	List request not valid for VS1.
	905	FREEMAIN	Address of area to be released not multiple of 8.
	A05	FREEMAIN	Area to be released overlaps existing free area.
	B05	FREEMAIN	Subpool number greater than 127.
Contents supervisor (SVC 06)	106	LINK, LOAD, ATTACH, XCTL	Error while loading module into virtual storage: invalid record type, invalid address, I/O error.
	406	LINK, ATTACH, XCTL	Module was only loadable.
	506	LINK, LOAD, ATTACH, XCTL	Not enough virtual storage for module and overlay supervisor.
	606	LINK, LOAD, ATTACH	Not enough virtual storage for module.
	706	LINK, LOAD, ATTACH, XCTL	Module marked "NOT EXECUTABLE".
	806	LINK, LOAD, ATTACH, XCTL	BLDL detected error: module not found, I/O error during directory search, or JOBLIB DCB not open.
	906	LINK	More than 255 tasks waiting for reentrable or serially reusable module.
	A06	LINK, LOAD, ATTACH, XCTL	Task already waiting for serially reusable module.
	B06	I/O activity	Abnormally terminating system error task reinstated; user task abnormally terminated.
	C06	XCTL	Abnormally terminating transient area task reinstated; user task abnormally terminated.
XCTL (SVC 07)	207	XCTL	An asynchronous exit routine attempted to execute XCTL.
BSAM	008	CHECK while creating data set	SYNAD returned to CHECK routine after save area was destroyed.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
GETMAIN, FREEMAIN with R operand (SVC 0A)	10A	GETMAIN for program in supervisor mode	Not enough available storage in local system queue space (VS2 only).
	20A	Getting, freeing, or replacing region for new job step	Storage still allocated to previous step; new step (VS2 only).
	30A	FREEMAIN	Area to be released not within correct subpool or not described by DQE (VS2 only).
	40A	FREEMAIN	Attempt to release all of subpool zero (VS2 only).
	60A	FREEMAIN	Invalid specification of an area to be freed.
	80A	GETMAIN	Request for zero bytes of virtual storage or not enough virtual storage available.
	90A	GETMAIN, FREEMAIN	Address of area to be released not a multiple of 8.
	A0A	GETMAIN, FREEMAIN	Area to be released overlaps an existing free area.
	B0A	GETMAIN, FREEMAIN	Subpool number greater than 127.
	D0A	FREEMAIN with R operand	Attempt to free system queue area storage not owned by task (VS2 only).
ABEND (SVC 0D)	D0D	ABEND	Invalid ABEND recursion during abnormal termination of subtask; job step terminated.
	E0D	ABEND	Insufficient virtual storage available for ABEND processing of subtask, job step terminated.
OPEN (SVC 13)	013	OPEN	Conflicting or unsupported parameters in DCB; member name specified in DD not found; no directory allocation subparameter in DD.
	113	OPEN, OPEN with TYPE=J	I/O error reading or writing JFCB or in reading JFCB extension block; no exit code provided.
	213	OPEN	INPUT, INOUT, or RDBACK specified but no serial number for SER in DD; I/O error reading volume label, could not mount volume on device; more devices allocated than volumes.
	313	OPEN	I/O error in reading format 2 or 3 DSCB.
	413	OPEN	INPUT specified but no serial number for SER in DD; I/O error in tape positioning or label processing; could not mount volume on device; more devices allocated than volumes.
	513	OPEN	Attempting to open second DCB for same tape volume.
	613	OPEN	I/O error in label processing or tape positioning.
	713	OPEN	Expiration date not reached, but data set opened for output and DD contained MOD in DISP.
	813	OPEN	Verification error in label processing.
	913	Supplying password	Incorrect password entered twice; third byte of 80-byte record for data set not used to indicate read, write, or read and write; ASCII tape accessibility error, ASCII tape security error.
	A13	OPEN	File sequence number in LABEL on DD not on volume.
	B13	OPEN for UCS printer	Operator canceled UCS load or permanent I/O error detected.
	C13	OPEN	I/O error in reading JFCB or DSCB for concatenated data set; JFCB or DSCB not found for one data set in concatenation; graphic device already opened by another task.
	D13	OPEN for graphics	Graphic device not opened by closing task.
	E13	OPEN for graphics	DCBGNCP field not 1 through 99.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
CLOSE (SVC 14)	214	CLOSE	I/O error in tape positioning or volume disposition.
	314	CLOSE	I/O error reading DXCB.
	414	CLOSE	I/O error writing updated DSCB.
	514	CLOSE	I/O error reading JFCB.
	614	CLOSE	I/O error writing file mark.
	714	CLOSE	I/O error processing label.
	A14	CLOSE	I/O error releasing unused D/A space.
	B14	CLOSE	STOW unable to store, modify, or delete data from partitioned data set directory because name already in directory, no space available in directory, or I/O error searching directory.
	D14	CLOSE for graphics	Graphics device already opened by another task.
TCLOSE	117	BSAM CLOSE with TYPE=T	I/O error positioning tape or writing file mark.
	217	BSAM CLOSE with TYPE=T	I/O error reading JFCB.
	317	BSAM CLOSE with TYPE=T	I/O error reading DSCB.
	417	BSAM CLOSE with TYPE=T	I/O error writing updated DSCB.
	717	BSAM CLOSE with TYPE=T	I/O error processing label.
BDAM	020	OPEN	Invalid DCBMACRF field.
Master Scheduler (SVC 22)	122		Operator canceled job; requested dump.
	222		Operator canceled job; did not request dump.
	322		Execution of job step or cataloged procedure taking longer than time specified.
	422		Job required too much queue space for initiation.
	522		All tasks in SVC wait state for time specified by WAIT parameter of CTRLPROG macro or for time specified in JWT parameter (in systems with SMF).
Misc	722		OUTLIM keyword specified on SYSOUT DD statement exceeded.
	822		V=R area not obtained.
WTO/ WTOR (SVC 23)	D23	WTO, WTOR	Parameter list does not begin on proper boundary, text length zero or less, or no buffers available.
	E23	Reply command processing (in response to WTOR)	Invalid ECB address, RB address, or reply address.
BDAM	025		Address in DCBSQND field outside task.
	026	Processing with exclusive control	Invalid DCBXARG field or exclusive control status not indicated.
Paging	028		Page file I/O error.
EXTRACT (SVC 28)	128	EXTRACT	Output list not on fullword boundary or not contained in storage assigned to job step.
	228	EXTRACT	Input parameter list not on fullword boundary or does not begin in storage assigned to job step.
	328	EXTRACT	TCB not for immediate subtask.
ATTACH (SVC 2A)	42A	ATTACH	Address for ECB to be posted upon subtask termination is not multiple of 4, higher than highest virtual storage, or does not have same protect key as attaching task.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
CHAP (SVC 2C)	12C	CHAP	TCB address (for subtask) does not point to valid TCB or TCB of immediate subtask, is not a multiple of 4, or points to a task that has terminated.
	22C	CHAP	TCB address (for subtask) not a multiple of 4, higher than highest virtual storage, or does not have same protect key as CHAP issuer.
Overlay supervisor (SVC 2D)	12D		Words 3 and 4 of segment table invalid.
	22D		Address in segment table or entry table outside storage for job step.
	32D		Wrong length record or I/O error when loading segment.
	C2D		Invalid scatter record found while loading program segment.
	D2D		Invalid record type found while loading program segment.
	E2D		Invalid address found while loading program segment.
BISAM/ QISAM	030	OPEN	Invalid DCBMACRF field.
DEQ (SVC 30)	130	DEQ	DEQ for resource not enqueued by prior ENQ, DEQ RET =HAVE operand missing.
	230	DEQ	Invalid length specified for name of resource.
	330	DEQ	Supervisor state option specified in problem program state.
	430	DEQ	Invalid parameter list.
	530	DEQ	Task does not yet control specified resource.
BISAM/ QISAM	031	OPEN	QISAM I/O error; no SYNAD specified.
	032	OPEN	Invalid DCBMACRF field for BISAM.
	033	OPEN	I/O error reading highest level index or validating last record pointers, or address in DCBMSHI field outside task or under incorrect protection key.
	034	OPEN	DCBSMSI field specifies area too small for highest level index; invalid address in DCBMSWA (BISAM).
	035	OPEN	DCBSMSW field specifies area too small for one track for BISAM.
	036	OPEN	No prime area specified.
	037	OPEN	User supplied buffers too small.
EOV (SVC 37)	137	End of volume	I/O error in label processing.
	237	End of volume	Verification error in label processing; tape label block count not same as DCB block count; interrupt during command chaining.
	337	End of data set	No address specified in DCBEODAD field.
	437	End of volume	Protect key different in TCBPKF field of TCB than in DEBDEBID field of DEB.
	537	End of volume for multiple volumes	Specified volume being used for another data set.
	637	End of volume	I/O error writing tape mark, positioning tape, reading label, sensing for file protect ring; DCB bit does not indicate concatenation of unlike attributes.
	737	End of volume or allocation of secondary quantity	DA I/O error; DSCB not found for multivolume or concatenated data set.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
EOV (SVC 37) (continued)	837	End of volume for sequential data set	I/O error reading or writing JFCB from or onto direct access; JFCB extension needed but not found.
	A37	End of volume	DCB not open; DCB not pointing to a proper DEB.
	B37	End of volume	Volumes must be demounted from a device allocated to the data set, but system unable to demount volume.
	D37	Output operation	More space needed but no secondary quantity specified to SPACE in DD.
	E37	Output operation	More space needed but no more volumes specified in SER, volume count, or REF in DD; more space needed but unavailable in partitioned data set, additional space would require another volume or 17 extents.
QISAM	038	OPEN	Index area too small or crosses volumes.
ENQ (SVC 38)	138	ENQ without RET = TEST, USE or HAVE	Second ENQ without intervening DEQ.
	238	ENQ	Invalid length for resource name.
	338	ENQ	Set - must - complete (SMC) or directed ENQ specified in problem program state.
	438	ENQ	Invalid parameter list.
QISAM	039	Scanning	End of data set; no address in DCBEODAD field.
	03A	CLOSE	I/O error writing updated format 2 DSCB.
	03B	OPEN	ISAM data set to be processed, but not created or its DCB not closed after creation.
QISAM/ BISAM/ BDAM	03D	OPEN	Indexed sequential not specified, or serial numbers for SER in DD not in order or not all present.
QISAM	03E	OPEN	No space available for resume loading.
DETACH (SVC 3E)	13E	DETACH	Subtask being detached not yet terminated.
	23E	DETACH	Address of subtask TCB is not multiple of 4, is higher than highest virtual storage, does not have same protect key as DETACH issuer; address specified not valid TCB or TCB of immediate subtask.
CHKPT	13F		Error during execution of checkpoint restart.
TCAM	040	OPEN	Error in opening a TCAM line group data set or establishing VTAM interface.
RDJFCB	140	RDJFCB	I/O error reading JFCB.
	240	RDJFCB	(1) No foundation extension block in DCB. (2) No EXLST address in DCB. (3) No JFCB exit in DCB exit list. (4) JFCB buffer not in user's virtual storage.
TCAM	041	OPEN	Error in opening a TCAM message queues data set.
	042	Processing	Error in running a TCAM MCP with the telecommunication on-line test executive.
	043	OPEN	Error in opening a TCAM application program data set.
	044	Processing	Error in processing the FE common write subtask.
	045	Message Control Program (MCP)	I/O error or logical read error.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
TCAM (continued)	046	CLOSE	TCAM MCP is scheduled to be terminated, application program data set active. Completion code is for the applications program data set.
SWAP (SVC 55)	155		SVC 85 (in decimal) issued by user's task, but is restricted for use by Dynamic Device Reconfiguration.
Graphics access method (GAM)	056	Graphics attention service routine	ANALYZ or GSERV specified DCB, which pointed to DEB, which pointed to invalid UCB.
	057	Graphics Attention service routine	ANALYZ or GSERV specified DCB, which pointed to DEB, which pointed to UCB for other than graphics device.
	061	CLOSE	CLOSE issued DAR to GACB that was not specified (via SPAR) by the closing task.
	062	Graphics Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I	Return code equal to or greater than absolute value of null argument produced.
	063		2250 operator depressed alphameric keyboard CANCEL key and used lightpen to terminate program.
ESR (SVCs 6D, 74, 7A)	16D		Invalid ESR code in register 15.
Supervisor (SVC 75)	16E	DEBCHK	Control program requested a DEBCHK function be performed on a data extent block (DEB) obtained from the DCB passed by the program. Function could not be completed.
EVENTS (SVC 7A)	17A	Delete or create EVENTS table	Invalid EVENTS table address.
	27A		EVENTS table not created.
	37A		System waiting for specified EVENTS table.
	47A		Specified number of entries not 1 to 32, 767.
	57A		Non-job-step task or unauthorized user attempted to use restricted EVENTS option.
EVENTS (SVC 7D)	17D	Process completed events	Invalid EVENTS table address.
	27D		EVENTS table not created.
	37D		System waiting for specified EVENTS table.
	47D		Specified number of entries not 1 to 32,767.
	57D		Invalid ECB address or nonmatching keys.
	67D		ECB already waiting when initialization as such specified.
	77D		Invalid parameters in registers 1 and 0.
BTAM	090	OPEN	UCB for other than communications device.
	091	OPEN	UCB specified invalid or unsupported transmission control unit.
	092	OPEN	UCB specified invalid or unsupported terminal control or adapter.
	093	OPEN	UCB specified invalid or unsupported terminal.
	094	OPEN	UCB specified invalid or unsupported optional feature or mode of operation.
	095	OPEN	Line group did not have identical terminal types and/or optional features.
	096	OPEN	DCBBFTECK field specified dynamic buffer allocation, but DCBBUFCE, DCBBUFNO, and DCBBUFI fields not specified.
	097	OPEN	Device I/O directory full.

Completion Code Summary (cont'd)

Group	Completion Code	Operation of Macro Instruction	Explanation
BTAM (continued)	098	OPEN	Transmission control unit not a 2701 or the Dual Communication interface, or Dual Code Feature not specified in UCB.
VTAM	0A8 0A9	Application program A VTAM module	Invalid user's RPL or ECB. Finished processing while holding a lock; invalid DVT; request for (or release of) storage failed; returned control to user's STAE exit routine.
Job Scheduler	0B0 1B0 0B6		I/O error in reading or writing SYS1.SYSJOBQE, SYS1.SYSWADS, or SWADS. Invalid TTR for SYS1.SYSJOBQE found by system conversion routine. System logic error while accessing resident job list or SYS1.SYSJOBQE.
Prologue	0Cx		Program interruption, not in I/O interruption handler or type 1 SVC routine; no program routine to handle interruption; x=program interruption code.
Paging	0D0 0D1		Invalid segment translation interrupt. Invalid page-fault interrupt.
Prologue	0D3 0F1 0F2 0F3		Program interrupt caused by invalid set system mask (SSM) instruction. Program interrupt in I/O interrupt handler. Program interrupt in type 1 SVC routine. Machine-check interrupt; MCH able to abnormally terminate job step and continue operating system.
System Restart	2F3		Job was being executed when system failure occurred; a system restart was performed.
Prologue	0F5		Program interrupt occurred while loading transient area for type 3 or 4 SVC.
RTAM	1F9	MODIFY command	A writer task was abended when: • it had an outstanding reply, and • a line communication logically disconnected.
Misc	3FE 4FE		Task attempted to end normally but teleprocessing I/O requests were active or pending. Task attempted to end normally but nontele-processing I/O requests were pending and no teleprocessing I/O requests were outstanding.

Wait State Codes

001 IPL/NIP: Not operational in response to a test I/O instruction (reg. 10=unit address).

002 IPL/NIP: I/O operation not initiated, CSW stored, and channel not busy (reg. 10=unit address).

003 IPL/NIP: I/O operation not initiated, CSW not stored, and channel not busy; or SYS1.LINKLIB not cataloged, no UCB found for IPL device, unexpected "not operational" condition found (reg. 10=unit address).

004 IPL/NIP: I/O operation not initiated, CSW not stored, channel not busy (reg. 10=unit address).

005 IPL/NIP: I/O interrupt because of unit check (if IPL, fourth byte of PSW= X'00'; X'4C'=address of CCW causing check; X'54'=sense bytes describing check. Reg. 10=unit address).

006 IPL/NIP: interface control check, channel control check, channel data check, program check, or channel chaining check occurred.

007 NIP: Console not available.

008 NIP: I/O interruption because of unit check. Record not found, unformatted direct access volume mounted on online device, or volume label on an alternate track (reg. 10=unit address).

009 NIP: I/O interruption because of unit check. File mask violation (reg 10= unit address).

00A SYS1.LINKLIB not found in catalog

00F IPL: Volume not containing IPL text used for IPL.

010 NIP: I/O interruption because of unit check. End of cylinder (reg 10=unit address).

011 NIP: I/O interruption because of unit check. Track condition check occurred (reg. 10=unit address).

012 NIP: System generated with ECPS but ECPS hardware not present.

013 NIP: Recovery not possible. Check system completion code.

017 IPL/NIP: Unit check while executing sense instruction (reg. 10=unit address).

018 IPL: Nucleus too big for machine size. Space for RLD records exceeded.

019 IPL: Program interruption because of hardware errors or SYS1.NUCLEUS occupying more than one extent.

021 NIP: I/O interruption on teleprocessing or graphic console (reg. 1=pointer to IOB for failing EXCP operation).

026 IPL: PWF support successfully refreshed real storage from a warn data set, or after a utility power disruption, PWF support successfully transferred real storage to a warn data set.

027 IPL: PWF support was unable to refresh real storage from a warn data set, or after a utility power disruption, PWF support was unable to transfer real storage to a warn data set.

0E2 NIP: Machine check interruption occurred before machine check handler initialized.

0F1 DDS: Error processing encountered error preventing reinstatement of VS processing (Message IQA016W).

0FA DSS: Translation specification exception (Message IQA010W).

900 NIP: Error occurred during initialization of one or more page data sets (Messages IEA750W, IEA751W, IEA752W, IEA753W, IEA754W, IEA755W).

901 SUPVR: Channel program check during paging I/O operation.

902 SUPVR: Uncorrectable I/O error while pageable supervisor was being read into real storage from SYS1.PAGE data set.

903 SUPVR: Page supervisor ended abnormally.

904 NIP: Unable to complete system initialization.

905 IPL: CPU model number obtained by a Store CPUID instruction not found in IPL list of supported models.

906 IPL: Machine check interruption because of either a malfunction other than a storage error or uncontrollable real storage error in first 256K of storage.

Wait State Codes (cont'd)

907 NIP: Link or load failed for an essential module (Message IEA782W).
908 NIP: System generated with extended timer support but clock comparator and CPU timer not supported by hardware.
A01 RMS: Error occurred while performing recovery. Probable machine check on machine check. (Message IGF910W).
A02 RMS: Error occurred while performing a recovery. Probable machine check on machine check.
A03 RMS: Error occurred while performing a recovery. Probable program check on machine check (Message IGF910W).
A04 MCH: I/O error during machine check recovery (Message IFG930W).
A05 MCH: Unrecoverable failure within supervisor area (Message IGF900W).
A0A MCH: Encountered failure that channel check handler could not correct.
A0C MCH: Unable to load a page on a low end system (Message IGF930W).
A11 RMS: Error occurred while performing a recovery. Probable invalid machine check interrupt code (Message IGF910W).
A16 MCH: Failure in time of day clock, clock comparator, or CPU timer (Message IGF950W).
A17 MCH: Failure in interval timer (Message IGF950W).
B01 3211 Utility: Completed normally.
B02 3211 Utility: Control card missing or out of order.
B03 3211 Utility: JOB statement is incorrect.
B04 3211 Utility: DFN statement is incorrect.
B05 3211 Utility: UCS statement is incorrect.
B06 3211 Utility: FCB statement is incorrect.
B07 3211 Utility: END statement is incorrect.
B0A 3211 Utility: External interrupt has occurred. Interrupt key was pressed.
B0B 3211 Utility: Program check interrupt has occurred.
B0C 3211 Utility: Machine check interrupt has occurred.
B11 3211 Utility: Reader is not online.
B12 3211 Utility: Reader is not ready.
B13 3211 Utility: Reader is not ready.
B14 3211 Utility: Reader channel error has occurred.
B15 3211 Utility: No device end is indicated on the reader.
B19 3211 Utility: Printer is not online.
B1B 3211 Utility: Unit check has occurred on the printer.
B1C 3211 Utility: Printer channel error has occurred.
B1D 3211 Utility: No device end is indicated on the printer.
D01 SUPVR: ABTERM or PROLOG failed while processing the terminating program.
E02 CONSOLE: Permanent I/O error on 2250 display unit. No alternate console was available.
E04 SUPVR: SQA had less than 288 bytes. GETMAIN request was issued for more SQA space than was available.

System ENQ/DEQ Names

Major	Minor	Resource Serialized
SYSABEND	DUMP	ABEND dump dataset.
SYSCTLG	SYSCTLG	Catalog management.
SYSDSN	dsname	Non-temporary dsnames (Dataset Integrity).
SYSDSNI	dsname	Non-temporary dsnames for initiators.
SYSIEA01	IEA	User SNAP dump dataset.
SYSIEC16	x'F0'	DEB chain for purge.
SYSIEECT	IEEWQE IEERQE	WQE when all console buffers are full. ORE when the number of outstanding reply requests is at the system limit.
SYSIEFDR	JOBQ TTR of DER	Disk entry record (DER).
SYSIEFPL	SPOOL job number	SPOOL allocation.
SYSIEFSD	BMRSCH	Buffer User Table (SPOOL).
	BUF	SMF buffer.
	CMCM	Checkpoint MCM (SPOOL).
	DS	Dataset Integrity process.
	INNWE	Input No-Work-Element overflow chain (Job List).
	IOBF	Work Area Allocation buffer (SPOOL).
	JEPS2	Initiator/JEPS parameter area.
	JOBNO	SPOOL job number.
	JSCBWTP	System Message dataset for multitasked job (Write-to-Programmer).
	MT	Mount TRCB chain.
	OUTNWE	Output No-Work-Element overflow chain (Job List).
	PARTBNDS	Checkpoint/Restart partition boundaries.
	Q1	Job List.
	Q2	Master QCR (System Swads).
	Q3	No-space ECB (System Swads).
	Q4	I/O device allocation.
	Q5	UCBs (Scheduler fields).
	Q7	Allocation for system starts.
	Q8	I/O device allocation mount verification interface.
	Q12	Job Queue VCBs.
	Q13	No-space ECB (Job List).
	Q14	No-space ECB (Job Queue).
	RMCM	Resident MCM (SPOOL).
	SR	System TRCB chain.
	WD	Accounting dataset.
SYSIEFSM	SMRSCE	SPOOL User Table.
	SMRSCT	Lock ECB for SPOOL DSD use counter (JES writer request).
SYSIEFUM	ISSPAREA	Request for ISSP if exit definition tables are not defined.
SYSIEWL	(dsname for SYSLMOD)	Linkage editing.
SYSIGGLG	MBBGCHHR	R0 (capacity record) for BDAM.
SYSPSWRD	PASSWORD	Security dataset for OPEN/EOV.
SYSSMF01	BUF	Common SMF record buffer.
SYSSMFDP	DATASET	SYS1.MAN dataset during dump processing.
SYSVTOC	vol-ser	Volume Table of Contents.

How To Find Associated Logical Channel Word

CVT PTR (+140) to LCW Table

$$+8 \begin{pmatrix} \text{UCB+10} \\ \text{LCH TAB} \end{pmatrix} = \text{LCW}$$

LOGICAL CHANNEL WORD TABLE

The logical channel word table consists of the logical channel words that control the logical channel queues. It is used by the I/O supervisor and the I/O purge and SVC purge routines.

The logical channel word table has the following characteristics:

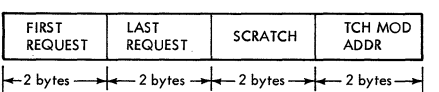
1. Creation: The table is created at system generation time.

2. Storage Area: The table resides, as a permanent part of the fixed nucleus.

3. Size: The table contains one 8-byte logical channel word per logical channel queue.

4. Means of Access: Find the start of the LCW table in CVT + 8C; add to this pointer the value of the LCHTAB byte in the UCB (UCB + A) multiplied by eight.

The format of a logical channel word is:



FIRST REQUEST (2 bytes)
These two bytes contain either an address or an index value to the first request element in the logical channel queue.

LAST REQUEST (2 bytes)
These two bytes contain either an address or an index value to the last request element in the logical channel queue.

SCRATCH (2 bytes)
This field is used as a temporary storage area for an address or index value. The field is used when more than one logical channel queue for a physical channel is searched in order to find the highest priority I/O request with which to restart the channel.

TCH MOD ADDR (2 bytes)
This field addresses the device-dependent test channel module.

Notes:

1. When a logical channel queue is void, the FIRST REQUEST field contains a dummy link address of hex FFFF and the LAST REQUEST field contains the address of that logical channel word.

2. When there is only one request element in the queue, both FIRST REQUEST and LAST REQUEST contain the address of that element.

How To Find the Entry Point of Types I and II SVCs

A. Pick up the instruction address from the SVC New PSW at X'60'.

B. Starting at the location in Step A, search for the first L instruction (58 80 0XXX).

C. The pointer to the SVC table prefix is at address XXX (step B).

D. The pointer to the SVRB create routine in IEATA00 is at XXX+4.
The pointer to the SVC SLIH is at XXX+8. The SVC table is at XXX+x'100'.

E. Add the SVC number (hex) to the address of the SVC prefix table.

F. Pick up the byte value pointed to by the result of step E.

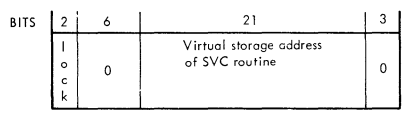
G. Multiply the value picked up in step F by 4.

H. Add the results of step G to the address of the SVC table determined in Step C.

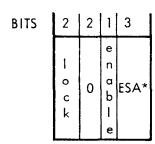
I. The value computed in step H points to a 4-byte address constant which is the SVC routine entry point.

Note that all entry points are on a doubleword boundary. The three low-order bits in the entry address are not part of the actual address. Bits 5 and 6 equal the number of doublewords in register save area in SVRB. Bit 7 indicates an SVRB is needed.

SVC TABLE FORMAT

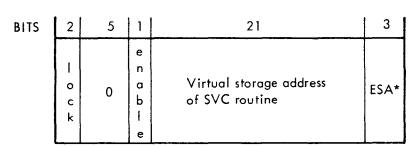


4-byte entry for type 1 SVC routines



1-byte entry for type 3 and 4 SVC routines

* ESA is the number of doublewords required for the extended save area in the request block.



4-byte entry for type 2 SVC routines

How To Find (cont'd)

SVC Table Format (cont'd)

BITS	10	8	12	2	12	1	3
	Track Address	Record Number	Length of first text record	lock	Attributes of SVC routine	enable	ESA*

6-byte entry for transient type 3 and 4 SVC routines

BITS	8	22	2	12	1	3
	X'FF'	Address of SVC routine	lock	0	enable	ESA*

6-byte entry for fixed type 3 and 4 SVC routines

* ESA is the number of doublewords required for the extended save area in the request block.

How to Find Resident Build List

A. Pick up CVT pointer in location X'10'.

B. Add X'20' to this pointer. This is CVTPRLTV pointer.

C. Locate CVTPRLTV entry-8.

D. This is pointer to resident build list.

E. Format of resident build list:

0-1 number of entries
2-3 length of each entry
4-43 entry number 1
44-? entry number 2, etc.

Normal length of each entry is 40 bytes.

How to Find Resident SVC Load List and RAM List

The resident SVC load list and RAM list pointers are two fullwords located before the constant IGG019 IFG019. The pointers are known as IEAARSV1 and IEAARAM4, respectively, and may be located from these names in LMODMAP. These constants are defined in IGC007, IGC008, or LINK, XCTL, and LOAD code if a system generation listing is available.

VS1 STORAGE SUBPOOLS

A subpool is a group of logically related storage blocks identified by a subpool number and assigned through a GETMAIN instruction. The subpool number indicates the type of storage that is requested. The meaning of each subpool number is listed below.

Subpool Number	Attributes	GQE Built[1]	Control Blocks Affected	Function	Remarks
0 - 127[2]	Job-oriented Pageable Key = User Allocated within partition	Subtask	Partition PFQE's	Problem Program[2]	
128	Job-oriented Pageable Key = User Freed at EOT Allocated from the high end of the partition	Subtask	Partition PFQE's	Partition Allocation[3]	Used by Supervisor and Scheduler.
229 - 232	NOT SUPPORTED				
233	Job-oriented Fixed Key = 0 Freed at EOT Task-related Allocated in the Fixed PQA that is within the partition	Job Step Task Subtask	FQE for Fixed PQA PPQE for Partition	Fixed PQA	
234	Job-oriented Fixed Key = 0 Freed at EOS Job-Step-related Allocated in the Fixed PQA that is within the partition	Job Step	FQE for Fixed PQA PPQE for Partition	Fixed PQA	GETMAIN limited to 2K maximum.

Subpool Number	Attributes	GQE Built[1]	Control Blocks Affected	Function	Remarks
235	Job-oriented Fixed Key = 0 Must be explicitly freed Explicitly assigned Allocated in the Fixed PQA that is within the partition	Job Step task Subtask	FQE for Fixed PQA PPQE for Partition	Fixed PQA	GETMAIN limited to 2K maximum FREEMAIN must be issued even if ABTERM occurs.
236, 237	Reserved for VS2 Scheduler				
238	Job-oriented Pageable Key = User Freed at EOT Allocated within partition	Subtask	Partition PFQE's	Partition Allocation[4]	Used by Supervisor and Scheduler, only.
239	NOT SUPPORTED				
240	Job-oriented Pageable Key = User Freed at EOS Job-Step-related Allocated from the high end of the partition	no	Partition PFQE's	Partition Allocation[3]	
241	Pageable Key = User Key = 0 System-oriented Must be explicitly freed Explicitly assigned	no	FQE for PSQA PCVT Boundary Box	Pageable SQA[5]	GETMAIN limited to 2K maximum. FREEMAIN must be issued even if ABTERM occurs.
242	Fixed Key = 0 V=R	no	FQE's for SQA MSS Boundary Box	V=R space	For use by Scheduler, only.

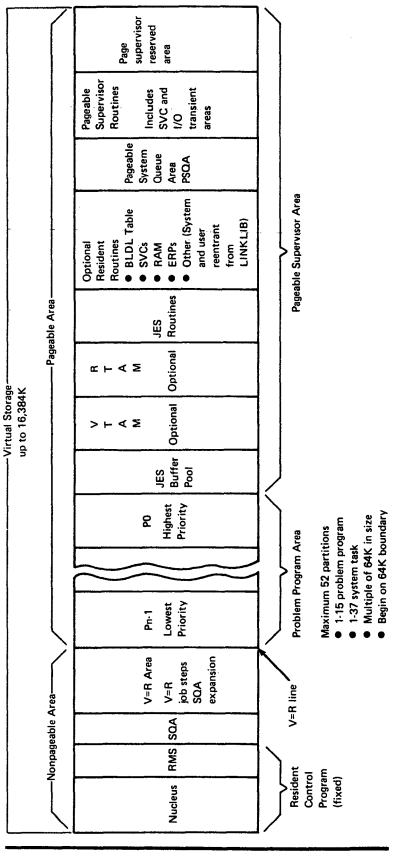
Subpool Number	Attributes	GQE Built[1]	Control Blocks Affected	Function	Remarks
243, 244	Reserved for VS2, Release 1				
245	Fixed Key = 0 System-oriented Must be explicitly freed V=R	no	FQE's for SQA Master Scheduler Resident Data Area Boundary Box	Fixed SQA[6]	GETMAIN limited to 2K maximum.
246	NOT SUPPORTED				
247 - 249	Reserved for VS2 and/or OS/MVT				
250	Job-oriented Pageable Key = User Allocated within partition	no	Partition PFQE's	Partition Allocation[3]	
251	Job-oriented Pageable Key = User Freed at EOS Job-Step-related Allocated from the low end of the partition	no	Partition PFQE'a	Partition Allocation[4]	For programs, only.
252	Job-oriented Pageable Key = 0 Freed at EOS Job-Step-related	Job Step task Subtask[7]	PFQE for Pageable PQA PPQE for Partition	Pageable PQA	
253	Job-oriented Pageable Freed at EOT Key = 0 Task-related Allocated in the pageable PQA that is in the partition	Job Step task Subtask	PFQE for Pageable PQA PPQE for Partition	Pageable PQA	Shared components must not use this subpool.

Subpool Number	Attributes	GQE Built[1]	Control Blocks Affected	Function	Remarks
254	Job-oriented Pageable Key = 0 Freed at EOS Job-Step-related Allocated in the pageable PQA that is in the partition	Job Step task Subtask[7]	PFQE for Pageable PQA	Pageable PQA	Shared components must not use this subpool.
255	Job-oriented Pageable Key = 0 Must be explicitly freed Explicitly assigned Allocated in the pageable PQA that is in the partition	no	PFQE for Pageable PQA PPQE for Partition	Pageable PQA	Shared components must not use this subpool. FREEMAIN must be issued even if ABTERM occurs.

NOTES:

1 GQE's are never built for a system task. A GQE is built for a subtask only if the GETMAIN was done by a key-0 program running under that subtask.
2 If the requester for subpools 0-127 is in the supervisor state, the request is from problem program low storage. Problem program requests are from high storage.
3 128, 240, and 250 from high end.
4 238 and 251 from low end.
5 System tasks use PSQA. Their BBX is invalid. All requests for subpools 0-128, 238, 241, 250, and 251-255 are mapped in 241 for system tasks.
6 System task requests for fixed PQA (233-235) are mapped to fixed SQA.
7 GQE queued on job step GQE queue.

VS1 Virtual Storage Organization



The information in this section is of interest mainly to the systems programmer who is modifying or debugging the VS1 Supervisor.

Source Publications

Additional information about the supervisor macro outlines and SVC is in *OS/VS1 Supervisor Services and Macro Instructions*, GC24-5103 and *OS/VS1 Planning and Use Guide*, GC24-5090.

Supervisor Macro Outlines

Supervisor Macro General Outline

Symbol	Macro Name	Parameters

ABEND	completion code,[DUMP][,STEP]
ATTACH	{EP=symbol / EPLOC=address of name / DE=address of list entry} [,DCB=dcb address] [,PARAM=(addresses)[,VL=1]][,ECB=ecb address] [,ETXR=exit routine address][,LPMOD=number] [,DPMOD=number] [,TASKLIB=joblib DCB addr] [,TQE={YES/NO}][,FPREGSA={YES/NO}]
ATTACH (list form)	{EP=symbol / EPLOC=address of name / DE=address of list entry} [,DCB=dcb address] [,ECB=ecb address] [,ETXR=exit routine address] [,LPMOD=number] [,DPMOD=number],SF=L [,TASKLIB=joblib DCB addr] [,TQE={YES/NO}][,FPREGSA={YES/NO}]
ATTACH (execute form)	{EP=symbol / EPLOC=address of name / DE=address of list entry} [,DCB=dcb address] [,PARAM=(addresses) [,VL=1]] [,ECB=ecb address] [,ETXR=exit routine address] [,LPMOD=number] [,DPMOD=number] [,TASKLIB=joblib DCB addr] [,TQE={YES/NO}][,FPREGSA={YES/NO}] {,MF=(E,{problem program list address (1)}) / ,SF=(E,{control program list address (15)}) / ,MF=(E,{address (1)}),SF=E,{address (15)})}

Supervisor Macro Outlines (cont'd)

CALL	{entry point name / (15)} [,(address parameters)[,VL]] [,ID=number]
CALL (list form)	,(address parameters) [,VL],MF=L
CALL (execute form)	{entry point name / (15)} [,(address parameters)] [,VL][,ID=number] ,MF=(E,{problem program list address / (1)})
CHAP	priority change value [,tcb location address / ,'S']
DELETE	{EP=symbol / EPLOC=address of name / DE=address of list entry}
DEQ	(qname address,rname address,[rname length] ,[STEP / SYSTEM],...)[,RET=HAVE]
DEQ (list form)	([qname address],[rname address],[rname length] ,[SYSTEM / STEP],...)[,RET=HAVE],MF=L
DEQ (execute form)	[([qname address],[rname address],[rname length] ,[SYSTEM / STEP],...)][,RET=HAVE / ,RET=NONE] ,MF=(E,{control program list address / (1)})
DETACH	tcb location address
DOM	{MSG=register / MSGLIST=address}
DXR	reg1,reg2
ENQ	(qname address,rname address,[E / S],[rname length] ,[SYSTEM / STEP],...)[,RET=TEST / ,RET=USE / ,RET=HAVE / ,RET=CHNG]

ENQ (list form)	([qname address],[rname address],[E/S] ,[rname length],[SYSTEM/STEP],...) [,RET=HAVE/,RET=TEST/,RET=USE/,RET=CHNG] ,MF=L
ENQ (execute form)	[([qname address],[rname address],[E/S] ,[rname length],[SYSTEM/STEP],...)][,RET=HAVE/,RET=TEST/,RET=USE/,RET=NONE/,RET=CHNG] ,MF=(E,{control program list address / (1)})
EXTRACT	answer area address [,tcb location address/,'S'] ,FIELDS=(codes)
EXTRACT (list form)	[answer area address][,tcb location address/,'S'] [,FIELDS=(codes)],MF=L
EXTRACT (execute form)	[answer area address][,tcb location address/,'S'] [,FIELDS=(codes)] ,MF=(E,{control program list address / (1)})
FREEMAIN	{E,LV=number,A=address[,SP=number] R,SP=(0) R,LV=(0),A=address R,LV=(0),A=(1) R,LV=number,A=address[,SP=number] R,LV=number,A=(1)[,SP=number] V,A=address[,SP=number]}
FREEMAIN (list form)	{[E][,LV=number][,A=address][,SP=number] [V][,A=address][,SP=number]} ,MF=L
FREEMAIN (execute form)	{[E][,LV=number][,A=address][,SP=number] [V][,A=address][,SP=number]} ,MF=(E,{control program list address / (1)})

Supervisor Macro Outlines (cont'd)

GETMAIN	EC, LV=number, A=address [, SP=number] [, BNDRY= {DBLWD / PAGE}] EU, LV=number, A=address [, SP=number] [, BNDRY= {DBLWD / PAGE}] R, LV=number[, SP=number] R, LV=(0) VC, LA=address, A=address[, SP=number] [, BNDRY= {DBLWD / PAGE}] VU, LA=address, A=address[, SP=number] [, BNDRY= {DBLWD / PAGE}]
GETMAIN (list form)	{[EC][, LV=number] / [EU][, LV=number] / [VC][, LA=address] / [VU][, LA=address]} [, A=address][, SP=number] [, BNDRY= {DBLWD / PAGE}], MF=L
GETMAIN (execute form)	{[EC][, LV=number] / [EU][, LV=number] / [VC][, LA=address] / [VU][, LA=address]} [, A=address][, SP=number] [, BNDRY= {DBLWD / PAGE}], MF=(E, {control program list address / (1)})
GTRACE	DATA=address, LNG=number, ID=number[, FID=number]
GTRACE (list form)	[DATA=address][, LNG=length][, FID=number], MF=L
GTRACE (execute form)	ID=value, MF=(E, {parameter list address / (1-12)}) [, DATA=address][, LNG=length][, FID=number]
IDENTIFY	{EP=symbol / EPLOC=address of name}, ENTRY=entry point address
LINK	{EP=symbol / EPLOC=address of name / DE=address of list entry} [, DCB=dcb address] [, PARAM=(addresses)][, VL=1][, ID=number]
LINK (list form)	{EP=symbol / EPLOC=address of name / DE=address of list entry} [, DCB=dcb address], SF=L

Supervisor Macro Outlines (cont'd)

LINK (execute form)	[EP=symbol / EPLOC=address of name / DE=address of list entry] [,DCB=dcb address] [,PARAM=(addresses)][,VL=1][,ID=number] { ,MF=(E, {problem program list address / (1)}) / ,SF=(E, {control program list address / (15)}) / ,MF=(E, {address / (1)}),SF=(E, {address / (15)}) }
LOAD	{ EP=symbol / EPLOC=address of name / DE=address of list entry } [,DCB=dcb address]
PGOUT	R,A={addr1 / (1-14)} [,EA={addr2 / 2-15}]
PGOUT (list form)	L,LA={addr / (1)}
PGRLSE	LA={addr1 / (reg1)} ,HA={addr2 / (reg2)}
PGRLSE (list form)	MF=L[,LA=addr1][,HA=addr2]
PGRLSE (execute form)	MF=(E, {listaddr / (reg3)}) [,LA={addr1 / (reg1)}] [,HA={addr2 / (reg2)}]
POST	ecb address[,completion code]
RETURN	[(reg1[,reg2])][,T] [,RC={number / (15)}]
SAVE	(reg1[,reg2])[,T][,identifier name]
SEGWT	external segment name
SNAP	DCB=dcb address[TCB=address][,ID=number] [,SDATA=(code for control program blocks)] [,PDATA=(code for problem program areas)] [,STORAGE=(starting address,ending address,...) / ,LIST=address of list]
SNAP* (list form)	[DCB=address] [,ID=number] [,SDATA=(code)] [,PDATA=(code)] [,STORAGE=(address,address,...) / ,LIST=address] ,MF=L
SNAP (execute form)	[DCB=address] [,TCB={address / 'S'}] [,ID=number] [,PDATA=code] [,SDATA=code] [,STORAGE=(address,address,...) / ,LIST=address] ,MF=(E, {control program list address / (1)})
SPIE	[interruption exit address, (interruptions)]

SPIE (list form)	[interruption exit address] [,(interruptions)],MF=L
SPIE (execute form)	[interruption exit address] [,(interruptions)] ,MF=(E, { control program list address (1) })
STAE	{ 0 / exit address } { ,OV / ,CT } [,PARAM=list address] [,XCTL= { YES / NO }] [,PURGE= { QUIESCE / HALT / NONE }] [,ASYNCH= { YES / NO }]
STAE (list form)	[exit address][,PARAM=list address] [,PURGE= { QUIESCE / HALT / NONE }] [,ASYNCH= { YES / NO }] ,MF=L
STAE (execute form)	{ 0 / exit address } { ,OV / ,CT } [,PARAM=list address] [,XCTL= { YES / NO }] [,PURGE= { QUIESCE / HALT / NONE }] [,ASYNCH= { YES / NO }] ,MF=(E, { remote list address (1) })
STIMER	{ REAL[,timer completion exit address] / TASK[,timer completion exit address] / WAIT } [,DINTVL=address / ,BINTVL=address / ,TUINTVL=address / ,TOD=address]
STIMERE	ID= { value / addr / ALL[1] } { { ,BINTVL= { value / addr } / ,DINTVL= { value / addr } / ,MICVL= { value / addr } } [,REPLACE=YES] } { { ,TEST= { (BIN,addr) / (DEC,addr) / (MIC,addr) } / ,CANCEL=YES / ,CANCEL=YES, TEST= { (BIN,addr) / (DEC,addr) / (MIC,addr) } } } [,ECB=addr / ,EXIT=addr[,SVAREA= { YES / NO }]] [,ERRET=addr]

Note[1]: ID=ALL can be used with CANCEL if TEST is not specified; ID=ALL is defaulted if CANCEL appears alone. ALL is invalid for all other uses of ID.

Supervisor Macro Outlines (cont'd)

STIMERE (list form)	[ID=value] [,BINTVL=value ,DINTVL=value ,MICVL=value] [,ECB=addr ,EXIT=addr[,SVAREA={YES/NO}] ,MF=L]
STIMERE (execute form)	[ID={value/addr}] [,BINTVL={value/addr} ,DINTVL={value/addr} ,MICVL={value/addr}] [,REPLACE=YES] [,ECB=addr ,EXIT=addr[,SVAREA={YES/NO}]] [,ERRET=addr] ,MF=(E,{parameter list address/(1)})
TIME	[DEC/BIN TU MIC,address]
TTIMER	[CANCEL]
WAIT	[number of events,]{ECB=address/ECBLIST=address}
WAITR	[number of events,]{ECB=address/ECBLIST=address}
WTL	'message'
WTL (list form)	'message',MF=L

Supervisor Macro Outlines (cont'd)

WTL (execute form)	MF=(E, {control program list address / (1)})
WTO	{'message' / ('text'[,line type]),...} [,ROUTECDE=(number[,number],...)] [,DESC=number]
WTO (list form)	{('text'[,line type]),... / 'message'} [,ROUTCDE=(number[,number...])] [,DESC=number] ,MF=L
WTO (execute form)	MF=(E, {control program list address / (1)})
WTOR	'message',reply address,length of reply 'ecb address[,ROUTCDE=(number[,number...])] [,DESC=number]
WTOR (list form)	'message',[reply address],[length of reply] ,[ecb address][,ROUTCDE=(number[,number...])] [,DESC=number],MF=L
WTOR (execute form)	,[reply address],[length of reply],[ecb address] ,MF=(E, {control program list address / (1)})
XCTL	[(reg1[,reg2])], {EP=symbol / EPLOC=address of name / DE=address of list entry} [,DCB=dcb address]
XCTL (list form)	{EP=symbol / EPLOC=address of name / DE=address of list entry} [,DCB=dcb address],SF=L
XCTL (execute form)	[(reg1[,reg2])] {,EP=symbol / ,EPLOC=address of name / ,DE=address of list entry} [,DCB=dcb address] {,MF=(E, {problem program list address / (1)}) / ,SF=(E, {control program list address / (15)}) / ,MF=(E {address / (1)}) SF=(E, {address / (15)})}

Note: Shaded area for Multiple Console Support.

Supervisor Macro Parameter Notation

Abbreviation	Meaning
Sym	Any symbol valid in the assembler language.
Deg Dig	Any decimal digits, up to the value indicated in the associated macro instruction description. If both Sym and Dec Dig are checked, an absolute expression is also allowed.
Register	A general register, always coded within parentheses, as follows:
(2-12) -	one of the general registers 2 through 12, previously loaded with the right-adjusted value or address indicated in the macro-instruction description. The unused high-order bits must be set to zero. The register may be designated symbolically or with an absolute expression.
(1) -	general register 1, previously loaded as indicated above. Designate the register as (1) only.
(0) -	general register 0, previously loaded as indicated above. Designate the register as (0) only.
RX type	Any address that is valid in an RX-type instruction (e.g., LA) may be designated.
A-Type Adcon Type	Any address that may be written in an A-type address constant may be designated.

Summary of Supervisor Macro Operands

MACRO INSTRUCTION	OPERANDS	WRITTEN AS						
		Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	A-type Adcon type
ABEND	completion code	S	S	S	S			
	DUMP	written as shown						
	STEP	written as shown						
ATTACH	DCB			SE			E	SL
	DE=			SE			E	SL
	DPMOD=	SLE	SLE	SE				
	ECB=			SE			E	SL
	EP=	SLE						
	EPLOC=			SE			E	SL
	ETXR=			SE			E	SL
	FPREGSA=	YES or NO						
	LPMOD=	SLE	SLE	SE				
	PARAM=			SE			E	S
	TQE=	YES or NO						
	VL=1	written as shown						
CALL	entry point name	SE						
	address parameters			SE			E	SL
	VL	written as shown						
	ID=	SE	SE					
CHAP	priority change value	S	S	S		S		
	tcb location address			S	S		S	
DELETE	DE=			S		S	S	
	EP=	S						
	EPLOC=			S		S	S	

S=standard; L=list; E=execute

Summary of Supervisor Macro Operands (cont'd)

MACRO INSTRUCTION	OPERANDS	WRITTEN AS						
		Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	A-type Adcon type
DEQ	qname address			SE			E	SL
	rname address			SE			E	SL
	rname length	SLE	SLE	SE				
	STEP or SYSTEM	written as shown						
	RET=HAVE	written as shown						
	RET=NONE	written as shown (E form only)						
DETACH	tcb location address	S		S	S		S	
DOM	MSG=			S	S			
	MSGLIST=	S		S	S		S	
DXR	reg1	S	S					
	reg2	S	S					
ENQ	qname address			SE			E	SL
	rname address			SE			E	SL
	E or S	written as shown						
	rname length	SLE	SLE	SE				
	STEP or SYSTEM	written as shown						
	RET=	TEST, USE, CHNG, or HAVE						
	RET=NONE	written as shown (E form only)						
FREEMAIN	E, R or V	written as shown						
	A=(with E, L, or V)			SE			E	SL
	A=(with R)			S	S		S	
	LV=(with E)	SLE	SLE	SE				
	LV=(with R)	S	S	S		S		
	SP=(with E or V)	SLE	SLE	SE				
	SP=(with R)	S	S	S		S		
GETMAIN	EC, EU, VC, or VU	refer to macro description						
	A=			SE			E	SL
	BNDRY=	DBLWD or PAGE						
	LA=			SE			E	SL

Summary of Supervisor Macro Operands (cont'd)

MACRO INSTRUCTION	OPERANDS	WRITTEN AS							
		Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	Register (15)	RX-type	A-type Adcon type
GETMAIN (cont'd)	LV=(with E)	SLE	SLE	SE					
	LV(with R)	S	S	S		S			
	SP=(with E or V)	SLE	SLE	SE					
	SP=(with R)	S	S	S		S			
GTRACE	DATA=			S				S	SLE
	LNG=	SLE	SLE	SLE					
	FID=	SLE	SLE	SLE					
	ID=	SE	SE						
IDENTIFY	ENTRY=			S	S			S	
	EP=	S							
	EPLOC=			S		S		S	
LINK	DCB=			SE				E	SL
	DE=			SE				E	SL
	EP=	SLE							
	EPLOC=			SE				E	SL
	ID=	SE	SE						
	PARAM=			SE				E	S
	VL=1	written as shown							
LOAD	DCB=			S	S			S	
	DE=			S		S		S	
	EP=	S							
	EPLOC=			S		S		S	
PGOUT	A=	S	S	S	S				S
	EA=	S	S	S			S		S
	LA=	L	L	L	L				L
PGRLSE	LA=			SE		SE			SLE
	HA=			SE	SE				SLE
	list addr=							E	
	reg 3=			E					
POST	ecb address			S	S			S	
	completion code	S	S	S		S			

Summary of Supervisor Macro Operands (cont'd)

MACRO INSTRUCTION	OPERANDS	WRITTEN AS						
		Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	A-type Adcon type
RETURN	(reg1,reg2)		S					
	T	written as shown						
	RC=	S	S	or (15)				
SAVE	(reg1,reg2)		S					
	T	written as shown						
	identifier name	character string or *						
SEGWT	external segment name	S						
SNAP	DCB=			SE			E	SL
	ID=	SLE	SLE	SE				
	LIST=			SE			E	SL
	PDATA	refer to macro description						
	SDATA	refer to macro description						
	STORAGE			SE			E	SL
	TCB=			SE			E	S
SPIE	interruption exit address			SE			E	SL
	interruptions		SLE					
STIMER	REAL, TASK or WAIT	written as shown						
	timer completion exit addr			S		S	S	
	BINTVL=			S	S		S	
	DINTVL=			S	S		S	
	TOD=			S	S		S	
	TUINTVL=			S	S		S	

Summary of Supervisor Macro Operands (cont'd)

MACRO INSTRUCTION	OPERANDS	WRITTEN AS						
		Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	A-type Adcon type
STIMERE	ID=addr	SE		SE			SE	
	BINTVL=addr	SE		SE			SE	
	DINTVL=addr	SE		SE			SE	
	MICVL=addr	SE		SE			SE	
	ID=value		SLE					
	BINTVL=value		SLE					
	DINTVL=value		SLE					
	MICVL=value		SLE					
	ID=ALL	written as shown (S form only)						
	REPLACE=YES	written as shown (SE forms only)						
	TEST=(BIN DEC MIC ,addr)	S		S			S	
	CANCEL=YES	written as shown (S form only)						
	ECB=addr	SLE		SLE			SLE	
	EXIT=addr	SLE		SLE			SLE	
	SVAREA=	YES or NO						
	ERRET=addr	SE		SE			SE	
TIME	DEC or BIN or TU	written as shown						
	MIC	written as shown						
	address			S		S	S	
TTIMER	CANCEL	written as shown						
	TU	written as shown						

Summary of Supervisor Macro Operands (cont'd)

MACRO INSTRUCTION	OPERANDS	WRITTEN AS						
		Sym	Deg Dig	Register (2-12)	Register (1)	Register (0)	RX-type	A-type Adcon type
WAIT WAITR	number of events	S	S	S		S		
	ECB=			S	S		S	
	ECBLIST=			S	S		S	
WTL	message	any message within apostrophes						
WTO	message	any message within apostrophes						
	text	any text within apostrophes						
	line type	C,L,D,DE, or E						
	ROUTCDE=		SL					
	DESC=		SL					
WTOR	message	any message within apostrophes						
	reply address			SE			E	SL
	length of reply	SLE	SLE	SE				
	ecb address			SE			E	SL
	ROUTCDE=		SL					
	DESC=		SL					
XCTL	(reg1,reg2)		SE				E	S
	DCB=			SE			E	SL
	DE=			SE			E	SL
	EP=	SLE						
	EPLOC=			SE			E	SL

Programming Conventions for SVC Routines

Conventions	Type 1	Type 2	Type 3	Type 4
Part of resident control program	Yes	Yes	No	No
Size of routine	Any	Any	≤2048 bytes	Each load module ≤2048 bytes
Reenterable routine	Optional, but must be serially reusable	Yes	Yes	Yes
May allow I/O and external interruptions	No	Yes	Yes	Yes
Number of routine	Numbers assigned to your SVC routines should be in descending order from 255 through 200			
Name of routine	IGCnnn	IGCnnn	IGC00nnn	IGCssnnn
Register contents at entry time	Registers 3, 4, 5, and 14 contain communication pointers; registers 0, 1, and 15 are parameter registers			
May contain relocatable data	Yes	Yes	No	No
Can supervisor request block (SVRB) be extended	Not applicable	Yes	Yes	Yes
May issue WAIT macro instruction	No	Yes	Yes	Yes
May issue XCTL macro instruction	No	No	No	Yes
May issue SVCs	No	Yes	Yes	Yes
Exit from SVC Routine	Branch using return register 14			
Method of abnormal termination	Use resident abnormal termination routine	Use ABEND macro instruction or resident abnormal termination routine		

SVC Register Contents

DEC (hex) No.	Type	Macro	Register 0	Register 1
0(0)	I	EXCP		IOB address
0(0)	I	XDAP		
1(1)	I	WAIT	Event count	ECB address or 2's complement of ECB list address
1(1)	I	WAITR	2's complement of event count	ECB address or 2's complement of ECB list address
1(1)	I	PRTOV		
2(2)	I	POST	Completion code	ECB address or parm list address with high-order bit on
3(3)	I	EXIT		
4(4)	I	GETMAIN		Parameter list address
5(5)	I	FREEMAIN		Parameter list address
6(6)	II	LINK		
7(7)	II	XCTL		
8(8)	II	LOAD	Address of entry point address	DCB address
9(9)	II	DELETE	Address of program name	
10(A)	I	GETMAIN or FREEMAIN	Subpool number (byte 0), length (bytes 1-3)	If negative, indicates GETMAIN. If positive, contains address of area to be freed
11(B)	I	TIME	Pointer to a doubleword to store TOD if MIC specified	Time units code
12(C)	II	SYNCH		
13(D)	IV	ABEND		Completion code
14(E)	II	SPIE		PICA address
15(F)	I	ERREXCP		Address of request queue element
16(10)	III	PURGE		
17(11)	III	RESTORE		IOB chain address
18(12)	II	BLDL	Address of build list	DCB address
18(12)	II	FIND		
19(13)	IV	OPEN		Address of parameter list of DCB addresses
20(14)	IV	CLOSE		Address of parameter list of DCB addresses
21(15)	III	STOW	Parameter list address	DCB address

SVC Register Contents (cont'd)

Dec (hex) No.	Type	Macro	Register 0	Register 1
22(16)	IV	OPEN TYPE=J		Address of parameter list of DCB addresses
23(17)	IV	CLOSE TYPE=T		Address of parameter list of DCB addresses
24(18)	III	DEVTYPE		ddname address
25(19)	III	TRKBAL		DCB address
26(1A)	IV	CATALOG		Parameter list address
26(1A)	IV	INDEX		Parameter list address
26(1A)	IV	LOCATE		Parameter list address
27(1B)	IV	OBTAIN		Parameter list address
28(1C)	IV	CVOL		
29(1D)	IV	SCRATCH	UCB address	Parameter list address
30(1E)	IV	RENAME	UCB address	Parameter list address
31(1F)	IV	FEOV		DCB address
32(20)	IV	ALLOC		Address of UCB list
33(21)	III	IOHALT		UCB address
34(22)	IV	MGCR (MAST CMD EXCP)		
34(22)	IV	QEDIT		
35(23)	IV	WTO		Message address
35(23)	IV	WTOR		Message address
36(24)	IV	WTL		Message address
37(25)	II	SEGLD		Segment name addr
37(25)	II	SEGWT		Segment name addr
38(26)		Reserved		
39(27)	III	LABEL		Parameter list address
40(28)	II	EXTRACT		Parameter list address
41(29)	II	IDENTIFY	Entry point name address	Size of work area in doublewords
42(2A)	II	ATTACH		May contain user parm list address
43(2B)	III	CIRB	Entry point address	Size of work area in doublewords

SVC Register Contents (cont'd)

Dec (hex) No.	Type	Macro	Register 0	Register 1
44(2C)	III	CHAP	+Increase priority -Decrease priority	TCB address
45(2D)	II	OVLYBRCH		
46(2E)	I	TTIMER		1: Cancel
47(2F)	II	STIMER	Exit address (Option flags in high order byte)	Timer interval address
48(30)	II	DEQ		DEQ parameter list address
49(31)		Reserved		
50(32)		Reserved		
51(33)	IV	SNAP		Parameter list address
52(34)	IV	RESTART		DCB address
53(35)	III	RELEX	Key address	DCB address
54(36)	II	DISABLE		
55(37)	IV	EOV	IOB address	DCB address
56(38)	II	ENQ		ENQ parameter list address
57(39)	III	FREEDBUF	DECB address	DCB address
58(3A)	II	RELBUF		DCB address
58(3A)	II	REQBUF		DCB address
59(3B)	IV	OLTEP		
60(3C)	III	STAE	0 Create SCB 4 Cancel SCB 8 Overlay SCB	Parameter list address
61(3D)		Reserved		
62(3E)	III	DETACH		TCB address location
63(3F)	IV	CHKPT		DCB address
64(40)	III	RDJFCB		Address of parameter list of DCB addresses
65(41)		Reserved		
66(42)	IV	BTAMTEST		
67(43)	II	ENDREADY		
68(44)	IV	SYNADAF	Same as reg 0 on entry to SYNAD	Same as reg 1 on entry to SYNAD
68(44)	IV	SYNADRLS		
69(45)	III	BSP		DCB address
70(46)	II	GSERV		Parameter list address

SVC Register Contents (cont'd)

Dec (hex) No.	Type	Macro	Register 0	Register 1
71(47)	IV	RLSEBFR		Parameter list address
71(47)	IV	ASGNBRF		Parameter list address
71(47)	IV	BUFINQ		Parameter list address
72(48)	IV	CHATR		Parameter list address
73(49)	IV	SPAR		Parameter list address
74(4A)	IV	DAR		Parameter list address
75(4B)	III	DQUEUE		Parameter list address
76(4C)	IV	IFBSTAT		
77(4D)	IV	QTAMTEST		
78(4E)	IV	LSPACE		
79(4F)	I	STATUS		
80(50)		Reserved		
81(51)	IV	SETDEV		Parameter list address
81(51)	IV	SETPRT		
82(52)	IV	DASDR		
83(53)	III	SMFWTM		Message address
84(54)	I	GRAPHICS		
85(55)	IV	DDRSWAP		
86(56)	IV	ATLAS		Parameter list address
87(57)	III	DOM		DOM message Id if reg 0=0 A pointer to a list of DOM message Ids if reg 0 negative.
88(58)	III	MOD88	Routine Code	DCB address
89(59)	III	EMSRV		Parameter list address
90(5A)	IV	XQMNGR	Address of list of ECB/IOB pointers (optional)	QMPA address
91(5B)	III	VOLSTAT	DCB address	Zero: Issued by CLOSE Non-zero: issued by EOV
92(5C) - 97(61)		Reserved		
98(62)	IV	PROTECT		
102(66)	I	AQCTL		Parameter list address
103(67)	III	XLATE		
104(68)	IV	TOPCTL		
105(69)	III	IMGLIB	0=OPEN SYS1.IMAGELIB DCB addr=CLOSE SYS1.IMAGELIB	
106(6A)		Reserved		
107(6B)	I	MODESET		Parameter list address
108(6C)	IV	ISOLATE		

SVC Register Contents (cont'd)

DEC (hex) No.	Type	Macro	Register 0	Register 1
109(6D)	II	Extended SVC Router (ESR) macros	Parameters to ESR	Parameters to ESR
110(6E)		Reserved		
111(6F)	II	JECS macros		Parameter List address
112(70)	I	PGRLSE	Low address	High address
113(71) (Note)	I	SIR macros	ECB address or TCB address	Parameter list address or begin address for request
114(72)	I	EXCPVR	If 0, EXCP for paging supervisor; If 4, EXCP for subsystem	IOB address
115(73)	I	BLKPAGE		
116(74)	I	Extended SVC Router (ESR) macros	Parameters to ESR	Parameters to ESR
117(75)	II	DEBCHK	Function	†DCB
118(76)	I	DSS	DSSAT	
119(77)	I	TESTAUTH		
122(7A)	II	Extended SVC Router (ESR)	Parameters to ESR	Parameters to ESR
124(7C)	III	TPIO	Function code + ACB address	Address of Work Element
125(7D)	I	EVENTS	Flags + entry addr in EVENTS table, or ECB addr, or 0	EVENTS table addr
126 (7E)	II			Address of request block
127(7F)	II	DIS macros		Parameter List addr
129(81)	II	DIS macros		Parameter List addr

Note: Register 15 may contain zero or an address. See PGOUT macro.

SVC Directory

Dec. (hex) No.	Type	Macro	Module Name
0 (0)	1	EXCP	IEAIOS00
1 (1)	1	WAIT	IEAAWT
2 (2)	1	POST	IEAAPT
3 (3)	1	EXIT	IEAATA00
4 (4)	1	GETMAIN	IEAAMS00
5 (5)	1	FREEMAIN	IEAAMS00
6 (6)	2	LINK	IEAATC00
7 (7)	2	XCTL	IEAATC00
8 (8)	2	LOAD	IEAATC00
9 (9)	2	DELETE	IEAJDL00
10(A)	1	REGMAIN	IEAAMS00
11(B)	1	TIME	IEA0RT01
12(C)	2	SYNCH	IEAASY00
13(D)	4	ABEND	IEANTM00- IEANTM0M
14(E)	2	SPIE	IEAAPX00
15(F)	1	ERREXCP	IEAIOS00
16(10)	3	PURGE	IECIPRL2
17(11)	3	RESTORE	IGC0001G
18(12)	2	BLDL/FIND	IGC018
19(13)	4	OPEN	IGC00011
20(14)	4	CLOSE	IGC00020
21(15)	3	STOW	IGC0002A
22(16)	4	OPEN TYPE=J	IGC0002B
23(17)	4	CLOSE TYPE=T	IGC0002C
24(18)	3	DEVTYPE	IGC0002D
25(19)	3	TRKBAL	IGC0002E
26(1A)	4	CATALOG	IGC0002F
26(1A)	4	INDEX	IGC0002F

SVC Directory (cont'd)

Dec. (hex) No.	Type	Macro	Module Name
26(1A)	4	LOCATE	IGC0002F
27(1B)	4	OBTAIN	IGC0002G
28(1C)	4	CVOL	IGC0002H
29(1D)	4	SCRATCH	IGC0002I
30(1E)	4	RENAME	IGC00030
31(1F)	4	FEOV	IGC0003A
32(20)	4	ALLOC	IGC0003B
33(21)	3	IOHALT	IGC0003C
34(22)	4	MGCR	IEE0303D
34(22)	4	QEDIT	IEE0303D
35(23)	4	WTO	IEEMFWTO
35(23)	4	WTOR	IEEMFWTO
36(24)	4	WTL	IEE0303F
37(25)	2	SEGLD	IEWSUOVR
37(25)	2	SEGWT	IEWSUOVR
39(27)	3	LABEL	IGC00031
40(28)	2	EXTRACT	IEABXR00
41(29)	2	IDENTIFY	IEAAID00
42(2A)	2	ATTACH	IEAQAT
43(2B)	3	CIRB	IEAAEF00
44(2C)	3	CHAP	IEAQTB00
45(2D)	2	OVLYBRCH	IEWSVOVR
46(2E)	1	TTIMER	IEA0ST01
47(2F)	2	STIMER	IEA0ST01
48(30)	2	DEQ	IEAGENQ1
51(33)	4	SNAP	IEAAAD00- IEAAAD05 and IEAAAD0A- IEAAAD0L
52(34)	4	RESTART	IEFVSMBR
53(35)	3	RELEX	IGC0005C

SVC Directory (cont'd)

Dec. (hex) No.	Type	Macro	Module Name
54(36)	2	DISABLE	IGC054
55(37)	4	EOV	IGC0005E
56(38)	2	ENQ	IEAGENQ1
57(39)	3	FREEDBUF	IGC0005G
58(3A)	2	RELBUF	IGC058
58(3A)	2	REQBUF	IGC058
59(3B)	4	OLTEP	IGC0005I
60(3C)	3	STAE	IEAAST00
62(3E)	3	DETACH	IEAGED02
63(3F)	4	CHKPT	IHJACP00
64(40)	3	RDJFCB	IGC0006D
66(42)	4	BTAMTEST	IGC0006E
67(43)	2	ENDREADY	IGC0010C
68(44)	4	SYNADAF	IGC0006H
68(44)	4	SYNADRLS	IGC0006H
69(45)	3	BSP	IGC00061
70(46)	2	GSERV	IGC070
71(47)	4	RLSEBFR	IGC0007A
71(47)	4	ASGNBRF	IGC0007A
71(47)	4	BUFINQ	IGC0007A
72(48)	4	CHATR	IEECMCTR
73(49)	4	SPAR	IGC0007C
74(4A)	4	DAR	IGC0007D
75(4B)	3	DQUEUE	IGC0007E
76(4C)	3	IFBSTAT	IFBSTAT
77(4D)	4	QTAMTEST	
78(4E)	4	DSCAN	IGC0007H
79(4F)	1	STATUS	IEAQSETS
81(51)	4	SETPRT	IGC0008A
81(51)	4	SETDEV	IGC0008A
82(52)	4	DASDR	IGC0008B
83(53)	3	SMFWTM	IEESMF8C
84(54)	1	GRAPHICS	IGC084
85(55)	4	DDRSWAP	IGC0008E

SVC Directory (cont'd)

Dec. (hex) No.	Type	Macro	Module Name
86(56)	4	ATLAS	IGC0008F
87(57)	3	DOM	IEECXDOM
88(58)	3	MOD88	IGC0008H
89(59)	3	EMSRV	IGC00081
90(5A)	4	XQMNGR	IEFXQM00
91(5B)	3	VOLSTAT	IGC0009A
98(62)	4	PROTECT	
102(66)	1	AQCTL	IEDQEB
103(67)	3	XLATE	IGG0010C
104(68)	4	TOPCTL	IEDQEB
105(69)	3	IMGLIB	IGC0010E
107(6B)	1	MODESET	IEAVMODE
108(6C)	4	ISOLATE	
109(6D)	2	ESR (extended SVC router) macros	IGC116
111(6F)	2	JECS macros	IFGAZ016
112(70)	1	PGRLSE	IEAAIH00
113(71)	1	SIR macros	IEAAIH00
114(72)	1	EXCPVR	IEAIOS00
115(73)	1	BLKPAGE	
116(74)	1	ESR macros	IGC116
117(75)	2	DEBCHK	IFGDEBCHK
118(76)	1	AT	IEAAIH00
119(77)	1	TESTAUTH	IEAVTEST
122(7A)	2	ESR macros	IGC116
124(7C)	3	TPIO	ISTAP21
125(7D)	1	EVENTS	IEAAWT
126(7E)	2		ICBSVC IGC126
127(7F)	2	DIS macros	
129(81)	2	DIS macros	

Load Module Control

Explanation of Style	Footnotes:
Words in all capitals are coded as shown; appropriate values are to be substituted for words in lower case letters. Brackets, [ ], enclose operands that may be used or omitted as required; stacking within braces, { }, is used to indicate a choice of operands or values. Underlining, ___, indicates a default value.	* In full-word on full-word boundary ** In double-word on double-word boundary + Left justified in double-word on byte boundary o Multiple of eight; value given in bytes

Pass control and initiate execution	CALL	entry point name [,(address parameter [,address parameter]...)[,VL]] [,ID=0 to 65535]
Dynamically load and initiate execution	LINK	{ EP=entry point name EPLOC=address of entry point name+ DE=address of list entry } [,DCB=dcb address] [,PARAM=(address parameter [,address parameter]...) [,VL=1]] [,ID=0 to 65535]
Transfer control	XCTL	[range of registers to be restored], { EP=entry point name EPLOC=address of entry point name+ DE=address of list entry } [,DCB=dcb address]
Dynamically load	LOAD	{ EP=entry point name EPLOC=address of entry point name+ DE=address of list entry } [,DCB=dcb address]
Delete	DELETE	{ EP=entry point name EPLOC=address of entry point name+ DE=address of list entry }
Identify embedded entry point	IDENTIFY	{ EP=entry point name EPLOC=address of entry point name+ }, ENTRY=entry point address
Load overlay segment	SEGWT	external segment name

Synchronization

Wait for event	WAIT	[number of events,] { ECB=ecb address / ECBLIST=address of list of ecb addresses* }
Wait for event while lower priority task is executed	WAITR	[number of events,] { ECB=ecb address / ECBLIST=address of list of ecb addresses* }
Post event completion	POST	ecb address [,completion code]
Request control of serially reusable resource	ENQ	(qname address, rname address, [E/S], [rname length], [SYSTEM/STEP] ,...) [,RET=TEST / ,RET=USE / ,RET=HAVE / ,RET=CHNG]
Release serially reusable resource	DEQ	(qname address, rname address, [rname length], [STEP/SYSTEM] ,...) [,RET=HAVE] E means exclusive control S means shared control } default is E SYSTEM means resource used by more than one job STEP means resource used by issuing job
Set interval timer	STIMER	{ REAL, [address of interval end routine] / TASK, [address of interval end routine] / WAIT } { ,DINTVL=address of decimal interval** / ,BINTVL=address of binary interval in seconds* / ,TUINTVL=address of binary interval in timer units* / ,TOD=address of time-of-day of interval end** }
Test interval timer	TTIMER	[CANCEL] [,TU]

TIME AND TIME INTERVALS FOR TTIMER AND STIMER

Decimal (DINTVL operands):
Eight unpacked decimal digits in format HHMMSSth
HH = hours in 24-hour clock
MM = minutes
SS = seconds
t = tenths of seconds
h = hundredths of seconds

Binary in seconds (BINTVL operands):
Unsigned 32-bit binary number in a full-word on full-word boundary; least significant bit has a value of 0.01 second

Binary in timer units (TU or TUINTVL operands):
Unsigned 32-bit binary number in a full-word on full-word boundary; least significant bit has a value of 1 timer unit (1timer unit = 26 micro-seconds)

Synchronization (cont'd)

Set interval timer extended	STIMERE	ID= {value / addr / ALL[1]} { {,BINTVL= {value / addr} ,DINTVL= {value / addr} [,REPLACE=YES] ,MICVL= {value / addr} } { ,TEST = {(BIN,addr) / (DEC,addr) / (MIC,addr)} ,CANCEL=YES ,CANCEL=YES,TEST= {(BIN,addr) / (DEC,addr) / (MIC,addr)} } [,ECB=addr ,EXIT=addr[,SVAREA= {YES / NO}]] [,ERRET=addr]

Note[1]: ID=ALL can be used with CANCEL if TEST is not specified;
ID=ALL is defaulted if CANCEL appears alone. ALL is invalid for all other uses of ID.

TIME INTERVALS FOR STIMERE

Decimal (DINTVL operand):
Eight unpacked decimal digits in format HHMMSSth
HH = hours in 24-hour clock
MM = minutes
SS = seconds
t = tenths of seconds
h = hundredths of seconds

Binary in seconds (BINTVL operand):
31-bit positive binary number in a full-word on full-word boundary; least significant bit has a value of 0.01 second

Binary in microseconds (MICVL operand):
Unsigned 64-bit binary number in a double-word on a double-word boundary. Bit 51 is the low order digit of the interval value.

Program Interrupt Control

(see explanation of style - page 3-27)

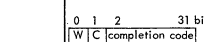
Enable and disable program interruptions and transfer control to interruption exit routine	SPIE	[interruption exit routine address] [,(interruption type[,interruption type],...)]

INTERRUPTION TYPES FOR SPIE

Type	Meaning	Maskable	Type	Meaning	Maskable
1	Operation	No	9	Fixed-point divide	No
2	Privileged operation	No	10	Decimal overflow	Yes
3	Execute	No	11	Decimal divide	No
4	Protection	No	12	Exponent overflow	No
5	Addressing	No	13	Exponent underflow	Yes
6	Specification	No	14	Significance	Yes
7	Data	No	15	Floating-point divide	No
8	Fixed-point overflow	Yes			

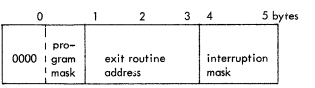
CONTROL BLOCKS

Event control block (ECB):

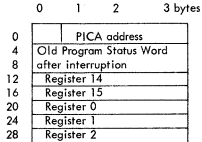


W = wait flag
C = completion flag

Program interruption control area (PICA):



Program interruption element (PIE):



General Services

(see explanation of style - page 3-27)

Delete message(s) from display	DOM	{MSG=register containing 24-bit, right-justified message number MSGLIST=address of list of fullwords, each a 24-bit, right-justified identification number of message to be deleted}
Write to operator	WTO	{'message' ('text'[,line type]),...} [,ROUTCDE=(number [,number] ,...)] [,DESC=number]
Write to operator and wait for reply	WTOR	'message',address of reply area,length of reply,ecb address [,ROUTCDE=(number [,number] ,...)] [,DESC=message descriptor code(s
Write to log	WTL	'message'
Divide extended precision floating point number	DXR	register containing dividend,register containing divisor Only registers 0 and 4 can be used; they may be specified in either order.
Get time and date	TIME	[DEC BIN TU MIC,address]

TIME FOR TIME

Decimal (DEC operand):
Eight packed decimal digits in format HHMMSSth

HH = hours in 24-hour clock
MM = minutes
SS = seconds
t = tenths of seconds
h = hundredths of seconds

Binary in seconds (BIN operand):
Unsigned 32-bit binary number in a full-word on full-word boundary; least significant bit has a value of 0.01 second

Binary in timer units (TU operand):
Unsigned 32-bit binary number in a full-word on full-word bondary; least significant bit has a value of 1 timer unit (1 timer unit = 26 micro-seconds)

Binary in microseconds (MIC operand):
Unsigned 64-bit binary number in a double-word on a double-word boundary. Bit 51 is the low order digit of the interval value.

General Services (cont'd) - Termination

(see explanation of style - page 3-27)

Save register contents	SAVE	(range of registers to be stored) [,T] [,identifier] In SAVE, T means: save registers 14 and 15.
Dump storage and continue	SNAP	DCB=address of data control block [,TCB=address of TCB address*] [,ID=1 to 127] [,SDATA=(ALL / NUC / TRT / CB / Q) [,ALL / ,NUC / ,TRT / ,CB / ,Q] ...)] [,PDATA=(ALL / PSW / REGS / SA or SAH / JPA or LPA or ALLPA / SPLS) [,ALL / ,PSW / ,REGS / ,SA or ,SAH / ,JPA or ,LPA or ,ALLPA / ,SPLS] ...)] SNAP SDATA VALUES ALL = all of the following fields NUC = all of nucleus except trace table TRT = trace table CB = TCB, active RBs, JPACQ, and MSS control blocks SNAP PDATA VALUES ALL = all of the following fields (assume SA and ALLPA) PSW = Program Status Word when SNAP was issued REGS = contents of general registers when SNAP was issued SA = linkage information and back trace SAH = linkage information only JPA = all virtual storage assigned to job step LPA = contents of resident reenterable load module ALLPA = JPA + LPA SPLS = contents of virtual storage subpools 0 - 127 [,STORAGE = (starting address, ending address,...) / ,LIST = address of list]
Record trace data	GTRACE	DATA=address,LNG=number of bytes of data,ID=record ID [,FID=format identifier routine]

Termination

Terminate normally	RETURN	[(range of registers to be restored)] [,T] [{ ,RC=0 to 4095 / ,RC=(15) }] In RETURN, T means: place all ones in high-order byte of save area word 4.
Terminate abnormally	ABEND	0 to 4095, [DUMP] [,STEP]

Task Control

(see explanation of style - page 3-27)

Dynamically load and initiate execution	ATTACH	{ EP=entry point name EPLOC=address of entry point name+ DE=address of name field of list entry } [,DCB=dcb address] [,PARAM=(address parameter [,address parameter]...) [,VL=1]] [,ECB=ecb address] [,ETXR=address of routine to be entered when subtask terminates [,LPMOD=number subtracted from limit priority] [,DPMOD=signed number algebraically added to dispatching priority] [,TOE={YES/NO}] [,FPREGSA={YES/NO}]
Delete	DETACH	address of tcb address*
Change priority	CHAP	signed number to be algebraically added to dispatching priority [,address of tcb address ,'S'] 'S' indicates that the priority of the active task is to be changed.

Virtual Storage Allocation

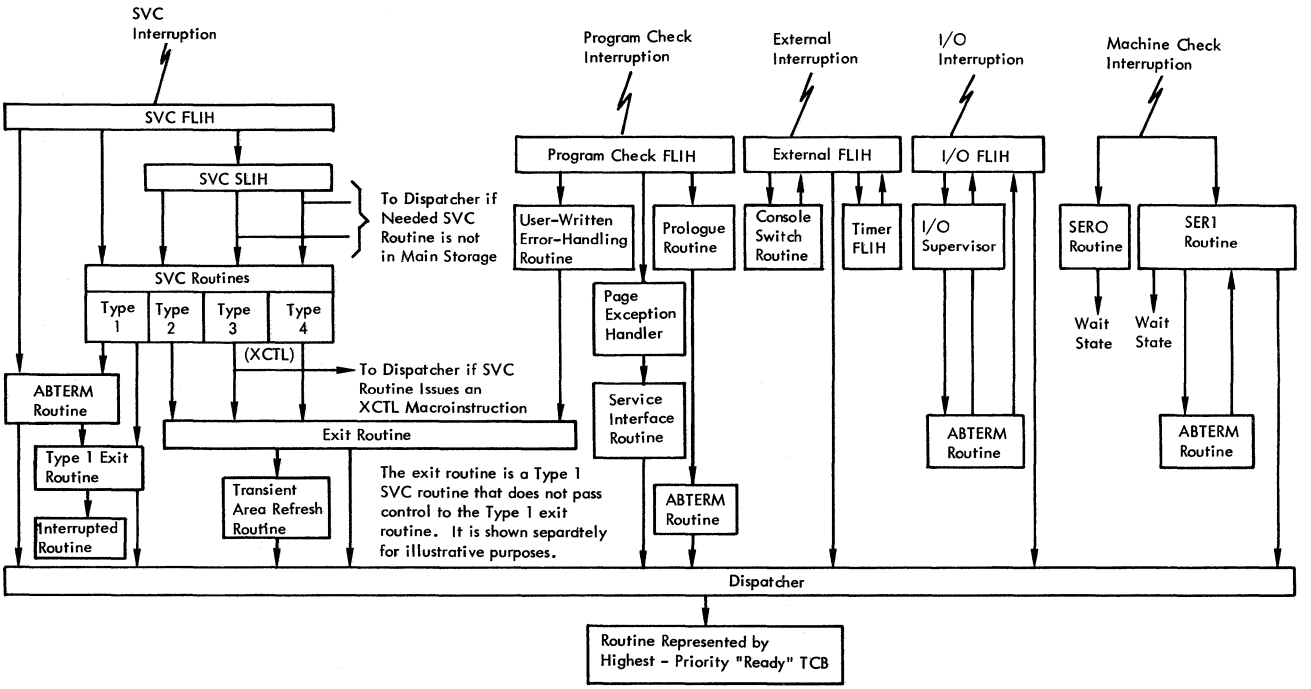
(see explanation of style - page 3-27)

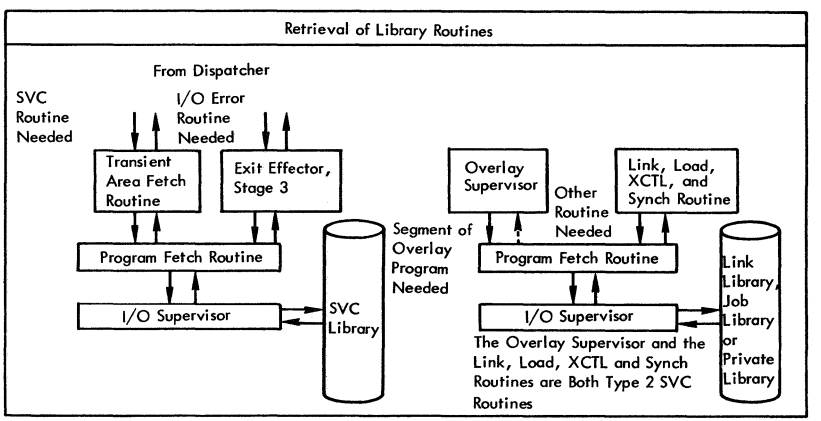
Allocate storage	GETMAIN	R, LV=length° [, SP=0 to 127]
	GETMAIN	{ EC / EU }, LV=length° / { VC / VU }, LA=address of length° list , A=address of specification list [, SP=0 to 127] [, BNDRY={ DBLWD / PAGE }]
Release storage	FREEMAIN	{ R, LV=Length°, A=address of storage area address* list [, SP=0 to 127] / R, SP=(0) }
	FREEMAIN	{ E, LV=length° / V }, A=address of storage area address* list [, SP=0 to 127]
Release virtual storage	PGRLSE	LA=low address of area, HA=high address*1 of area

MODE OPERANDS FOR GETMAIN AND FREEMAIN

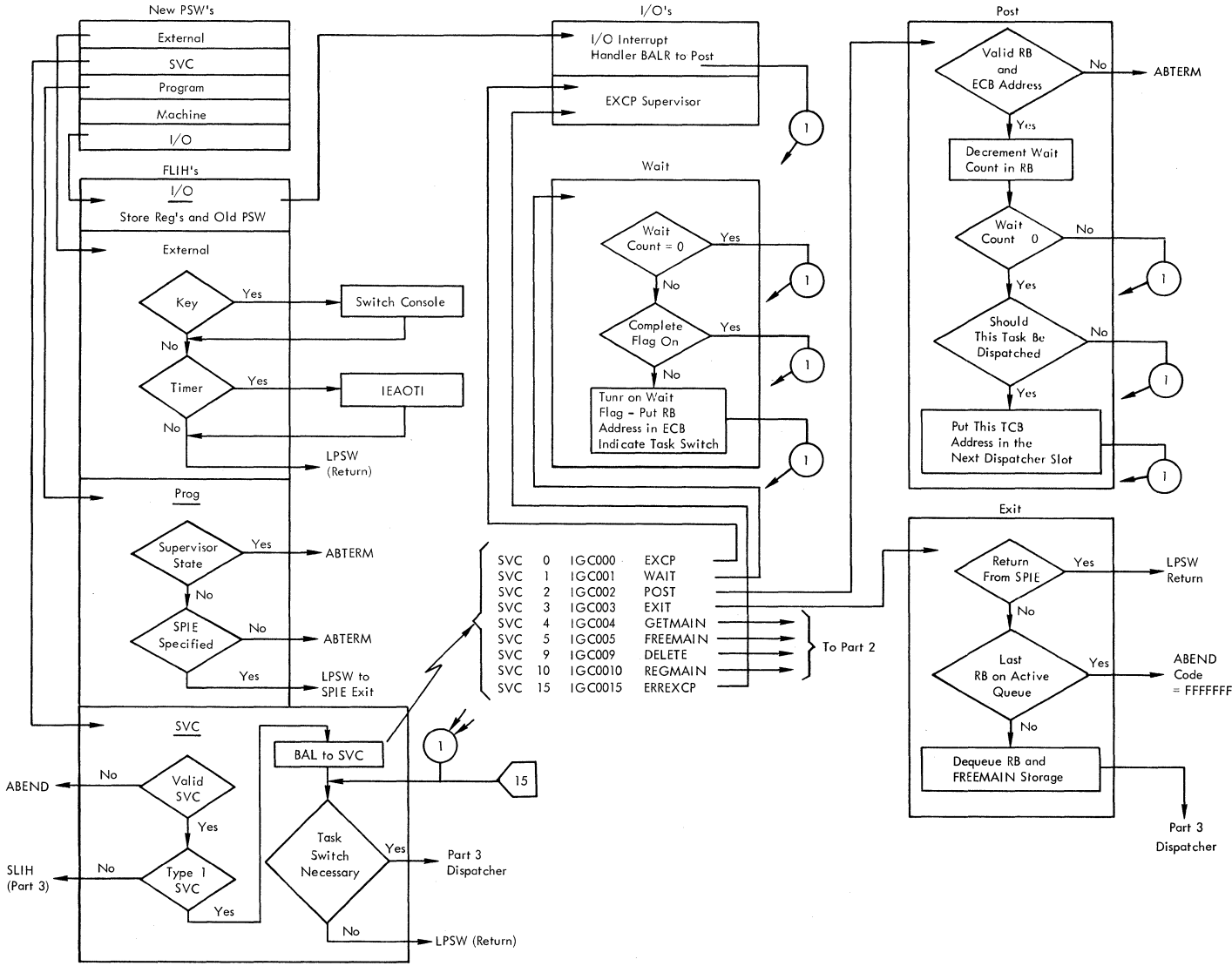
R=register type
E=single area, fixed length
V=single area, variable length
U=unconditional
C=conditional

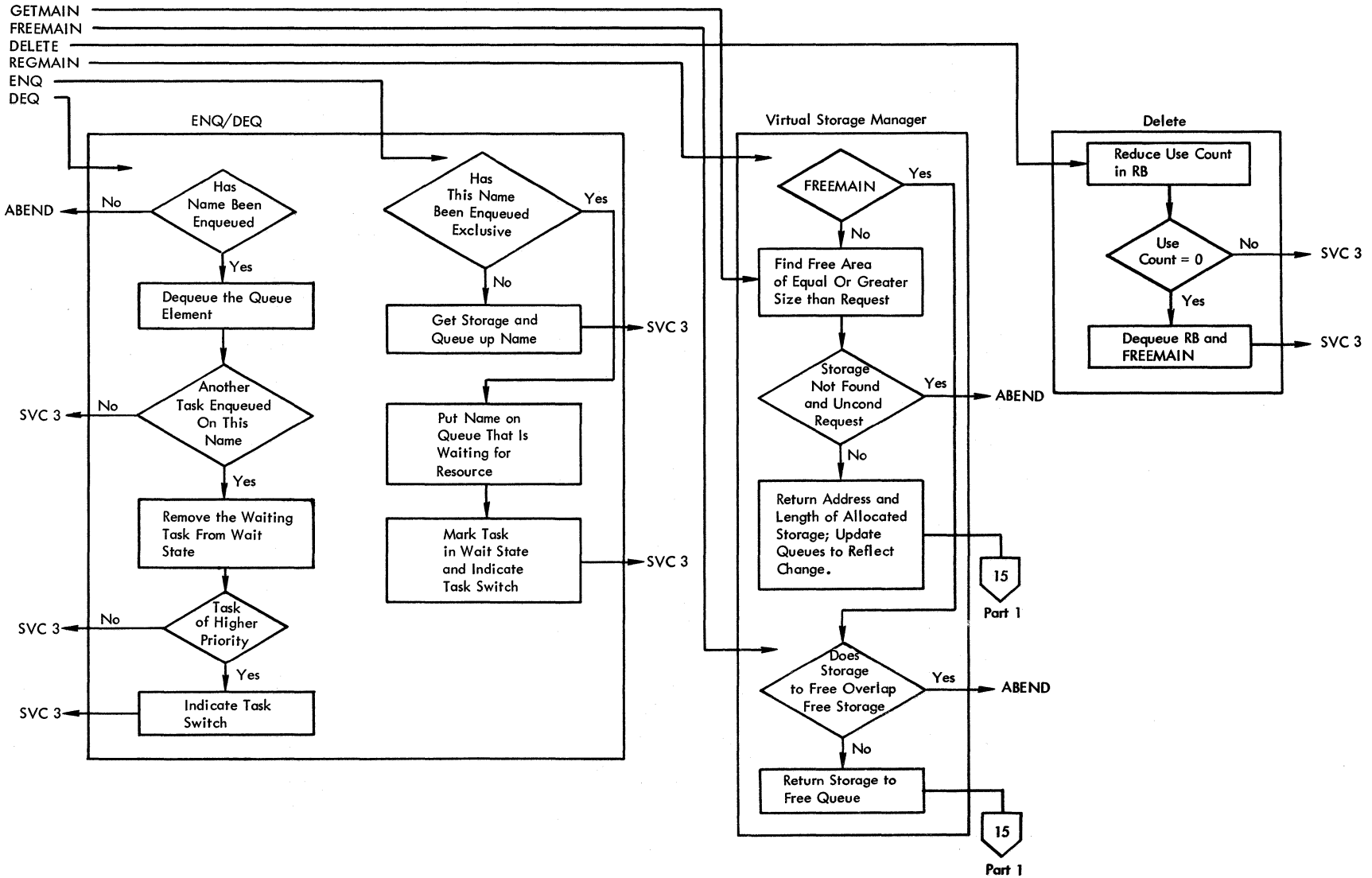
Overall Control Flow of Supervisor

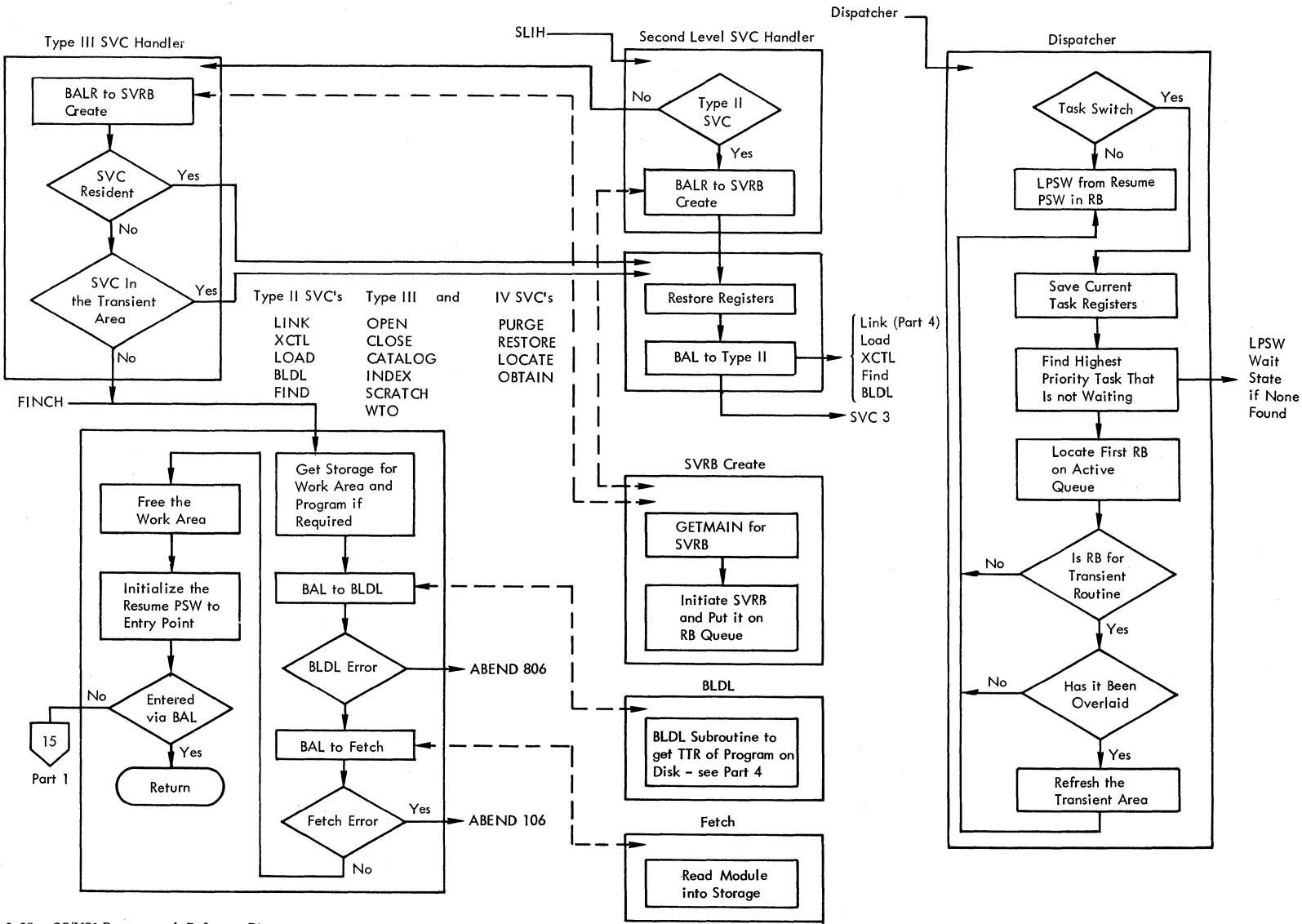


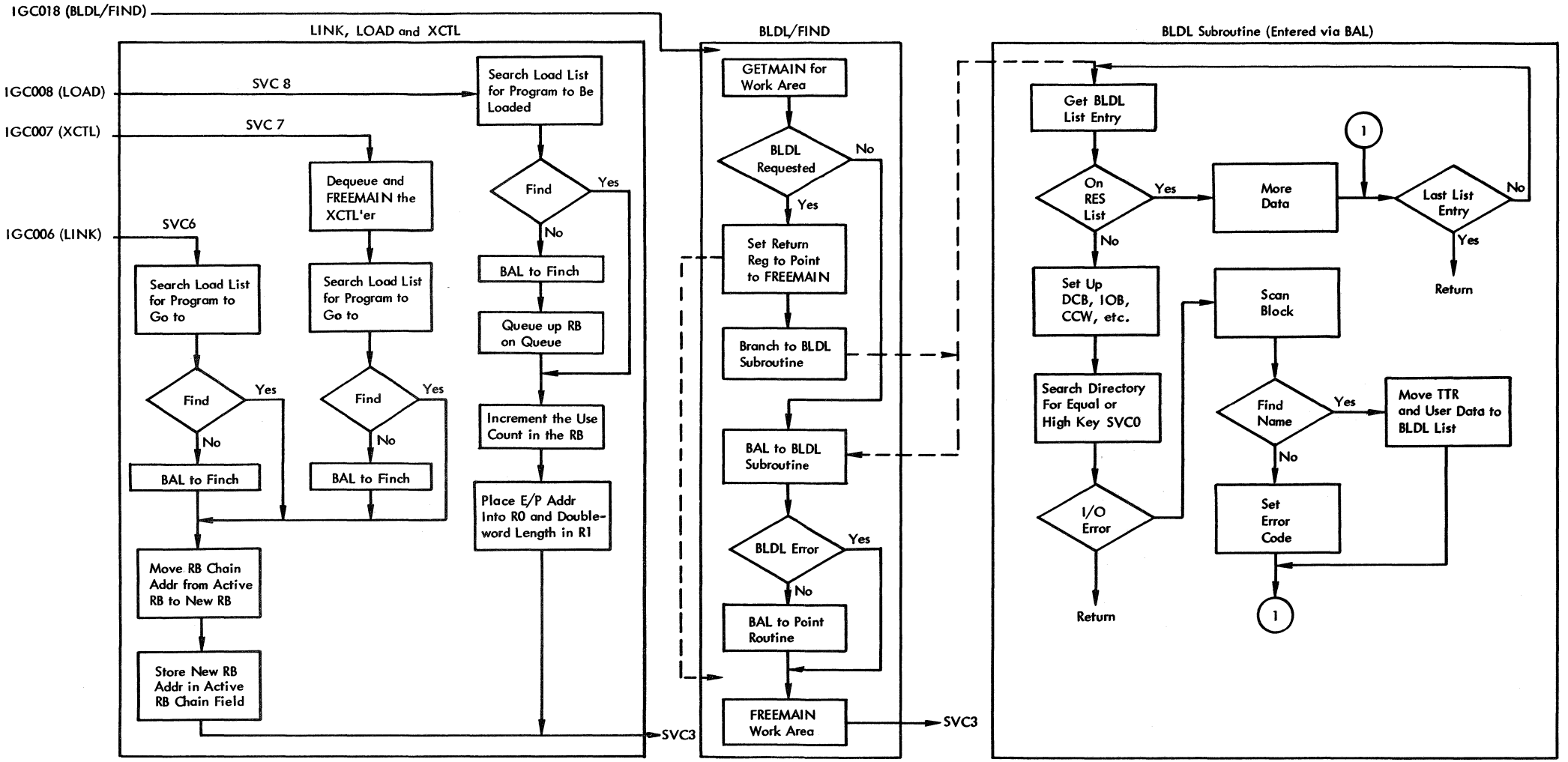


General Flow Diagrams (Part 1 of 4)




GETMAIN
FREEMAIN
DELETE
REGMAIN
ENQ
DEQ
ENQ/DEQ
Has Name Been Enqueued
No
ABEND
Yes
Dequeue the Queue Element
Another Task Enqueued On This Name
No
SVC 3
Yes
Remove the Waiting Task From Wait State
Task of Higher Priority
No
SVC 3
Yes
SVC 3
Indicate Task Switch
Has This Name Been Enqueued Exclusive
Yes
No
Get Storage and Queue up Name
SVC 3
Put Name on Queue That Is Waiting for Resource
Mark Task in Wait State and Indicate Task Switch
SVC 3
Virtual Storage Manager
FREEMAIN
Yes
No
Find Free Area of Equal Or Greater Size than Request
Storage Not Found and Uncond Request
Yes
ABEND
No
Return Address and Length of Allocated Storage; Update Queues to Reflect Change.
15
Part 1
Does Storage to Free Overlap Free Storage
Yes
ABEND
No
Return Storage to Free Queue
15
Part 1
Delete
Reduce Use Count in RB
Use Count = 0
No
SVC 3
Yes
Dequeue RB and FREEMAIN
SVC 3


Type III SVC Handler
BALR to SVRB Create
SVC Resident
Yes
No
SVC In the Transient Area
Yes
No
FINCH
SLIH
Second Level SVC Handler
No
Type II SVC
Yes
BALR to SVRB Create
Restore Registers
BAL to Type II
Link (Part 4)
Load
XCTL
Find
BLDL
SVC 3
Type II SVC's
LINK
XCTL
LOAD
BLDL
FIND
Type III and IV SVC's
OPEN
CLOSE
CATALOG
INDEX
SCRATCH
WTO
PURGE
RESTORE
LOCATE
OBTAIN
Free the Work Area
Initialize the Resume PSW to Entry Point
No
Entered via BAL
Yes
Return
15
Part 1
Get Storage for Work Area and Program if Required
BAL to BLDL
BLDL Error
ABEND 806
BAL to Fetch
Fetch Error
Yes
ABEND 106
No
SVRB Create
GETMAIN for SVRB
Initiate SVRB and Put it on RB Queue
BLDL
BLDL Subroutine to get TTR of Program on Disk - see Part 4
Fetch
Read Module into Storage
Dispatcher
Dispatcher
Task Switch
Yes
No
LPSW from Resume PSW in RB
Save Current Task Registers
Find Highest Priority Task That Is not Waiting
LPSW Wait State if None Found
Locate First RB on Active Queue
No
Is RB for Transient Routine
Yes
No
Has it Been Overlaid
Yes
Refresh the Transient Area


IGC018 (BLDL/FIND)
LINK, LOAD and XCTL
BLDL/FIND
BLDL Subroutine (Entered via BAL)
IGC008 (LOAD)
SVC 8
IGC007 (XCTL)
SVC 7
IGC006 (LINK)
SVC6
Search Load List for Program to Be Loaded
Find
Yes
No
BAL to Finch
Queue up RB on Queue
Increment the Use Count in the RB
Place E/P Addr Into R0 and Double-word Length in R1
Dequeue and FREEMAIN the XCTL'er
Search Load List for Program to Go to
Find
Yes
No
BAL to Finch
Search Load List for Program to Go to
Find
Yes
No
BAL to Finch
Move RB Chain Addr from Active RB to New RB
Store New RB Addr in Active RB Chain Field
SVC3
GETMAIN for Work Area
BLDL Requested
No
Yes
Set Return Reg to Point to FREEMAIN
Branch to BLDL Subroutine
BAL to BLDL Subroutine
BLDL Error
Yes
No
BAL to Point Routine
FREEMAIN Work Area
SVC3
Get BLDL List Entry
On RES List
Yes
No
More Data
1
Last List Entry
No
Yes
Return
Set Up DCB, IOB, CCW, etc.
Search Directory For Equal or High Key SVC0
I/O Error
Return
Scan Block
Find Name
Yes
No
Move TTR and User Data to BLDL List
Set Error Code
1

This section contains information that is mainly of interest to the systems or applications programmer who works with programs that require data management support.

Source Publications

Details of data management macros for BSAM, BDAM, BPAM, BISAM, QSAM, and QISAM, as well as DASD track capacities, are found in *OS/VS Data Management Macro Instructions,* GC26-3793.

You can obtain additional tape label information from

- *OS/VS Tape Labels,* GC26-3795

Data set record format information is available in *OS/VS Data Management Services Guide,* GC26-3783.

Additional VSAM information is available in *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide,* GC26-3838; *OS/VS1 Access Method Services,* GC26-3840; and *OS/VS Virtual Storage Access Method (VSAM) Options for Advanced Applications,* GC26-3819.

For information about MICR/OCR data management refer to these publications:

- *OS Data Management Services and Macro Instructions for IBM 1419/1275,* GC21-5006
- *OS Data Management Services and Macro Instructions for IBM 1285/1287/1288,* GC21-5004

Data Management Macros - Introduction

Data Management Macros for:

BDAM	BISAM
BSAM	QSAM
BPAM	QISAM

Completion codes for D/M macros are contained in the low-order byte of general register 15. Unless otherwise indicated the letter codes used here mean:

A - Successful completion.

B - Completion, but one or more errors occurred that may invalidate the results of macro execution.

C - Permanent I/O error

D - Track, block, or device address not within data set.

E - Not complete or no operation performed.

Introduction

A/M	Macro	Parameters	Completion Codes
BPAM	BLDL	{dcb address (1-12) (0)} , {list address (2-12) (0)}	00=A 04=B 08=C
BSAM	BSP	{dcb address (1-12)}	00=A 04=B 08=E (SYSIN or SYSOUT)
BDAM BISAM BPAM BSAM QISAM QSAM	BUILD	{area address, (1-12)} {{{number of buffers (2-12)} , {buffer length (2-12)}} (0)}	
QSAM	BUILDRCD	{area address (2-12)} , {number of buffers (2-12)} , {buffer length (2-12)} , {record area address (2-12)} [, {record area length (2-12)}]	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Code
QSAM	BUILDRCD (list form)	area address, number of buffers, buffer length ,record area address[,record area length],MF=L	
	BUILDRCD (execute form)	[area address], [number of buffers], [buffer length] (2-12) ,[record area address], [record area length] (2-12) ,MF=(E,{control program list address}) (1)	
BDAM BISAM BPAM BSAM	CHECK	{decb address}[,DSORG={IS}] (1-12) {ALL}	
BDAM BISAM BPAM BSAM QISAM QSAM	CHKPT	{dcb address[,checkid address[,checkid length['S']]]} {CANCEL}	00=Successful completion 04=Restart occurred 08=Unsuccessful completion: Macro error 0C=Unsuccessful completion: I/O error 10=Successful completion: Possible error 14=Chkpt not taken
	CHKPT (list form)	[dcb address],[checkid address],[checkid length['S']] ,MF=L	
	CHKPT (execute form)	[dcb address],[checkid address],[checkid length['S']] ,MF=(E,{control program list address}) (1)	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BDAM BISAM BPAM BSAM* QISAM QSAM	CLOSE	(dcb address [, REREAD / , LEAVE / , REWIND] [, DISP= (PASS / DELETE / KEEP / CATLG / UNCATLG)] , ...) [, TYPE=T]*	
	CLOSE (list form)	([dcb address] [, REREAD / , LEAVE / , REWIND] [, DISP= (PASS / DELETE / KEEP / CATLG / UNCATLG)] , ...) [, TYPE=T]*, MF=L	
	CLOSE (execute form)	[([dcb address] [, REREAD / , LEAVE / , REWIND] [, DISP= (PASS / DELETE / KEEP / CATLG / UNCATLG)] , ...)] [, TYPE=T]*, MF=(E, {data management list address (1-12)})	
BSAM QSAM	CNTRL	dcb address, { SS, {1 / 2} / SP, {1 / 2 / 3} / SK, {1 through 12} / BSM / FSM / BSR [, number of blocks] / FSR [, number of blocks] / ESP† / LMK† / DMK† / DSG / ENG }	Not available to user program

† See OS/VS IBM 3886 Optical Character Reader Model 1 Reference.

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BDAM BISAM BPAM BSAM QISAM QSAM	DCB	The DCB macro is too complex to properly document in this publication. For a list of the proper parameters to use, see the section following, DCB parameters. For detailed information about this macro, please refer to OS/VS Data Management Macro Instructions, GC26-3793.	
BDAM BISAM BPAM BSAM QISAM QSAM	DCBD	[DSORG=([BS][,DA][,IS][,LR][,PO][,PS][,QS]) GS] [,DEVD=([DA][,PC][,PR][,PT][,RD][,TA][,MR][,OR])]	
QISAM	ESETL	{dcb address} {(1-12)}	
BSAM QSAM	FEOV	{dcb address}[,{REWIND}] {(1-12)} [,{LEAVE }]	
BPAM	FIND	{dcb address}, {{name address} , D {(1-12)} {(2-12)} {(0)} {relative address list} ,C {(2-12)} {(0)}}	00=A 04=B 08=C Note: reladr, C always returns CC of 00
BDAM BISAM BPAM BSAM	FREEBUF	{dcb address} ,register* {(1-12)} *Note: Reg, any of 2 to 12, contains addr of buffer.	
BDAM BISAM	FREEDBUF	{decb address} ,{K} ,{dcb address} {(2-12)} {D} {(1-12)} {(0)}	
BDAM BISAM BPAM BSAM QISAM QSAM	FREEPOOL	{dcb address} {(1-12)}	
QISAM QSAM	GET	{dcb address } {pdab address}[,{area address}][,TYPE=P] {(1-12) } {(2-12)} {(0)}	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BDAM BISAM BPAM BSAM	GETBUF	{dcb address / (1-12)} ,register* *Note: Reg, any of 2 to 12, is where the system will place the buffer address.	
BDAM BISAM BPAM BSAM QISAM QSAM	GETPOOL	{dcb address / (1-12)}, {{number of buffers / (2-12) / (0)}, {buffer length / (2-12)}}	
BPAM BSAM	NOTE	{dcb address / (1-12)}	
BDAM BISAM BPAM BSAM QISAM QSAM	OPEN	({dcb address / (2-12)}, (options)],...)	
BDAM BISAM BPAM BSAM QISAM QSAM	OPEN (list form)	([dcb address],[(options)],...),MF=L	
	OPEN (execute form)	[([{dcb address / (2-12)}],[(options)],...)] ,MF=(E,{data management list address / (2-12)})	

Open Macro Options

ACCESS METHOD	DEVICE TYPE: MAGNETIC TAPE Option 1	MAGNETIC TAPE Option 2	DIRECT ACCESS Option 1	DIRECT ACCESS Option 2	OTHER TYPES Option 1	OTHER TYPES Option 2
QSAM	[INPUT / EXTEND / OUTPUT / RDBACK]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / EXTEND / OUTPUT / UPDAT]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / EXTEND / OUTPUT]	—
BSAM	[INPUT / EXTEND / OUTINX / OUTPUT / INOUT / OUTIN / RDBACK]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / EXTEND / OUTINX / OUTPUT / INOUT / OUTIN / UPDAT]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / EXTEND / OUTPUT]	—
QISAM (Load Mode)	—	—	[EXTEND / OUTPUT]	—	—	—
BPAM, BDAM	—	—	[INPUT / OUTPUT / UPDAT]	—	—	—

Optionally select one from vertical stack within [ ]

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
QSAM	PDAB	MAXDCB = dcb number	
QSAM	PDABD		
BPAM BSAM	POINT	{ dcb address (1-12) }, { block address (2-12) (0) }	
BSAM QSAM	PRTOV	{ dcb address (2-12) }, { 9 12 } [, { overflow exit address (2-12) }]	
QISAM QSAM	PUT	{ dcb address (1-12) } [, { area address (2-12) (0) }]	
QISAM QSAM	PUTX	{ dcb address (1-12) } [, { input dcb address (2-12) (0) }]	
BDAM	READ	decb name, { DI / DK / DIF / DIX / DKF / DKX } [R / RU] , { dcb address (2-12) }, { area address (2-12) 'S' } , { length (2-12) 'S' }, { key address (2-12) 'S' 0 }, { block address (2-12) } [, { next address (2-12) }]	
BSAM for BDAM data set	READ	decb name, SF, { dcb address (2-12) }, { area address (2-12) }	
BISAM	READ	decb name, { K / KU }, { dcb address (2-12) }, { area address (2-12) 'S' }, { length (2-12) 'S' } { key address (2-12) }	
BPAM BSAM	READ	decb name, { SF / SB / RBL† }, { dcb address (2-12) }, { area address (2-12) }, { length (2-12) 'S' }	
	READ (list form)	decb name, type*, [dcb address 'S'], [area address 'S'], [length 'S'] [key address 'S'], [block address], [next address], MF=L *Note: type will be one of the parameters (e.g., K, SF, DI) from the applicable standard form of the READ macro. †See OS/VS IBM 3886 Optical Character Reader Model 1 Reference.	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
	READ (execute form)	{decb address (2-12)}, type*, [dcb address (2-12)], [area address (2-12) 'S'], [length 'S'] , [key address (2-12) 'S'], [block address (2-12)], [next address (2-12)], MF=E *Note: type will be one of the parameters (e.g., K, SF, DI) from the applicable standard form of the READ mnemonic.	
BDAM	RELEX	D, {dcb address (1-12)}, {block address (2-12) (0)}	00=A 04=B 08=D
QISAM QSAM	RELSE	{dcb address (1-12)}	
QISAM	SETL	{dcb address (1-12)}, {K[H] KC KD[H] KCD I ID B BD}, {lower limit address , 2-12 , (0)}	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BSAM QSAM	SETPRT	{dcb address (2-12)} [,BURST = {N | Y}] [,CHARS = {{name | A (address) | R (register)} ({name | A (address) | R (register)),...)}] [,COPIES = number] [,COPYNR = number] [,FCB = {{imageid | A (address) | R (register)} ({imageid | A (address) | R (register)},{V | A})}] [,FLASH = {name ((name),count)}] [,INIT = {N | Y}] [,MODIFY = {{name | A (address) | R (register)} ({name | A (address) | R (register)},trc)}] [,OPTCD = {B | U ({B | U},{F | U})}] [,REXMIT = {N | Y}] [,UCS = {csc (csc,{F | F,V | ,V})}]	

SETPRT Completion Codes

3800 Printer Bits 8-15	FCB Bits 16-23	UCS Bits 24-31		FCB/UCS Bits 24-31	
00	00	00	Successful completion	18	NOP: incorrect specification
	04	04	Operator cancellation	1C	NOP: error during previous I/O attempt
04*			Image not found in image library	20	Insufficient space for SYS1.IMAGELIB blocks
08*	08	08	Permanent I/O error in image library	24	Unable to open SYS1.IMAGELIB
0C*	0C	0C	Permanent I/O error during load		
10	10	10	Permanent I/O error during image display		
	14	14	Operator cancellation: incorrect image		

3800 Printing Subsystem

Bits 8-15	
28	Operator canceled forms overlay request
2C	Operator canceled page threading request
30	Too many writable character generation modules requested
34	Invalid table reference character
38	Error while initializing 3800
40	Permanent I/O error during command execution
44	Unspecified character set referenced

* 3800 Printing Subsystem Reason Codes (Reg 0)

04	Character arrangement table module/record
08	Copy modification module/record
0C	Starting copy number
10	Graphic character modification module/record
14	Forms overlay sequence control record
1C	Writable character generation module (WCGM)
20	Forms control buffer module

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BSAM QSAM	SETPRT (list form)	dcb address [,BURST={N | Y}] [,CHARS=(name,...)] [,COPIES=number] [,COPYNR=number] [,FCB=(imageid[,V],A)]] [,FLASH=(name[,count])] [,INIT={N | Y}] [,MODIFY=(name[,trc])] [,OPTCD=({B | U}[,F | ,U])] [,REXMIT={N | Y}] [,UCS=(csc[,F],F,V | ,,V)] ,MF=L	
BSAM QSAM	SETPRT (execute form)	{dcb address / (2-12)} [,BURST={N | Y | *}] [,CHARS={{name A (address) R (register)}, / *(name A (address) R (register)),...)}] [,COPIES={number | *}] [,COPYNR={number | *}] [,FCB={{imageid | A (address) | R (register)} / *({imageid | A (address) | R (register)},(V | A))}] [,FLASH={name / *(name, count)}] [,INIT={N | Y}] [,MODIFY={{name | A (address) | R (register)} / *(name | A (address) | R (register)), trc)}] [,OPTCD={{B | U} / (B | U , F | U)}] [,REXMIT={N | Y | *}] [,UCS={csc / *(csc,(F | F,V | ,V))}] ,MF=(E,{data management list address (1-12)})	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BPAM	STOW	{dcb address (1-12)}, {list address (2-12) (0)} [, {A C D R}]	

Comp. Code (hex)	Directory Action A	R	D	C
00	Successful completion			
04	Name already in directory	—	—	New name already in directory
08	—	Name not found		Old name not found
0C	No space in directory		—	No space in directory
10	Permanent I/O error in directory			
14	Specified data control block not open			
18	Insufficient virtual storage			

BDAM BISAM BPAM BSAM QISAM QSAM EXCP	SYNADAF	{ACSMETH=BDAM / ACSMETH=BPAM / ACSMETH=BSAM / ACSMETH=QSAM / ACSMETH=BISAM} [,PARM1=parm reg*][,PARM2=parm reg**] ACSMETH=EXCP [,PARM1 = {iob address (1-12)}] ACSMETH=QISAM[,PARM1 ={dcb address (1-12)}][,PARM2=parm reg**] *Note: Any of registers 1 to 12 **Note: Any of registers 0 or 2 through 12.	00=A with no msg 04=A with msg. 08=B
BDAM BISAM BPAM BSAM QISAM QSAM EXCP	SYNADRLS		00=A 08=E

A/M	Macro	Parameters	Completion Codes
QSAM	TRUNC	{dcb address / (1-12)}	
BDAM BISAM BPAM BSAM	WAIT	[number of events,] {ECB = addr / ECBLIST = addr} [,LONG={YES / NO}]	
BSAM	WRITE	decb name, {SF / SFR / SD / SZ}, {dcb address (2-12)}, {area address (2-12)} [,length / ,(2-12) / ,'S'] [,next address / ,(2-12)]	

Meaning

Code	Fixed-Length (SF or SD)	Variable or Unspecified Length (SF or SFR)	Variable or Unspecified Length (SZ)
00	Block written. (If previous return code was 08, block written only if the DD statement specifies secondary space allocation and sufficient space available.)		Capacity record written; another track available.
04	Block written, followed by capacity record.	Block not written; write a capacity record (SZ) to complete current track, then reissue.	
08	Block written, followed by capacity record. Next block requires secondary space allocation.		Capacity record was written. Next block requires secondary space allocation. This code not issued if WRITE SZ is only WRITE macro instruction issued on a one-track secondary extent.
0C	Block will not be written; issue a CHECK macro instruction for the previous WRITE macro, then reissue the WRITE.		

BDAM	WRITE	decb name, {DA / DAF / DI / DIF / DIX / DK / DKF / DKX}, {dcb address (2-12)}, {area address (2-12) / 'S'} {length 'S' (2-12)}, {key address (2-12) 'S' 0}, {block address (2-12)}	
BISAM	WRITE	decb name, {K / KN}, {dcb address (2-12)}, {area address (2-12) 'S'} {length (2-12) 'S'}, {key address (2-12)}	

Data Management Macros (cont'd)

A/M	Macro	Parameters	Completion Codes
BPAM BSAM	WRITE	decb name, SF, {dcb address / (2-12)}, {area address / (2-12)} [, {length / (2-12) / 'S'}]	
	WRITE (list form)	decb name, type*, [dcb address], [area address / 'S'], [length / 'S'], [key address / 'S'], [block address], [next address], MF=L *Note: type will be one of the keyword parameters (e.g., SF, DA, K) from the applicable standard form of the WRITE macro.	
	WRITE (execute form)	{decb address / (2-12)}, type*, [dcb address / (2-12)], [area address / (2-12) / 'S'], [length / (2-12) / 'S'], [key address / (2-12) / 'S'], [block address / (2-12)], [next address / (2-12)], MF=E *Note: type will be one of the keyword parameters (e.g., SF, DA, K) from the applicable standard form of the WRITE macro.	
BDAM BISAM BPAM BSAM QISAM QSAM	XLATE	{area address / (2-12)}, {length / (2-12)} [,TO={A / E}]	

DCB Parameters for Access Methods

Use these DCB parameters	for these access methods . . .									
	BDAM	BISAM	BPAM	BSAM	BTAM	EXCP	GAM	QISAM	QSAM	TCAM
BFALN	*	*	*	*		*		*	*	
BFTEK	*			*	*	*			*	*
BLKSIZE	*		*	*				*	*	*
BUFCB	*	*	*	*	*			*	*	
BUFIN										*
BUFL	*	*	*	*		*		*	*	*
BUFMAX										*
BUFNO	*	*	*	*	*	*		*	*	
BUFOFF				*					*	
BUFOUT										*
BUFSIZE										*
CODE				*	*	*			*	
CYLOFL								*		
DDNAME	*	*	*	*	*	*	*	*	*	*
DEN				*		*			*	
DEVD				*	*					
DIAGNS	*	*	*	*	*	*	*	*	*	
DSORG	*	*	*	*	*	*	*	*	*	
EODAD			*	*				*	*	*
EROPT					*				*	
EXLST	*	*	*	*	*	*	*	*	*	*
FRID				*						
FUNC				*					*	
GNCP							*			
GTYPE							*			
KEYLEN	*		*	*		*		*		
LERB					*					
LIMCT	*									
LRECL			*	*				*	*	*
MACRF	*	*	*	*	*	*	*	*	*	*
MODE				*	*	*				
MSHI		*								
MSWA		*								
NCP		*	*	*						
NTM								*		
OPTCD	*		*	*		*		*	*	*
PCI										*
POLST							*			
PRTSP				*		*			*	
READYQ					*					
RECFM	*		*	*				*	*	*
RESERVE										*
RKP								*		
SMSW		*								
STACK				*		*			*	
STOP										*
SYNAD	*	*	*	*				*	*	*
THRESH										*
TRTCH				*		*			*	

Direct Access Device Capacities

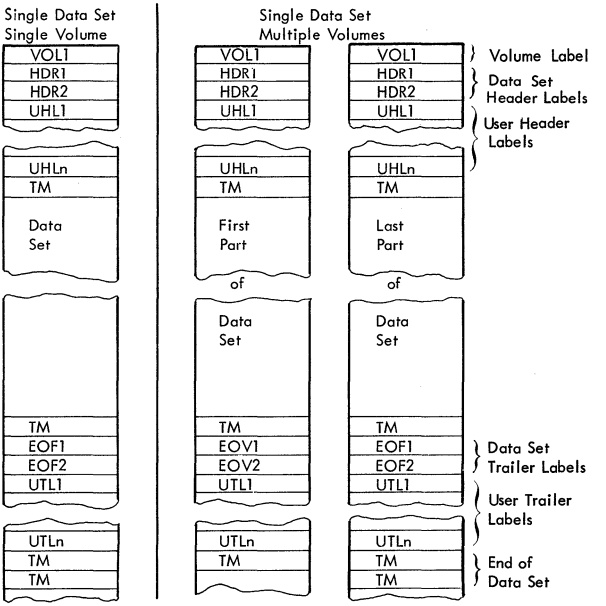
Device Type	Volume Type	Maximum Blocksize/ Track 1	Tracks/ Cylinder	No. of Cylinders 2	Total Capacity 1, 2
2314/ 2319	Disk	7294	20	200	29,176,000
3330/ 3333 (Model 1)	Disk	13030	19	404	100,018,280
3330/ 3333 (Model 11)	Disk	13030	19	808	200,036,560
3340	Disk	8368	12	696(70-megabytes) 348(35-megabytes)	69,889,536 34,944,768
3350	Disk	19069	30	555	317,498,850
2305-1	Drum	14136	8	48	5,428,224
2305-2	Drum	14660	8	96	11,258,880

1- Capacity indicated in bytes (when R0 is used by the IBM programming system).
2- Excluding alternate cylinders.

Device Type	Blocks with keys		Blocks without keys	
	Bi	Bn	Bi	Bn
2314/ 2319	$146+\frac{534}{512}$ (KL+DL)	45+KL+DL	$101+\frac{534}{512}$ (DL)	DL
3330/ 3333 (Model 1)	191+KL+DL	191+KL+DL	135+DL	135+DL
3330/ 3333 (Model 11)	191+KL+DL	191+KL+DL	135+DL	135+DL
3340	242+KL+DL	242+DL+DL	167+DL	167+DL
3350	267+KL+DL	267+KL+DL	185+DL	185+DL
2305-1	634+KL+DL	634+KL+DL	432+DL	432+DL
2305-2	289+KL+DL	289+KL+DL	198+DL	198+DL

Bi is any block but the last on the track
Bn is the last block on the track
KL is the key length
DL is the data length

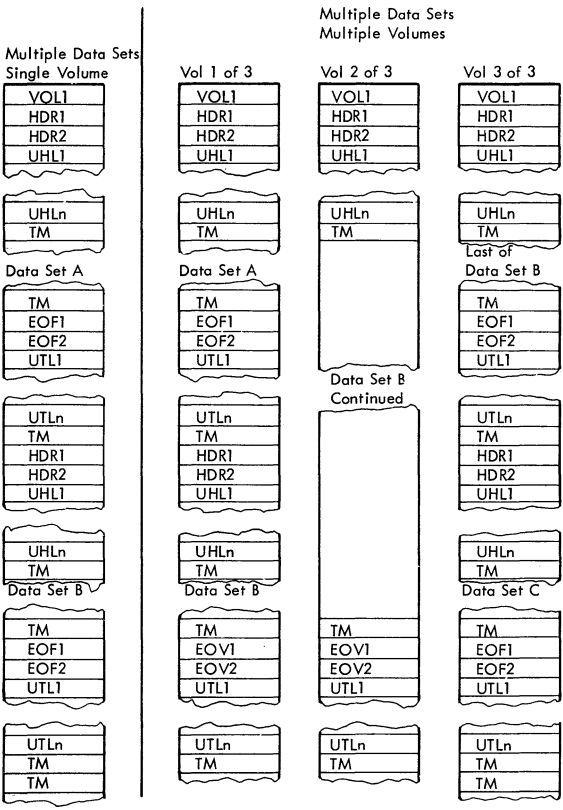
Volume Organization with IBM Standard Labels



Single Data Set/Single Volume: The volume label is followed by the data set header labels and optional user header labels. The data set is preceded and followed by a tapemark. The data set trailer labels are identified as EOF and followed by optional user trailer labels. Two tapemarks follow the trailer label group to indicate that the data set is the last data set on the volume and is not continued on another volume.

Single Data Set/Multiple Volumes: More than one volume is needed to contain the data set. The last volume is organized the same as a single volume. On the other volumes, the data set trailer labels are identified as EOV instead of EOF, and the trailer label group is followed by one tapemark instead of two. The data set and user labels are repeated on each volume, and there is a separate volume label for each tape.

Volume Organization with IBM Standard Labels (cont'd)



Multiple Data Sets/Single Volume: The tape begins with a volume label. Each data set is preceded by a header label group and a tapemark, and is followed by a tapemark and a trailer label group. The data set trailer labels are identified as EOF. Each trailer label group is followed by a tapemark; the tailer label group for the last data set on the volume is followed by two tapemarks.

Multiple Data Sets/Multiple Volumes: More than one volume is needed to contain the multiple data set aggregate. The last volume is organized the same as a multiple data set/single volume layout. On the other volumes, the last data set trailer labels are identified as EOV instead of EOF, and the last trailer label group is followed by one tapemark instead of two. There is a separate volume label for each tape.

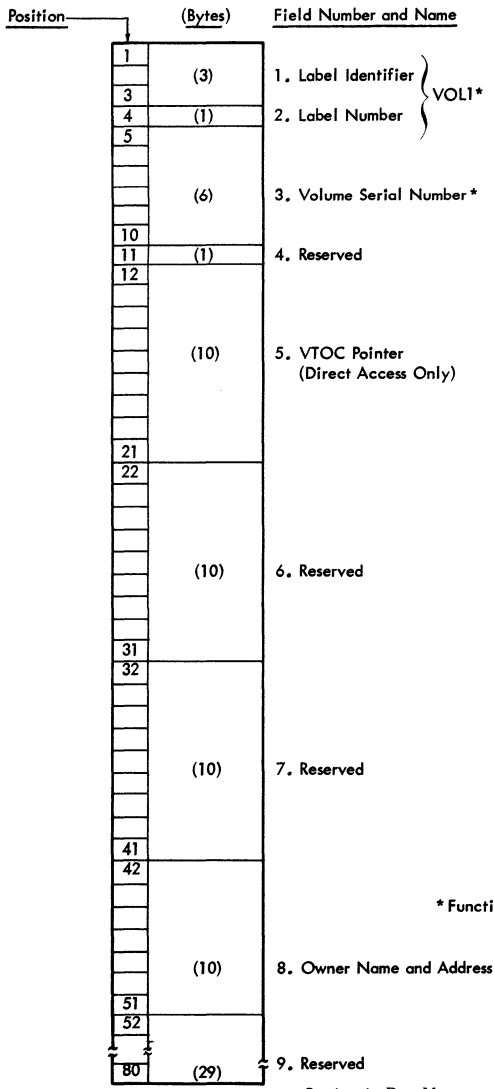
IBM Standard Label Processing by Data Management Routine

Processing	Volume Label	Header Labels[1]			Trailer Labels[1]		
	VOL1	HDR1	HDR2	UHL1-8	EOF1 or EOV1	EOF2 or EOV2	UTL1-8
First or Only Volume: 2							
Checks labels on input tape	Open	Open	Open	Open	EOV	bypassed	EOV
Checks existing labels on output tape before overwriting.	Open	Open	not read	not read	not read	not read	Open[5]
Writes new labels on output tape.	Open or user[4]	Open	Open	Open	Close or EOV	Close or EOV	Close or EOV
Second or Subsequent Volumes: 3							
Checks labels on input tape.	EOV	EOV	bypassed	EOV	EOV	bypassed	EOV
Check labels on output tape before overwriting.	EOV	EOV	not read	not read	not read	not read	not read
Writes new labels on output tape.	EOV or user[4]	EOV	EOV	EOV	Close or EOV	Close or EOV	Close or EOV

Notes:

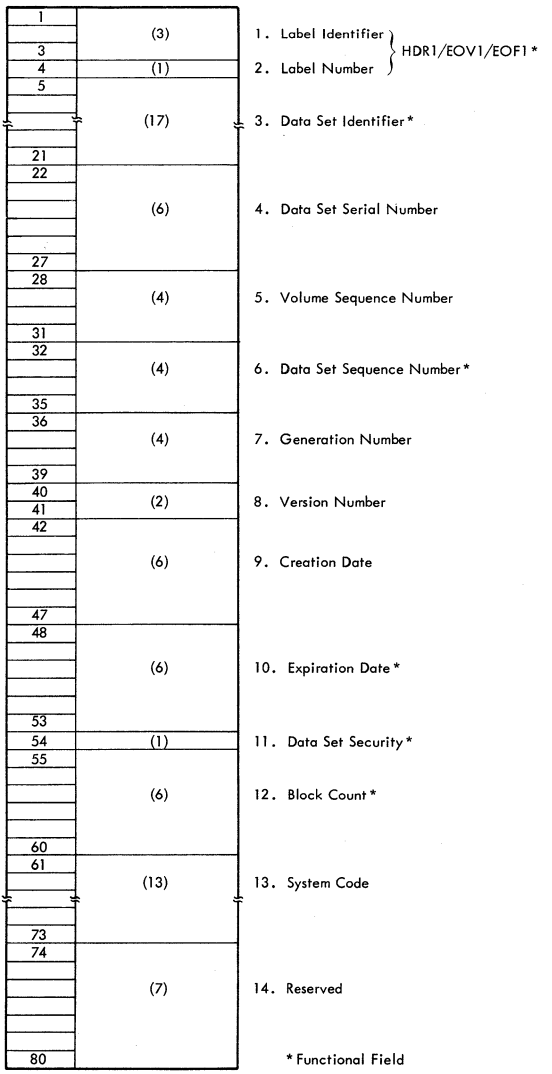
1. For read backward operations, the action on header and trailer labels is reversed.
2. Includes the first volume of concatenated data sets with unlike characteristics. Data sets with like characteristics can be processed correctly using the same data control block (DCB), input/output block (IOB), and channel program. Any exception in processing makes the data sets unlike.
3. Includes the first volume of concatenated data sets with like characteristics.
4. User can create the label with the IEHINITT utility program or a user program. Subsequently, the label may be rewritten by the Open and EOV routines.
5. If DISP=MOD is specified on the DD statement, the Open routine positions the tape at the end of the existing data set and allows an input user trailer label routine to process user trailer labels (prior to overwriting the existing labels).

IBM Standard Volume Label



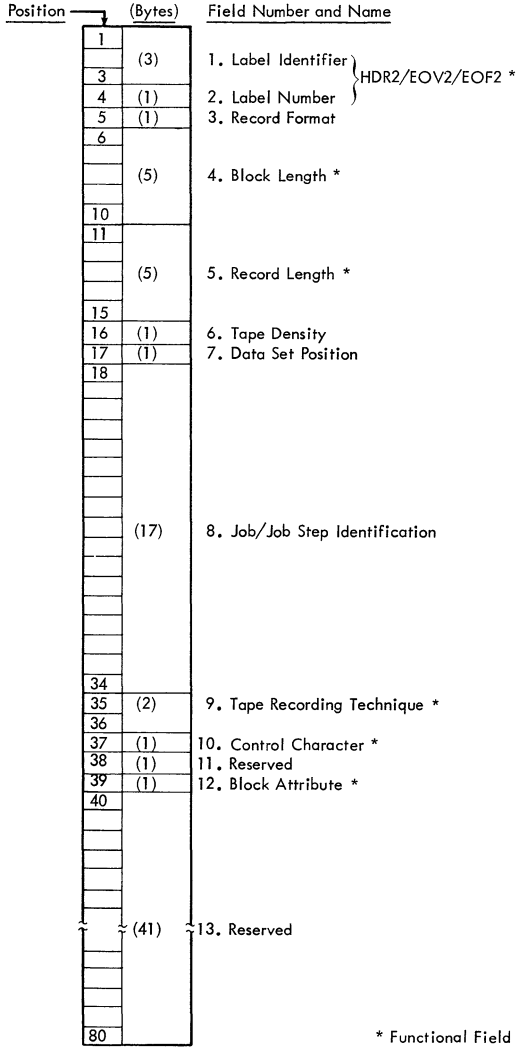
Label Formats (cont'd)

IBM Standard Data Set Label 1



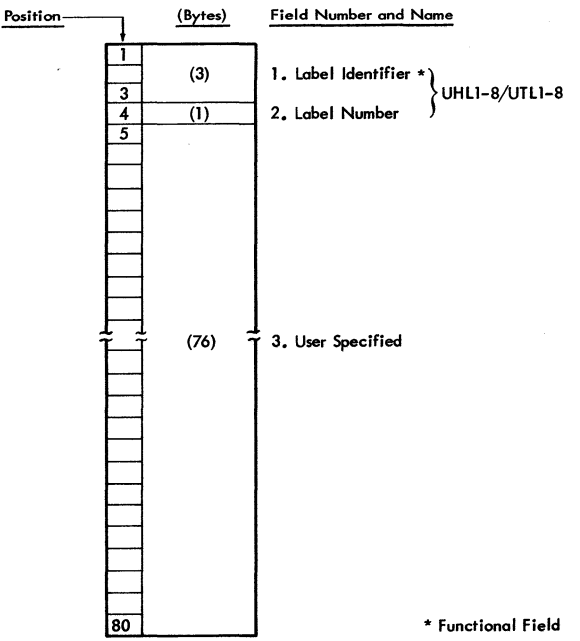
Label Formats (cont'd)

IBM Standard Data Set Label 2



Label Formats (cont'd)

User Label



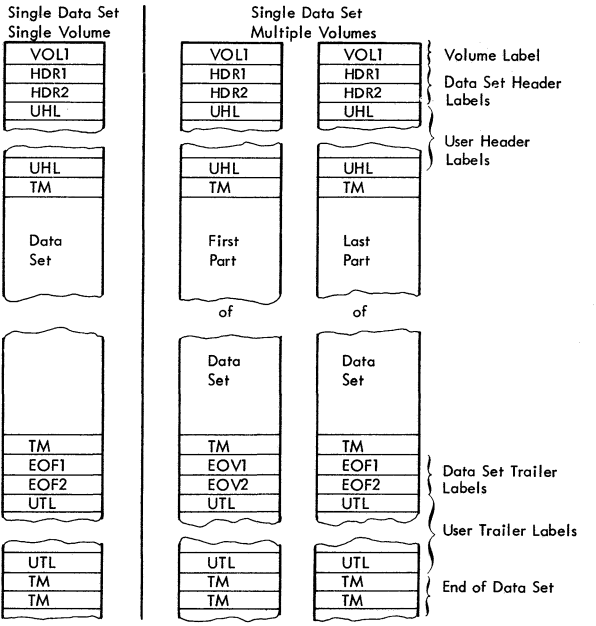
IBM 3540 Diskette Standard Volume (VOL1) Label (GC24-5110)

Position	Field Name	Contents/Use
1–4	Label - ID	Identifies the sector as a volume label and must be VOL1
5–10	Volume ID	The ID consists of from one to six alphameric characters. The characters must be left-justified.
11	Volume Accessibility	A blank permits access to the volume. A nonblank indicates authorization is required for further access to the volume. An operator message is issued requesting the nonblank character that allows further access to the volume.
80	Standard Label Version	Must be W to indicate IBM Standard Labels. The system processes only diskettes with IBM Standard Labels.

IBM 3540 Diskette Standard Header (HDR1) Label (GC24-5110)

Position	Field Name	Contents/Use
1–4	Label ID	Label identifier for system application. Must be HDR1 or DDR1.
6–13	Data Set Identifier	User assigned data set name. Must be one to eight alphameric characters, hyphens, or left braces. The first character must be alphabetic or national. The name must be left-justified and no duplicate data set identifiers should be on the same diskette.
23–27	Block/Record Length	Indicates the length of the data recorded in each of the 128-position sectors for the data set. Must be decimal characters greater than 0 and less than or equal to 128.
29–33	Beginning of Extent (BOE)	Identifies the address, track and sector, (ttss), of the first sector of the data set, where $01 \leq tt \leq 73$. $01 \leq ss \leq 26$, $BOE \leq EOE$, and $BOE \leq EOD$.
35–39	End of Extent (EOE)	Identifies the address (ttss) of the last sector of the data set, where $01 \leq tt \leq 73$, $01 \leq ss \leq 26$, $EOE \geq BOE$, EOE +1 address position $\geq$ EOD.
41	Bypass Indicator	Must contain blank or B. If set to blank, the data set is intended for processing. If set to B, the data set is not intended for processing even though it resides on the diskette (that is, the 3540 does not read the data set when a B is in this field).
42	Data Set Security Indicator	A blank permits access to the data set. A nonblank indicates authorization is required for further access to the data set. An operator message is issued requesting the nonblank character that allows further access to the data set. This field is processed only when the VOL1 accessibility indicator is a nonblank. It must be blank if the VOL1 accessibility indicator is blank.
43	Writer Protect Indicator	Must contain a blank or P. A blank indicates no write protection. P protects the data set from being overwritten. When P is specified for a data set, the diskette cannot be used to receive any SYSOUT data sets.
44	Interchange Type Indicator	Must be blank for Standard Data Interchange.
45	Multivolume Indicator	Must contain a blank, C, or L. A blank indicates the entire data set is on this diskette. C indicates the data set is continued on another volume. L indicates the data set is on the last volume.
46–47	Volume Sequence Number	Indicates the sequence in a multivolume data set. Must be blanks or 01–99. Blanks indicate no sequence checking is to be performed. Sequence numbers must be consecutive from 01–99.
67–72	Expiration Date	May be used to contain the date (yymmdd) that the data set and its labels may be purged. yy is the year, mm is the month, and dd is the day.
73	Verify Indicator	Must be blank, V, or C. A blank indicates the data set has not been verified or copied. V indicates the data set has been verified. C indicates the data set has been copied. If verify indicator processing is requested, this field must be a V; otherwise, the HDR1 label is considered invalid. This field is ignored when verify indicator processing is not performed.
75–79	End of Data (EOD)	Identifies the address (ttss) of the next unused section of the data set extent, where $EOD \geq BOE$, and EOE +1 address position $\geq$ EOD.

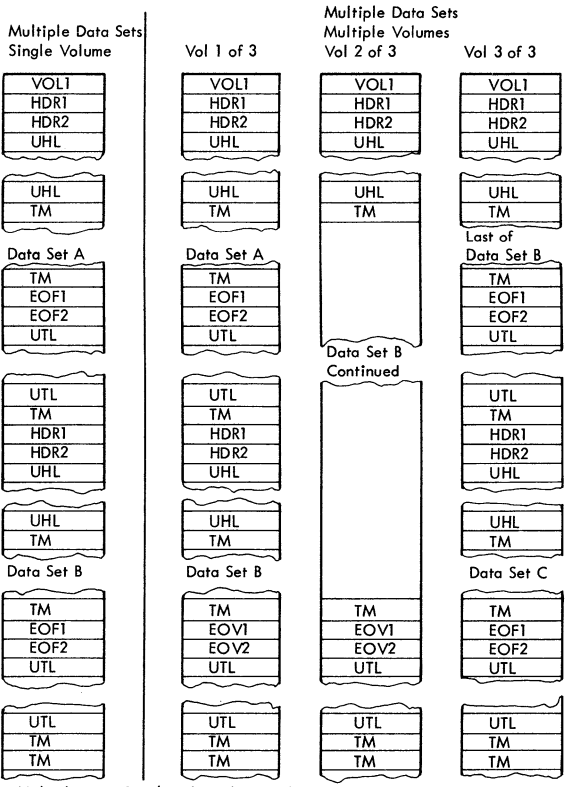
Volume Organization with ANSI Standard Labels



Single Data Set/Single Volume: The volume label is followed by the data set header labels and optional user header labels. The data set is preceded and followed by a tapemark. The data set trailer labels are identified as EOF and followed by optional user trailer labels. Two tapemarks follow the trailer label group to indicate that the data set is the last data set on the volume and is not continued on another volume.

Single Data Set/Multiple Volumes: More than one volume is needed to contain the data set. The last volume is organized the same as a single volume. On the other volumes, the data set trailer labels are identified as EOV instead of EOF, and the trailer label group is followed by two tapemarks. The data set and user labels are repeated on each volume, and there is a separate volume label for each tape.

Volume Organization with ANSI Standard Labels (cont'd)



Multiple Data Sets/Single Volume: The tape begins with a volume label. Each data set is preceded by a header label group and a tapemark, and is followed by a tapemark and a trailer label group. The data set trailer labels are identified as EOF. Each trailer label group is followed by a tapemark; the trailer label group for the last data set on the volume is followed by two tapemarks.

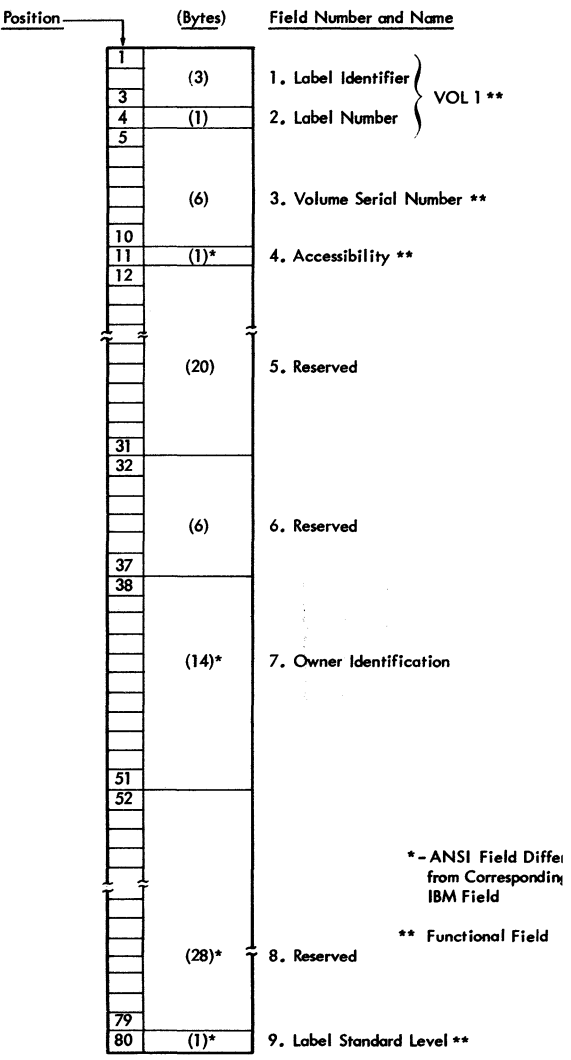
Multiple Data Sets/Multiple Volumes: More than one volume is needed to contain the multiple data set aggregate. The last volume is organized the same as a multiple data set/single volume layout. On the other volumes, the last data set trailer labels are identified as EOV instead of EOF, and the last trailer label group is followed by two tapemarks. There is a separate volume label for each tape.

ANSI Standard Label Processing by Data Management Routines

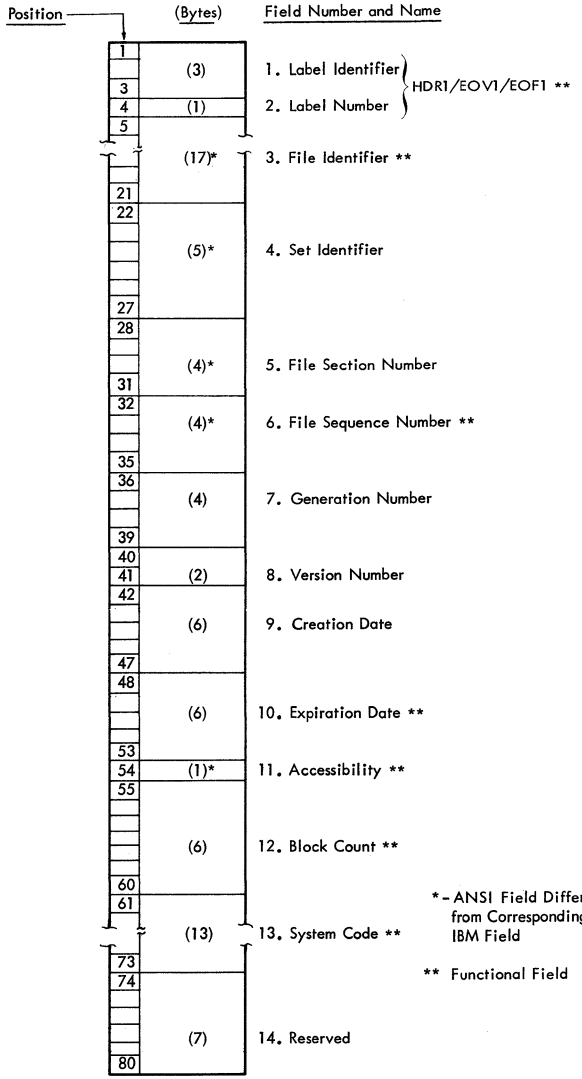
Processing	Volume Label		Header Labels[1]				Trailer Labels[1]			
	VOL1	USER VOLUME LABELS	HDR1	HDR2	HDR3–9	UHL	EOF1 or EOV1	EOF2 or EOV2	EOF3–9 or EOV3–9	UTL
First or Only Volume[2]:										
Checks labels on input tape.	Open	Ignored	Open	Open	Ignored	Open	EOV	bypassed	ignored	EOV
Checks existing lables on output tape before over-writing	Open	Ignored	Open	not read	not read	not read	not read	not read	not read	Open[5]
Writes new labels on output tape.	Open or user[4]	not written	Open	Open	not written	Open	Close or EOV	Close or EOV	not written	Close or EOV
Second or Subsequent Volumes[3]:										
Checks labels on input tape.	EOV	ignored	EOV	bypassed	ignored	EOV	EOV	bypassed	ignored	EOV
Checks existing labels on output tape before over-writing.	EOV	ignored	EOV	not read	not read	not read	not read	not read	not read	not read
Writes new labels on output tape.	EOV or user[4]	not written	EOV	EOV	not written	EOV	Close or EOV	Close or EOV	not written	Close or EOV

Notes:
1. For read backward operations, the action on header and trailer labels is reversed.
2. Includes the first volume of concatenated data sets with unlike characteristics. (Data sets with like characteristics can be processed correctly using the same data control block (DCB), input/output block (IOB), and channel program. Any exception in processing makes the data sets unlike.)
3. Includes the first volume of concatenated data sets with like characteristics.
4. User creates the label with the IEHINITT utility program or a user program.
5. If DISP=MOD is specified on the DD statement, the Open routine positions the tape at the end of the existing data set and allows an input user trailer label routine to process user trailer labels (before overwriting the existing labels).

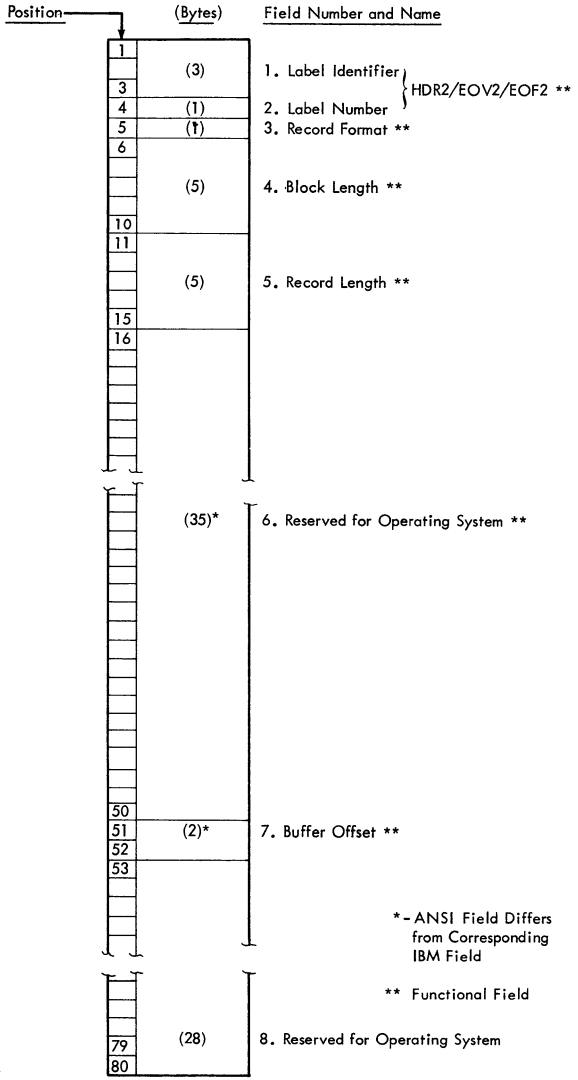
Format of ANSI Standard Volume Label



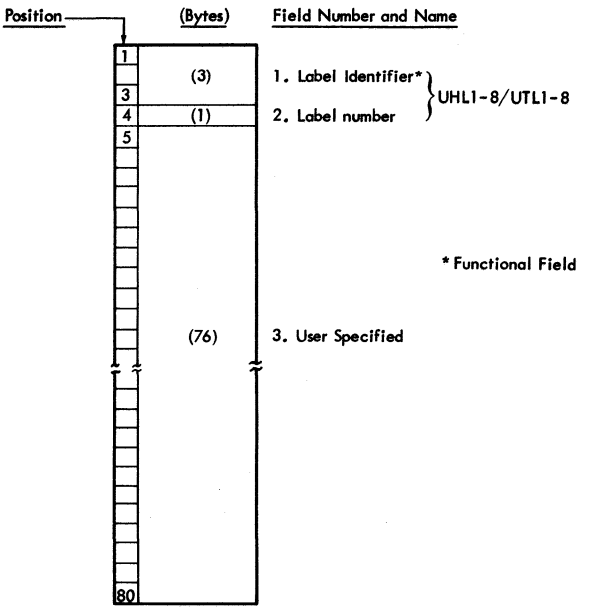
Format of ANSI Header 1 and Trailer 1 Labels



Format of ANSI Header 2 and Trailer 2 Labels



Format of ANSI User Labels



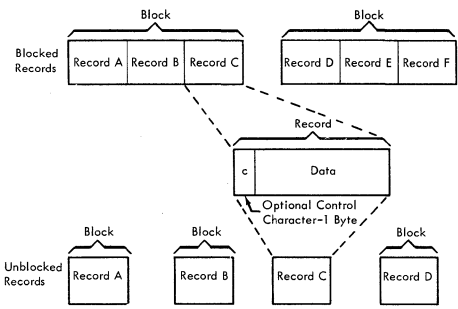
Component Support of Labels Processing Features

Item	Assembler	Linkage Editor	Sort/ Merge	Utilities	COBOL			FORTRAN	PL/1	RPG
					ANS V2	ANS V3	ANS V4			
Uses Data Management Facilities for Label Processing	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard Labels (SL, AL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Standard User Labels (SUL, AUL)	No	No	Yes	Yes	SUL-Yes AUL-No	SUL-Yes AUL-Yes	SUL-Yes AUL-Yes	No	No	No
Supports Nonstandard Labels (NSL)[1]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Unlabeled Tape (NL)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Bypass Label Processing Option (BLP)[2]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Supports Concatenated Data Sets with Unlike Attributes	No	Yes	No	No	No	No	No	No	No	No

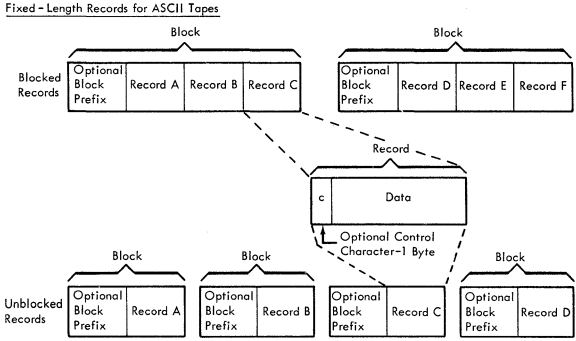
[1] NSL can be specified only when installation-written routines that write and process the nonstandard labels have been incorporated into the operating system.
[2] If the BLP option is not specified at system generation, its use defaults to NL.

Data Set Record Formats

Fixed - Length Records

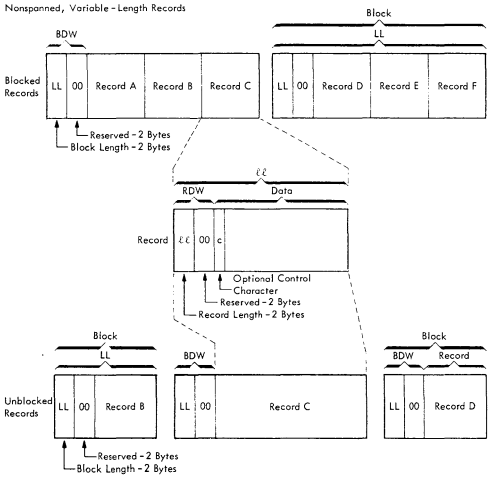


Fixed - Length Records for ASCII Tapes

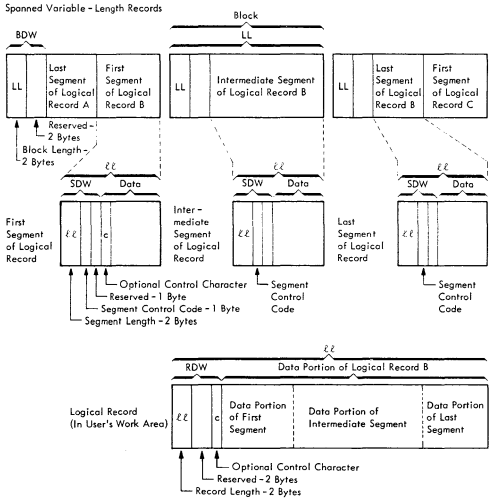


Data Set Record Formats (cont'd)

Nonspanned, Variable - Length Records



Spanned Variable - Length Records



Legend

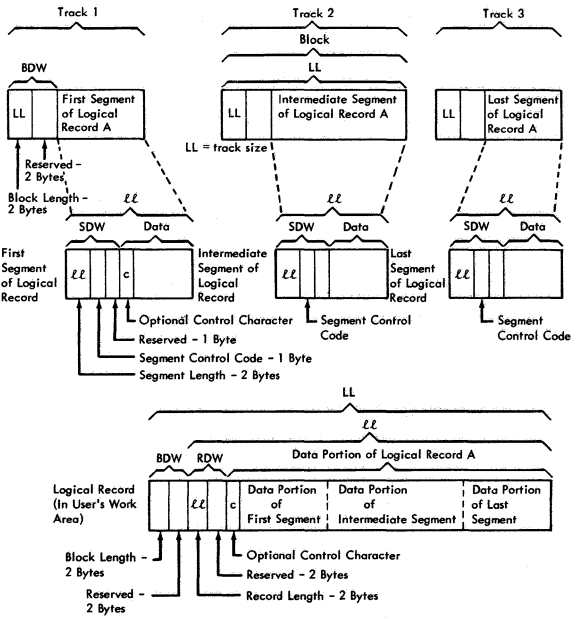
BDW = block descriptor word
RDW = record descriptor word
SDW = segment descriptor word
LL = block length
ℓℓ = segment length

Segment Control Codes

Binary Code	Relative Position of Segment
00	Complete logical record
01	First segment of a multisegment record
10	Last segment of a multisegment
11	Segment of a multisegment record other than the first or last segment

Data Set Record Formats (cont'd)

Spanned Variable - Length Records for BDAM Data Sets



Note: Not All Segment and Block Combinations are Represented

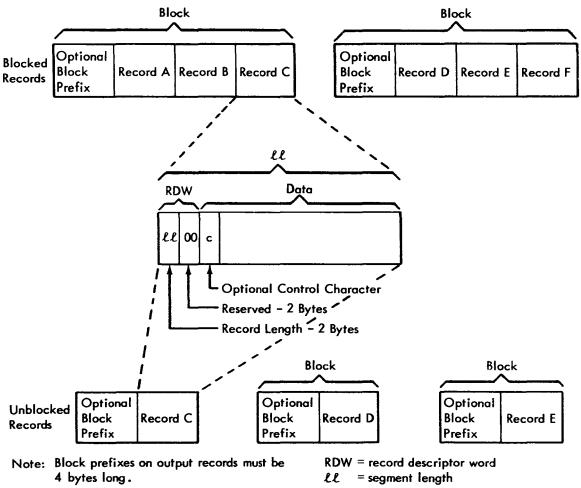
Legend

BDW = block descriptor word
RDW = record descriptor word
SDW = segment descriptor word
LL = block length
ℓℓ = segment length

Segment Control Codes

Binary Code	Relative Position of Segment
00	Complete logical record
01	First segment of a multisegment record
10	Last segment of a multisegment record
11	Segment of a multisegment record other than the first or last segment

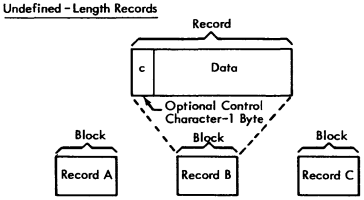
Variable - Length Records for ASCII Tapes



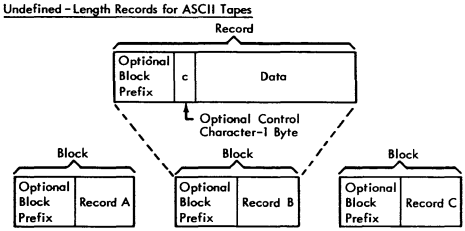
Note: Block prefixes on output records must be 4 bytes long.

RDW = record descriptor word
ℓℓ = segment length

Undefined - Length Records



Undefined - Length Records for ASCII Tapes



The BLDVRP, DLVRP, GETIX, MRKBFR, PUTIX, SCHBFR, SHOWCAT, and WRTBFR macros are described in *OV/VS Virtual Storage Access Method (VSAM) Options for Advanced Applications.*

ACB (Generate an Access-Method Control Block)

[*label*]	ACB	[AM=VSAM] [,BSTRNO=*number*] [,BUFND=*number*] [,BUFNI=*number*] [,BUFSP=*number*] [,CATALOG={YES | NO}] [,CRA={SCRA | UCRA}] [,DDNAME=*ddname*] [,EXLST=*address*] [,MACRF=([ADR] [,CNV] [,KEY] [,{CFX | NFX}] [,{DDN | DSN}] [,{DFR | NDF}] [,DIR] [,SEQ] [,SKP] [,{ICI | NCI}] [,IN] [,OUT] [,{NIS | SIS}] [,{NRM | AIX}] [,{NRS | RST}] [,{NSR | LSR}] [,{NUB | UBF}]) [,MAREA=*address*] [,MLEN=*number*] [,PASSWD=*address*] [,STRNO=*number*]

CHECK (Suspend Processing)

[*label*]	CHECK	RPL=*address*

CLOSE (Disconnect Program and Data)

[*label*]	CLOSE	(*address*,...) [,TYPE=T]

ENDREQ (Terminate a Request)

[*label*]	ENDREQ	RPL=*address*

ERASE (Delete a Record)

[*label*]	ERASE	RPL=*address*

EXLST (Generate an Exit List)

[*label*]	EXLST	[AM=VSAM] [,EODAD=(*address*[,{A | N}][,L])] [,JRNAD=(*address*[,{A | N}][,L])] [,LERAD=(*address*[,{A | N}][,L])] [,SYNAD=(*address*[,{A | N}][,L])]

GENCB (Generate an Access-Method Control Block)

[*label*]	GENCB	BLK=ACB [,AM=VSAM] [,BSTRNO=*number*] [,BUFND=*number*] [,BUFNI=*number*] [,BUFSP=*number*] [,CATALOG={YES | NO}] [,COPIES=*number*] [,CRA={SCRA | UCRA}] [,DDNAME=*ddname*] [,EXLST=*address*] [,LENGTH=*number*] [,MACRF=([ADR] [,CNV] [,KEY] [,{CFX | NFX}] [,{DDN | DSN}] [,{DFR | NDF}] [,DIR] [,SEQ] [,SKP] [,{ICI | NCI}] [,IN] [,OUT] [,{NIS | SIS}] [,{NRM | AIX}] [,{NRS | RST}] [,{NSR | LSR}] [,{NUB | UBF}]) [,MAREA=*address*] [,MLEN=*number*] [,PASSWD=*address*] [,STRNO=*number*] [,WAREA=*address*]

GENCB (Generate an Exit List)

[*label*]	GENCB	BLK=EXLST [,AM=VSAM] [,COPIES=*number*] [,EODAD=(*address* [,{A | N}] [,L])] [,JRNAD=(*address* [,{A | N}] [,L])] [,LENGTH=*number*] [,LERAD=(*address* [,{A | N}] [,L])] [,SYNAD=(*address* [,{A | N}] [,L])] [,WAREA=*address*]

GENCB (Generate a Request Parameter List)

[*label*]	GENCB	BLK=RPL [,ACB=*address*] [,AM=VSAM] [,AREA=*address*] [,AREALEN=*number*] [,ARG=*address*] [,COPIES=*number*] [,ECB=*address*] [,KEYLEN=*number*] [,LENGTH=*number*] [,MSGAREA=*address*] [,MSGLEN=*number*] [,NXTRPL=*address*] [,OPTCD=([{ADR | CNV | KEY}] [,{ARD | LRD}] [,{ASY | SYN}] [,{DIR | SEQ | SKP}] [,{FKS | GEN}] [,{FWD | BWD}] [,{KEQ | KGE}] [,{LOC | MVE}] [,{NSP | NUP | UPD}])] [,RECLEN=*number*] [,TRANSID=*number*] [,WAREA=*address*]

GET (Retrieve a Record)

```
[label ]  GET       RPL=address
```

MODCB (Modify an Access-Method Control Block)

```
[ label ]  MODCB    ACB= address
                    [,BSTRNO=number ]
                    [,BUFND=number]
                    [,BUFNI=number]
                    [,BUFSP=number]
                    [,CATALOG={YES | NO}]
                    [,CRA={SCRA | UCRA}]
                    [,DDNAME= ddname ]
                    [,EXLST=address]
                    /MACRF=([ADR] [,CNV] [,KEY] \
                            [,{CFX | NFX}]
                            [,{DDN | DSN}]
                            [,{DFR | NDF}]
                            [,DIR] [,SEQ] [,SKP]
                            [,{ICI | NCI}]
                            [,IN] [,OUT]
                            [,{NIS | SIS}]
                            [,{ NRM | AIX}]
                            [,{NRS | RST}]
                            [,{NSR | LSR}]
                            [,{NUB | UBF}]) /
                    [,MAREA=address ]
                    [,MLEN=number ]
                    [,PASSWD= address ]
                    [,STRNO= number ]
```

MODCB (Modify an Exit List)

```
[label ]  MODCB     EXLST=address
                    [,AM=VSAM]
                    [,EODAD=( address [,{A | N}] [,L])]
                    [,JRNAD=( address [,{A | N}] [,L])]
                    [,LERAD=( address [,{A | N}] [,L])]
                    [,SYNAD=( address [,{A | N}] [,L])]
```

MODCB (Modify a Request Parameter List)

```
[label ]  MODCB     RPL=address
                    [,ACB= address ]
                    [,AREA= address ]
                    [,AREALEN= number ]
                    [,ARG=address]
                    [,ECB= address ]
                    [,KEYLEN=number]
                    [,MSGAREA=address ]
                    [,MSGLEN= number ]
                    [,NXTRPL=address ]
                    [,OPTCD=([{ADR | CNV | KEY}]
                             [,{ARD | LRD}]
                             [,{ASY | SYN}]
                             [,{DIR | SEQ | SKP}]
                             [,{FKS | GEN}]
                             [,{FWD | BWD}]
                             [,{KEQ | KGE}]
                             [,{LOC | MVT}]
                             [,{NSP | NUP | UPD}])]
                    [,RECLEN=number ]
                    [,TRANSID= number ]
```

OPEN (Connect Program and Data)

[*label*]	**OPEN**	(*address*],[*options*)] ...)

POINT (Position for Access)

[[*label*]	**POINT**	**RPL=***address*

PUT (Store a Record)

[*label*]	**PUT**	**RPL=** *address*

RPL (Generate a Request Parameter List)

[*label*]	**RPL**	**ACB=***address* **[,AM=VSAM]** **[,AREA=***address* **]** **[,AREALEN=***number* **]** **[,ARG=** *address* **]** **[,ECB=***address* **]** **[,KEYLEN=** *number* **]** **[,MSGAREA=** *address* **]** **[,MSGLEN=***number***]** **[,NXTRPL=** *address* **]** **[,OPTCD=([{ADR | CNV | KEY}]** **[,{ARD | LRD}]** **[,{ASY | SYN}]** **[,{DIR | SEQ | SKP}]** **[,{FKS | GEN}]** **[,{FWD | BWD}]** **[,{KEQ | KGE}]** **[,{LOC | MVE}]** **[,{ NSP | NUP | UPD}])]** **[,RECLEN=** *number* **]** **[,TRANSID=***number***]**

SHOWCB (Display Fields of an Access-Method Control Block)

[*label*]	**SHOWCB**	**ACB=** *address* **,AREA=***address* **,LENGTH=***number* **[,OBJECT={DATA | INDEX}]** **,FIELDS=([ACBLEN] [,AVSPAC] [,BFRFND]** **[,BSTRNO] [,BUFND] [,BUFNI]** **[,BUFNO] [,BUFRDS] [,BUFSP]** **[,CINV] [,DDNAME] [,ENDRBA]** **[,ERROR] [,EXLST] [,FS]** **[,KEYLEN] [,LRECL] [,MAREA]** **[,MLEN] [,NCIS] [,NDELR]** **[,NEXCP] [,NEXT] [,NINSR]** **[,NIXL] [,NLOGR] [,NRETR]** **[,NSSS] [,NUIW] [,NUPDR]** **[,PASSWD] [,RKP] [,STMST]** **[,STRMAX] [,STRNO] [,UIW])**

SHOWCB (Display Fields of an Exit List)

[*label*]	**SHOWCB**	**AREA=** *address* **,EXLST=** *address* **,FIELDS=([EODAD] [,EXLLEN] [,JRNAD]** **[,LERAD] [,SYNAD])** **,LENGTH=***number*

SHOWCB (Display Fields of a Request Parameter List)

[*label*]	SHOWCB	AREA=*address* ,FIELDS=([ACB][,AIXPC][,AREA][,AREALEN] [,ARG][,ECB][,FDBK][,FTNCD][,KEYLEN] [,MSGAREA][,MSGLEN] [,NXTRPL][,RBA][,RECLEN] [,RPLLEN][,TRANSID]) ,LENGTH=*number* ,RPL=*address*

TESTCB (Test a Field of an Access-Method Control Block)

[*label*]	TESTCB	ACB=*address* [,ERET=*address*] [,OBJECT={DATA | INDEX}] , ATRB=([FSDS][,KSDS][,REPL] [,RRDS][,SPAN][,SSWD] [,UNQ][,WCK])| CATALOG= YES NO MACRF=([ADR][,AIX][,CFX][,CNV] [,DDN][,DFR][,DIR][,DSN] [,GSR][,ICI][,IN][,KEY] [,LSR][,NCI][,NDF][,NFX] [,NIS][,NRM][,NSR][,NUB] [,OUT][,RST][,SEQ][,SIS] [,SKP][,UFB])| OFLAGS=OPEN OPENOBJ={PATH | BASE | AIX}| ACBLEN=*number*| AVSPAC=*number*| BSTRNO=*number*| BUFND=*number*| BUFNI=*number*| BUFNO=*number*| BUFSP=*number*| CINV=*number*| DDNAME=*ddname*| ENDRBA=*number*| ERROR=*number*| EXLST=*address*| FS=*number*| KEYLEN=*number*| LRECL=*number*| MAREA=*address*| MLEN=*number*| NCIS=*number*| NDELR=*number*| NEXCP=*number*| NEXT=*number*| NINSR=*number*| NIXL=*number*| NLOGR=*number*| NRETR=*number*| NSSS=*number*| NUPDR=*number*| PASSWD=*address*| RKP=*number*| STMST=*address*| STRNO=*number*|

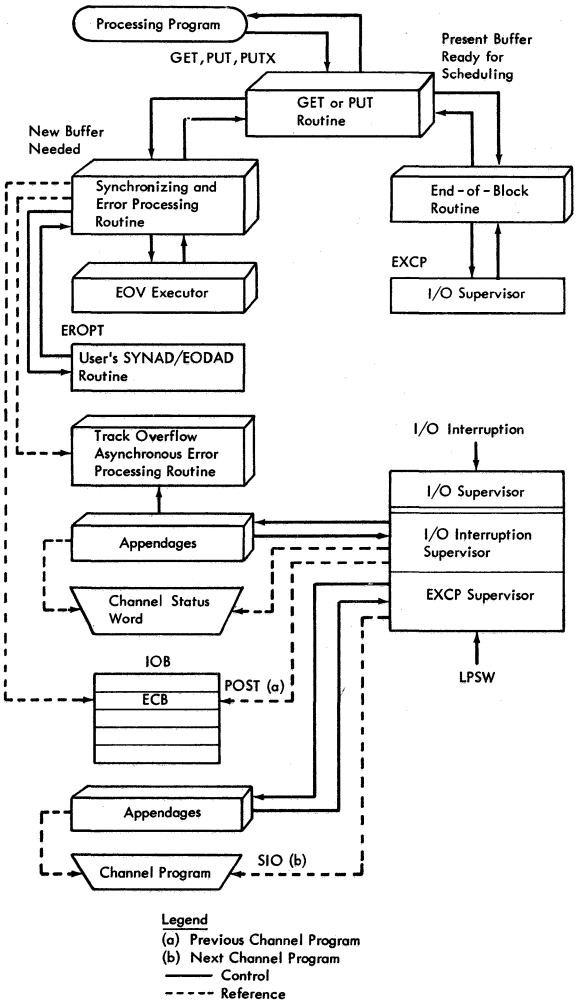
TESTCB (Test a Field of an Exit List)

[*label*]	TESTCB	EXLST=*address* [,ERET= *address*] {EODAD={0|([*address*] [,{A|N}] [,L])}| JRNAD={0|([*address*] [,{A|N}] [,L])}| LERAD={0|([*address*] [,{A|N}] [,L])}| SYNAD={0|([*address*] [,{A|N}] [,L])}} [,EXLLEN= *number*]

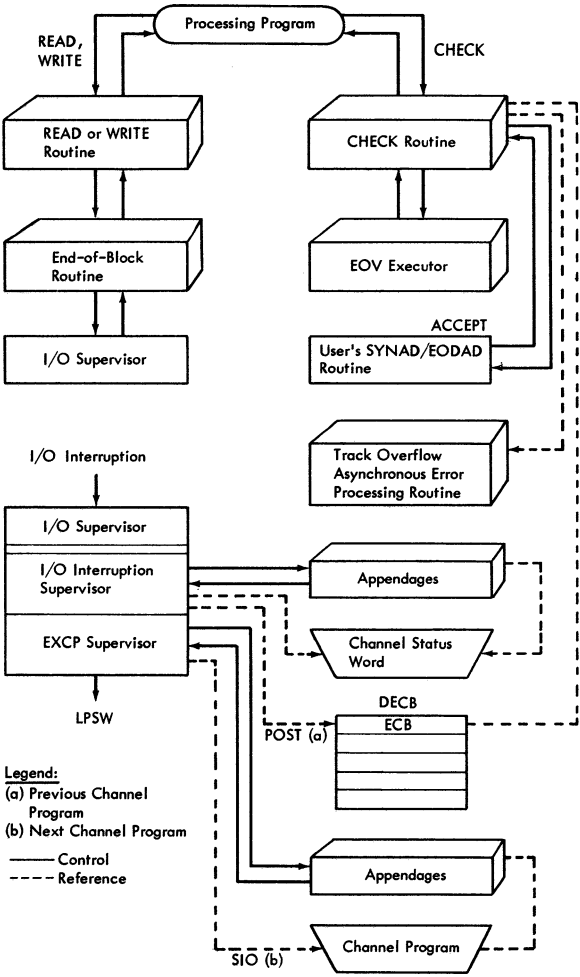
TESTCB (Test a Field of A Request Parameter List)

[*label*]	TESTCB	RPL=*address* [,ERET=*address*] { IO=COMPLETE| OPTCD=([ADR] [,ARD] [,ASY] [,BWD] [,CNV] [,DIR] [,FKS] [,FWD] [,GEN] [,KEQ] [,KEY] [,KGE] [,LOC] [,LRD] [,MVE] [,NSP] [,NUP] [,SEQ] [,SKP] [,SYN] [,UPD]) RBA= *number* | RECLEN= *number* | RPLLEN= *number* | ACB=*address* | AIXFLAG=AIXPKP | AIXPC=*number* | AREA= *address* | AREALEN=*number* | ARG=*address* | ECB=*address* | FDBK= *number* | FTNCD=*number* | KEYLEN=*number* | MSGAREA=*address* | MSGLEN=*number* | NXTRPL= *address* | TRANSID= *number*}

Flow of Control in QSAM



Flow of Control in BSAM and BPAM



SAM Open Executor Selector-Stage 1

These diagrams show the access method conditions that cause different executors to be selected, loaded, and to receive control after loading. X represents a condition that must be satisfied for the executor marked in that column. No indicates that the condition must not be specified for the executor to be selected. A blank in the upper portion of the table indicates that either the condition is not required for selection or not examined at the time

Access Method Options	Selections															
Actual Data Set	X	X	X	X	X			X	X	X	X	X	X		X	X
Dummy Data Set							X									
*, DATA, or SYSOUT specified or DD statement													X			
3505 (OMR/RCE) or 3525												X				
3886 (OCR)														X		
3800 Printer															X	X
Direct Access Device		X		X					X		X					
Printer with UCS Feature (1403 or 3211)					X											
Printer with forms control buffer (3211 or 2245)						X										
Buffer Pool Required			X	X						X	X	X				X
User totaling Specified								X	X	X	X					
Executors																
IGG0191A	1A	1A	1A	1A	1A	1A		1A	1A	1A	1A	1A	1A	1A	1A	1A
IGG0191B	1B	1B	1B	1B	1B	1B		1B	1B	1B	1B	1B	1B	1B	1B	1B
IGG0191C							1C									
IGG0191I			1I	1I						1I	1I	1I		1I		1I
IGG0191N		1N		1N					1N	1N	1N	1N	1N			
IGG0191T					1T	1T										
IGG0191U					1U											
IGG0191V					1V											
IGG0191Y								1Y	1Y	1Y	1Y					
IGG0193I			3I	3I						3I	3I	3I				3I
IGG0196A	6A	6A	6A	6A	6A	6A		6A	6A	6A	6A	6A	6A		6A	6A
IGG0196B	6B	6B	6B	6B	6B	6B		6B	6B	6B	6B	6B	6B		6B	6B
IGG196Q															6Q	6Q
IGG0196R														6R		
IGG01961	6I	6I	6I	6I	6I	6I		6I	6I	6I	6I	6I	6I		6I	6I
IGG0197E						7E										
IGG0197F						7F										
IGG0197L												7L	7L			
IGG0197M												7M	7M			
IGG0197U					7U											
IGG0199F													9F			
IGG0199G													9G			
IGG0199W													9W			

SAM Open Executor Selector - Stage 2

Access Method Options[1]	Selections																								
BSAM or	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
QSAM	X	X	X	X	X	X		X		X		X	X	X		X		X	X	X	X	X	X	X	
Input or		X	X	X						X									X	X	X	X			
Output	X				X											X				X		X	X		
Inout, Outin							X		X						X		X								
Update						No	No	No					X					X						No	
Unit Record or						X								X					X	X	X	X		X	
Magnetic Tape or				X	X	X	X							X	X										
Paper Tape						X																			
Direct-Access Storage	X	X	X	X	X						X	X			X			X	X	X					X
Write-Load (Create BDAM)											X	X													
Simple Buffering	X	X	X			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
Exchange Buffering				X	X																				
Track Overflow	No	No	No	No	No	No	No	X				X	X					X					X	No	
Chained Scheduling	No	No	No			No	No				X				X	X	X	X							No
Search Direct		X																					X		
RPS Device													No					X							
3505																			X		X				
3525																				X		X			
OMR or																					X				
RCE or																					X	X			
Print only and Associated Files																							X		
3890																									X
Executors																									
IGG0191D	1D	1D	1D																						
IGG0191E				1E																					
IGG0191F					1F																				
IGG0191G						1G	1G												1G	1G				1G	
IGG0191H								1H																	
IGG0191J									1J																
IGG0191K										1K															
IGG0191L											1L	1L													
IGG0191M												1M													
IGG0191O			1O																						
IGG0191P													1P					1P							
IGG0191Q														1Q											
IGG0191R															1R										
IGG0191S								1S															1S		
IGG0191W																1W									
IGG0191X																	1X								
IGG0191Z																		1Z							
IGG019123																		23							
IGG0196J				6J																					
IGG0196K																								6K	
IGG0196L									6L																
IGG0196P													6P												
IGG0197N																			7N	7N	7N	7N			
IGG0197P																					7P	7P			
IGG0197Q																					7Q	7Q			
IGG0197V																									7V
IGG0199K																							9K		
IGG0199L											9L														
IGG0199O		9O																							

[1] If *, DATA, or SYSOUT are specified on the DD statement, no stage 2 executors are loaded.

SAM Open Executor Selector-Stage 3

Access Method Options	Selection				
Paper Tape	X				
Update		X			
Chained Scheduling			X		
Exchange Buffering				X	
Track Overflow					X
None of the preceding					X
Input				X	
*, DATA, or SYSOUT specified on DD statement					X X
QSAM				X	
Variable-length Record Format					X X
Spanned Records					X
Executors					
IGG01910			10		
IGG01911				11	
IGG01912	12 12				
IGG01913		13	13		
IGG01914			14		
IGG01915				15	15
IGG01916					16	
IGG01917				17		
IGG01918	18 18				
IGG01919		19	19		19
IGG01926		26	26		26
IGG0198L					8L 8L
IGG01990			90		
IGG01991				91	
IGG01992					92	
IGG01993				93	
IGG01994					94

This section contains a variety of information about Job Control Language statements and certain terminal-based systems.

Source Publications

Additional JCL reference information is contained in *OS/VS1 JCL Reference*, GC24-5099.

Additional operator commands information can be found in *Operator's Library: OS/VS1 Reference*, GC38-0110.

Additional SMF information is contained in *OS/VS1 System Management Facilities (SMF)*, GC24-5115.

Additional RES information is contained in *OS/VS1 RES Workstation User's Guide*, GC28-6879.

Additional CRJE information is contained in *OS/MFT, OS/MVT, and OS/VS1 CRJE System Programmer's Guide*, GC30-2016; *OS/MFT, OS/MVT, and OS/VS1 CRJE Terminal User's Guide*, GC30-2014; and *Operator's Library: OS/VS1 CRJE*, GC38-0335.

Information on Mass Storage System (MSS) operator commands is contained in *OS/VS Operator's Library: IBM 3850 Mass Storage System (MSS) Operations Guide*, GC35-0014.

Comment - Delimiter - Null - PEND - PROC Statements

//	Operations
//*	Comments coded in free form. If all comments cannot be included on one statement, they can be continued on consecutive comment statements.

The Delimiter Statement

/*	Operations
/* or any two characters defined by the DLM parameter.	Comments coded in free form. If all comments cannot be included on one statement, they can be continued on consecutive comment statements.

The Null Statement

//	Operations
//	Blanks. The null statement placed at the end of job control statements and data indicates that the job is to be put on the queue of jobs ready for processing.

The PEND Statement

//Name	Operation	Comment Field
//name (up to 8 characters followed by one or more blanks)	PEND	Comments coded in free form. If all comments cannot be included on one statement, they can be continued on consecutive comment statements.

The PROC Statement

//Name	Operation	Operands
//name (up to 8 characters followed by one or more blanks)	PROC	Symbolic parameters and their corresponding default values, separated by commas: symbolic parm = val, symbolic parm = val In a catalogued procedure, the operand field is not optional. In an in-stream procedure, the operand field is optional.

JOB-EXEC-DD Statements

The JOB Statement

//Name	Operation	Operands
//jobname	JOB	[Positional Parameters] (Can be made mandatory) [,list of keyword parameters]

The EXEC Statement

//Name	Operation	Operands
// stepname	EXEC	{PGM= {[PROC=]procedurename} [,list of keyword parameters]

The DD Statement

//Name	Operation	Operands
// ddname procstepname. ddname	DD	[DUMMY] [DATA] [*] [,list of keyword parameters]

JOB Statement

Positional Parameters	Comments
([account number][,additional accounting information,...])[programmer's name]	Can be made mandatory

Keyword Parameters	Comments
[ADDRSPC={VIRT / REAL}]	Requests storage type
[CLASS=jobclass]	Assigns A-Z, 0-9
[COND=((code,operator),...)]	Specifies a maximum of 8 tests
[MPROFILE='profile string']	For ISSP only
[MSGCLASS=output class]	Assigns A-Z, 0-9
[MSGLEVEL=({0 / 1 / 2}[,0 / ,1])]	
[PROFILE='profile string']	For ISSP only
[PRTY=priority]	Assigns 0-13
[RD={R / RNC / NC / NR}]	Restart definition
[REGION=value K]	Specifies amount of storage space
[RESTART=({* / stepname / stepname.procstepname}[,checkid])]	For deferred restart
[TIME={([minutes][,seconds]) / 1440}]	Assigns job CPU time limit
[TYPRUN={HOLD / SCAN}]	Holds a job in job queue, or scans JCL for syntax errors

Legend:

{ } Mandatory - choose one.

[ ] Optional - choose one or none.

EXEC Statement

Positional Parameters	Comments
{PGM = {program name / *.stepname.ddname / *.stepname.procstepname.ddname}} [PROC =]procedure name	Identifies program or cataloged procedure

Keyword Parameters	Comments
[ACCT =(accounting information,...)] [ACCT.procstepname =(accounting information,...)]	Accounting information for step
[ADDRSPC = {VIRT / REAL}]	Requests storage type
[COND =({(code,operator) / (code,operator,stepname) / (code,operator,stepname.procstepname)}[,...][,][EVEN / ONLY])	Specifies a maximum of 8 tests, or 7 tests if EVEN or ONLY is coded
COND.procstepname =({(code,operator) / (code,operator,stepname) / (code,operator,stepname.procstepname)}[,...] [,][EVEN / ONLY])	Specifies a maximum of tests, or 7 tests if EVEN or ONLY is coded
[PARM=value] [PARM.procstepname=value]	Parentheses or apostrophes enclosing value may be required
[RD = {R / RNC / NC / NR}] [RD.procstepname = {R / RNC / NC / NR}]	Restart definition
[REGION = valueK]	Specifies amount of storage space
[TIME = {([minutes][,seconds]) / 1440}] [TIME.procstepname = {(minutes,seconds) / 1440}]	Assigns step CPU time limit

Legend:
- {} Mandatory - choose one.
- [] Optional - choose one or none.

DD Statement

Positional Parameters	Comments
[* / DATA]	Defines data set in the input stream or on diskette as associated data.
[DUMMY]	Bypass I/O operations on a data set (BSAM and QSAM).

Keyword Parameters	Comments
[AFF = ddname]	Requests channel separation
[AMP = [,'AMORG'] [,'BUFND =number'] [,'BUFNI =number'] [,'BUFSP =number'] [,'CROPS = { NCK' / NRC' / NRE' / RCK' }] [,'OPTCD= { I' / L' / IL' }] [,'RECFM = { F' / FB' / V' / VB' }] [,'STRNO =number'] [,'SYNAD =modulename'] [,'TRACE']]	Modifies the program processing VSAM clusters or components
[CHKPT = EOV]	Specifies checkpoint to be taken for the data set at end-of volume. For more information, see OS/VS Checkpoint Restart.
CHARS = (id_1[,id_2,...id_4])	Specifies 5th through 8th characters of names of load modules containing translate tables and table headers. ID_1 of DUMP specifies high-density dump.
COPIES = (n_0[,(n_1,n_2,...n_8)])	Specifies number and grouping of copies to be printed.
[DCB = (list of attributes) / DCB = ({ dsname / *.ddname / *.stepname.ddname / *.stepname.procstepname.ddname } [,list of attributes])]	Completes the data control block
[DDNAME =ddname]	Postpones the definition of a data set
[DEST =userid]	Specifies remote destination for SYSOUT data set
[DISP=({ NEW / OLD / SHR / MOD } [,DELETE / ,KEEP / ,PASS / ,CATLG / ,UNCATLG / ,] [,DELETE / ,KEEP / ,CATLG / ,UNCATLG])]	Assigns a status, disposition, and conditional disposition to the data set. CATLG, NEW, and UNCATLG are valid for VSAM components and clusters

Legend:

{ } Mandatory - choose one.

[ ] Enclosing subparameter, indicates that subparameter is optional; if more than one line is enclosed, choose one or more.

[ ] Enclosing entire parameter, indicates that parameter may be optional, depending on what type of data set you are defining.

DD Statement (cont'd)

Keyword Parameters	Comments
[DLM=delimiter]	Assigns delimiter other than /*
[DSID = (id[,V])]	Specifies identifier of associated data input or SYSOUT output, on diskette I/O device.
[{DSNAME} = { dsname dsname(member name) dsname(generation number) dsname(area name) &&dsname &&dsname(member name) &&dsname(area name) *.ddname *.stepname .ddname *.stepname .procstepname .ddname }]	Assigns a name to a new data set or to identify an existing data set. An unqualified name is 1-8 characters beginning with an alphabetic or national character. Area, generation, member, and temporary names are invalid for VSAM clusters and components
[FCB=(image-id[,ALIGN][,VERIFY])]	Specifies forms control information. The FCB parameter is ignored if the data set is not written to a 3211 or 3800 printer. An image-id of STD3 specifies a dump with 8 lines/inch and 80 lines/page.
FLASH=(id[,count])	Specifies forms overlay and number of copies. [Max count is COPIES= n_0 value]
[HOLD = {YES / NO}]	Specifies whether JES writer processing of a SYSOUT data set is to be deferred or processed normally
[LABEL =([data set seq#] [,SL / ,SUL / ,AL / ,AUL / ,NSL / ,NL / ,BLP / ,LTM / ,] [,PASSWORD / ,NOPWREAD] [,IN / ,OUT] [,EXPDT =yyddd / ,RETPD =nnnn])]	Supplies label information
MODIFY =(id[,trc])	Specifies copy modification pattern and translate table reference character. [See CHARS = parameter]
[MSVGP = group - id]	Specifies the ID of a group of Mass Storage Volumes. For more information, see OS/VS Mass Storage System (MSS) Utilities for MSS Space Management.

Legend:

{ } Mandatory - choose one.

[ ] Enclosing subparameter, indicates that subparameter is optional; if more than one line is enclosed, choose one or more.

[ ] Enclosing entire parameter, indicates that parameter may be optional, depending on what type of data set you are defining.

DD Statement (cont'd)

Keyword Parameters	Comments
[OUTLIM =number]	Limits the number of logical records you want included in the output data set (OUTLIM = is valid only on DD statement with SYSOUT = present)
[QNAME =process name]	Specifies the name of a TPROCESS macro that defines a destination queue for messages received by means of TCAM
[SEP =(ddname, ...)]	Requests channel separation
[{ SPACE =({ TRK / CYL / blocklength }, (primary [,secondary] [,directory] [,index]) }] [{ [,RLSE] [, CONTIG / , MXIG / , ALX / ,] [,ROUND] }] [SPACE =(ABSTR, (primary quantity, address [,directory] [,index]))]	Assigns space on a direct access volume for a new data set Assigns specific tracks on a direct access volume for a new data set
[[1]SPLIT = ({ (n, CYL, (primary quantity [, secondary quantity])) / n / (percent, blocklength, (primary [, secondary])) / percent })]	Assigns space on a direct access volume for a new data set. Data sets share cylinders
[[1]SUBALLOC =({ TRK / CYL / blocklength }, (primary [,secondary] [,directory]) { ,ddname / ,stepname.ddname / ,stepname.procstepname.ddname })]	Requests part of the space on a direct access volume assigned earlier in the job
[SYSOUT = ({ classname / (classname [,program name] [,form number]) / PROFILE ='sysout profile string' / ([,program name] [,form number] ,PROFILE ='sysout profile string') })]	Routes a data set through the output stream. For classname, assign A - Z or 0 - 9
[TERM =RT]	Indicates that an RTAM device is in use
[COMPACT = compaction table name]	
[UCS =(character set code [,FOLD] [,VERIFY])]	Requests a special character set for a 1403 printer
[{ UNIT = ([unit address] / device type / group name [,unit count / ,P] [,DEFER] [,SEP =(ddname, ...)]) / UNIT =AFF =ddname }]	Provides the system with unit information
[{ VOLUME } =([PRIVATE] [,RETAIN] [,volume seq#] / VOL } [, volume count] [,] [SER =(serial number, ...) / REF =dsname / REF =*.ddname / REF =*.stepname.ddname / REF =*.stepname.procstepname.ddname])]	Provides the system with volume information. REF =dsname, *.step.ddname, and *.stepname.procstepname are invalid for VSAM components and clusters

Legend:

{ } Mandatory - choose one.

[ ] Enclosing subparameter, indicates that subparameter is optional; if more than one line is enclosed, choose one or more.

[ ] Enclosing entire parameter, indicates that parameter may be optional, depending on what type of data set you are defining.

[1] Invalid for VSAM components and clusters.

Operator Command Outlines

This section contains outlines of OS/VS1 system operator commands and RES central operator commands. For details of usage and appropriate operands, see Operator's Library: OS/VS1 Reference, GC38-0110.

Operator commands that require no modification for RES. These commands are not valid from RES workstation.

CONTROL	SET
DEFINE	SWAP
DUMP	SWITCH
HALT	UNLOAD
LOG	VARY
MODE	WRITELOG
PAGETUNE	

Operator commands that use additional operands for RES.

CANCEL	REPLY
DISPLAY	RESET
HOLD	START
MODIFY	STARTF
MONITOR	STOP
RELEASE	STOPMN
	WRITER

Operator commands for RES.

LISTBC	ROUTE
LOGON	SEND
LOGOFF	

Operator Command Outlines (cont'd)

Operation	Operand
{CANCEL / C}	{[JBN=] jobname* [,DUMP] [,ALL] [,IN[=i / =HOLD]] [,OUT[=s / =HOLD]] [,USER=userid] / [DEV=] unitaddr* [procname] identifier*} * May be specified up to five times if separated by commas and enclosed in parentheses. Can be combined with the other parameters that are allowed to be specified up to five times.
{DEFINE / N}	[LIST / PARM=membername]
{DISPLAY / D}	{T / A / U [,TP / ,GRAPHIC / ,TAPE / ,DASD / ,UR] [,ONLINE / ,OFFLINE] [,cuu][,nnn] / R {,USER = userid / ,ALL} [,LIST] / RT, {ALL / ACT / INACT / TERM = {termid / nnn.aam[,nnn.aam]...}} [,L|,LB|,LS] / {N / Q} {[=qclass][,USER=userid] / [,ALLQ]} jobname *[,HOLD] [,ALLQ / ,USER=userid] / CONSOLES / P, {IN = class / IN = 'string[,string ...]'[,ALL] / OUT = class / OUT = 'string[,string ...]'[,ALL]} / SQA / USER [,L / , = userid]} * May be specified up to five times if separated by commas and enclosed in parentheses.
DUMP	[test]
{HALT / Z}	EOD
{HOLD / H}	{ALL / IN[=inclass] / Q [=inclass] / OUT [=outclass] [,JBN] / jobname* [,OUT[=outclass[outclass...]]] [,USER = userid]} * May be specified up to five times if separated by commas and enclosed in parentheses.
{LISTBC / LB}	[NOTICES] ,MAIL [=userid] [MAIL][=userid][,NOTICES]
{LOG / L}	'text'
LOGOFF	userid [,slow]

Operator Command Outlines (cont'd)

Operation	Operand
LOGON	userid[/password][TERM (termid)] [PROC (procname)] [NOTICES / NONOTICES] [MAIL / NOMAIL / UNATT]
MODE	{ STATUS RETRY,{RECORD / QUIET} MAIN,{RECORD / QUIET} CONTROL,{THRESHOLD / QUIET} } Note: Blanks may be used in place of the commas in this command.
{MODIFY / F}	{ [procname.] / [name.] / unitaddr / procname.Pnn,'data' } identifier { ,TYPRUN={HOLD / NOHOLD} ,CLASS=outclass ,CLASS=jobclass ,START={ALL / (n,...)} ,STOP={ALL / B / S / (n,...) / user-id / (user-id),...)} ,RESTART={ALL / (n,...)} ,V={Y / N} ,'text' } [,PAUSE={FORMS / DATASET}][,JOBCLASS=jobclass][,OUTCLASS=s] [,CPRES / ,NOCPRES] [,CPACT / ,NOCPACT] [,TRAN / ,NOTRAN] [,HFC / ,NOHFC] [,ATT / ,UNATT] [,USER=userid]
{MONITOR / MN}	{ JOBNAMES[,T] DSNAME SPACE STATUS A SESS[,T] MSG }
{MOUNT / M}	unitaddr,VOL={(NL,volserial) / (SL,volserial) / (AL,volserial)}[,USE={STORAGE / PUBLIC / PRIVATE}]
{MSGRT / MR}	{(D=(display-operand,...)[,MN=A][,K]) / (REF}
{PAGETUNE / PGT}	DISPLAY[=((][STOP] / [PAGEMEAS] / [REACT] / [STATUS])]
{PAGETUNE / PGT}	STOP=[(]{level / SYS}[)] PAGEMEAS={([ALL= / ln=]frequency[,ln=frequency]...) / frequency / (frequency) / SYS / (SYS)} REACT={[ALL= / ln=]{time / (time[,pagetran]) / (,pagetran)}[,ln={time / (time[,pagetran]) / (,pagetran)}]...) / time / (time[,pagetran]) / (,pagetran) / SYS / (SYS)}

Operator Command Outlines (cont'd)

Operation	Operand
{RELEASE / A}	(ALL / IN [=inclass] / Q [=inclass] / OUT [=outclass]) [,JBN jobname* [,OUT [=outclass[outclass...]]] [,USER=userid] * May be specified up to five times if separated by commas and enclosed in parentheses.
[REPLY / R]	{id / msgno} [.userid] { [,R] 'text' / [,R] text }
{RESET / E}	jobname*, { PRTY =nn / CLASS = c } [,OUT = s][,USER = userid] * May be specified up to five times if separated by commas and enclosed in parentheses.
{ROUTE / RO}	[{A | ALL} / {J | JBN}=jobname] [, {C | CLASS} = class] [, {D | DEST}= userid] [, {G | GROUP} = (class[,class...])] [, {H | HOLD} = {Y | YES / N | NO}] [, {U | USER} = userid]
{SEND / SE}	'text' [,ALL / [,USER = {userid / (userid,userid...)}] / ,OPERATOR =routecode] [,NOW / ,LOGON / ,SAVE] message-no. [,LIST / ,DELETE]
{SET / T}	DATE=yy.ddd[CLOCK =hh.mm.ss]
{SETPRT / SP}	ccu[,nn] [,{ FORMDEF / FD } = { H / P / B }] [,{ NONCRIT / N } = { H / C / B }] [,BURSTER={ Y / n }] [,CGS = { 1 / 2 }] [,LIST]
{START / S}	{ procname . { Pnn / ALL } / procname [.identifier] } [[unitaddr / ,devicetype], [volserial]] [,jobname ,outclass ,jobclass ,(JOBCLASS = class, OUTCLASS=s) ,(parm) ,([MODE = { INT / (INT,S) / EXT }] [,TIME=YES] [,DEBUG=YES] [,BUF=nnn]) [, { SCRT / RLSE / KEEP }] [,ID=x] [,USER=userid] [,keyword=option,...]* * Any valid keyword/option combination that can appear on a DD statement or EXEC statement can follow after the last positional parameter. It can be replaced by: PARM ='jobclasses [,SWA=nnnn / [,EXCPVR=NO] [,FMT=Y]] [,RESV=nn]'

Operator Command Outlines (cont'd)

Operation	Operand
{STARTF / SF}	[name] [.identifier], unitaddr [,,{jobname / outclass}] [,keyword=option,...]
{STOP / P}	({[procname.] / [name.]} identifier * [,USER =userid] unitaddr jobname* JOBNAMES DSNAME SPACE STATUS) May be specified up to five times if separated by commas and enclosed in parentheses. Can be combines with the other parameters that are allowed to be specified up to five times.
{STOPMN / PM}	(JOBNAMES DSNAME SPACE STATUS A SESS MSG)
{SWAP / G}	{OFF / ON / unitaddr, cuu}
{SWITCH / I}	SMF
{UNLOAD / U}	unitaddr
{VARY / V}	{unitaddr[,PATH,cuu] / (unitaddr,unitaddr...) / unitaddr-unitaddr} {ONLINE / OFFLINE}
{VARY / V}	{unitaddr / (I-cuu,O-cuu)},MSTCONS
{VARY / V}	{unitaddr / SYSLOG},HARDCPY [,CMDS / ,NOCMDS / ,OFF / ,INCMDS / ,STCMDS] [,ROUT={ALL / NONE / (routecode[,routecode]...)}]

Operator Command Outlines (cont'd)

Operation	Operand
{VARY / V}	({ unitaddr / O - cuu / (I - cuu, O - cuu) } [, { unitaddr / , O - cuu / , (I - cuu, O - cuu) }] ...) { , ONLINE / , OFFLINE / , CONSOLE [, AUTH={ ALL / INFO / ([SYS][, IO][, CONS]) } / , ROUT={ ALL / NONE / (routecode[, routecode]...) } / , ALTCONS={ unitaddr / O-cuu / (I-cuu, O-cuu) }] }
{WRITELOG / W}	{s / CLOSE}
{WRITER / WTR}	unitaddr, { FSP | F = { nnn / DS | D } / BSP | B = { nnn / DS | D / JOB | J } / LSP | L = { n / c } / HOLD | H / REPEAT | R = { (nnn, JOB | J) / nnn } } [, JBN | J = jobname] [, USER | U = userid]

Definition of Substitutional Operands

c	one input (A-Z, 0-9) or output (A-Z, 0-9) class.
class	one to fifteen job classes (A-Z, 0-9) without priorities.
cuu	the channel and unit address (cuu) on an I/O device.
device	symbolic remote device address used at RES workstation.
devicetype	a unit type, such as 2540 or 1403, of the output device to be used.
eeee	a four digit decimal number indicating an error count.
frequency	the number (0-9) of task dispatchings occurring before invocation of the page measurement routine.
hh.mm.ss	hour (00-23), minute (00-59), and second (00-59).
i	a single input class.
id	a two digit identifier that is identical to the identifier included in the system message.
identifier	a unique one to eight character alphanumeric name that starts with a letter and identifies one task started by a cataloged procedure.
inclass	one to four input queue classes (A-Z, 0-9).
I-cuu, O-cuu	the channel and unit addresses (cuu) of the input (I-cuu) and output (O-cuu) devices that make up a composite console.
jobclass	one to fifteen job classes (A-Z, 0-9). Priority of processing is from left to right.
jobname	the name of a specific problem program that appears on the JOB statement.
keyword=option	any valid keyword/option combination that may appear on a DD statement.
level	the in-use qu position (1-9 or N) on the STOP line.
n	a single digit decimal number.
nnn	a one to three digit decimal number.
outclass	one to eight output classes (A-Z, 0-9).
O-cuu	the channel and unit address (cuu) of an output only console.
pagetran	a number (0-255) of page transmission operations (page-ins and page-outs).
parm	information, of variable format, to be passed to a problem program.
Pnn	a partition number (P00-P15).
procname	the name of a cataloged procedure that resides on SYS1.PROCLIB.
qclass	one to four queue classes (A-Z, 0-9 for input queues, SOUT for the output queue, HOLD for the hold queue).
routecode	a system-to-operator message routing code.
s	a single output class (A-Z, 0-9).
text	information of extremely variable format.
time	a real time interval in seconds (0-9).
tttt	a four digit decimal number indicating an hour limit.
unitaddr	the channel and unit address (cuu) of an I/O device.
volserial	the volume serial number of a disk pack or magnetic tape.
x	a recording mode: either R (record) or Q (quiet).
yy.ddd	the year (00-99) and Julian day (000-366).

RES Workstation Command Outlines

Operation	Operand
{CANCEL / C}	{[JBN=] jobname | (jobname, jobname, . . .) ([,DUMP][,ALL] / ,IN[=class | HOLD] / ,OUT[=class | HOLD]) / [DEV=] unitaddr | (unitaddr, unitaddr, . . .)}
{DISPLAY / D}	{ADD | A / T / R[,LIST] / jobname (jobname, jobname, . . .) [,HOLD] / Q[=list] / N[=list] / RT, {All / ACT / INACT / TERM=term-id[.dev[,term-id.dev]. . .]} [,L | ,LB | ,LS] / USER [,L / =userid]}
{HOLD / H}	jobname | (jobname, jobname, . . .) [,OUT [=outclass [outclass. . .] / =(outclass, outclass, . . .)]]
{LISTBC / LB}	[NOTICES [,MAIL] / MAIL [,NOTICES]]
{LOG / L}	'text'
LOGOFF	[SLOW]
LOGON	userid[/password] TERM (term-id) [PROC(procname)] [NOTICES / NONOTICES] [MAIL / NO MAIL] [UNATT]
{MODIFY / F}	{identifier / procname.id / sfname.id / unitaddr} {,TYPERUN=HOLD | NOHOLD / ,CLASS=classnames [,PAUSE. . .] / ,PAUSE=FORMS | DATASET [,CLASS. . .] / [,CPRES / ,NOCPRES] [,HFC / ,NOHFC] / [,CPACT / ,NOCPACT] [,ATT / ,UNATT] / [,TRAN / ,NOTRAN]}
{MONITOR / MN}	{JOBNAMES [,T] / MSG}
{RELEASE / A}	jobname (jobname, jobname, . . .) [,OUT [=outclass [outclass. . .] / =(outclass, outclass, . . .)]]
[REPLY / R]	id [,/b] [text / 'text']

RES Workstation Command Outlines (cont'd)

Operation	Operand
{RESET | E}	jobname | (jobname, jobname, ...)[,PRTY=priority[,OUT=outclass]] [,CLASS=class,OUT=outclass]
{ROUTE | RO}	{A | ALL | J | JBN=jobname} [,{C | CLASS}=class] [,{D | DEST}=userid] [,{G | GROUP}=(class[,class...])] [,{H | HOLD}={Y | YES | N | NO}]
{SEND | SE}	'text' [,USER={userid | (userid,userid...)}] [,{NOW | LOGON | SAVE}] ,OPERATOR[=route-code]
SETPRT	unitaddr,LIST
{START | S}	procname[.id] [,unitaddr] [,,jobname | ,,outclass] [,keyword=option,...]
{STARTF | SF}	[name] [.identifier],unitaddr [,,jobname] [,,outclass] [,keyword=option,...]
{STOP | P}	[identifier | (identifier,identifier,...)] [procname.identifier | (procname)] [sfname.identifier | (sfname.identifier,sfname.identifier,...)] [unitaddr | (unitaddr,unitaddr,...)] Specify at least one operand, or any combination up to 5.
{STOPMN | PM}	{JOBNAMES | MSG}
{WRITER | WTR}	unitaddr, { FSP | F={nnn | DS | D} BSP | B={nnn | DS | D | JOB | J} LSP | L={n | c} HOLD | H REPEAT | R={(nnn,JOB | J) | nnn} } [,JBN | J=jobname]

Definition of Substitutional Operands

class	specifies an input or output class.
classnames	1-8 output class names to be associated with the writer.
data	specifies information to be passed to the procedure.
devicetype	specifies a device type (for example, PR1).
id	specifies any unique one to eight character name that starts with a letter (except for Pnn or ALL).
inclass	specifies an input queue class.
jobname	specifies the name of a specific problem program.
list	specifies one to four queue classes.
msgno	one or two character identification of a message reply.
n	1, 2, 3 (single space, double space, or triple space).
(n,...)	specifies a single digit decimal number, or a list of numbers.
nnn	specifies a one-to-three digit decimal number.
outclass	specifies an output class.
password	specifies an assigned sequence of one to eight alphameric characters.
pp	specifies numerical priority (decimal number from 0 to 13).
procname	specifies the name of a cataloged procedure.
rdr	specifies the name of the reader procedure being started.
route-code	specifies a value which identifies a central console.
term-id	specifies a unique number (1-200) assigned to a remote terminal.
text	specifies information to be entered in response to a message.
unitaddr	specifies the symbolic unit address (for example, PR1) of an I/O device.
userid	specifies an assigned sequence of one to seven alphameric characters.
wtr	specifies the name of a writer procedure being started.

SMF

SMF

SMFxxxxx parameters*

[OPT={1 / 2}]	1-collect system & job info 2-collect system, job, & job step info
[DSV={0 / 1 / 2 / 3}]	0-no data set or DASD info 1-collect DASD info 2-collect data set info 3-collect data set & DASD info
[REC={0 / 2}]	0-no temporary data set info 2-collect temporary data set info
[EXT={NO / YES}]	NO-no exits YES-take exits
JWT=nnn	nnn-wait state time limit in minutes
[BUF=nnnn]	nnnn-buffer size in bytes (max is 8192)
SID=xxxx	xxxx-system identification
[OPI={YES / NO}]	YES-operator allowed to modify parameters NO-operator not allowed to modify parameters
[MAN={NONE / USER / ALL}]	NONE-no records to SMF data set USER-only user records to SMF data set (type 128-255) ALL-all record types to SMF data set

* This is a SYS1.PARMLIB member that specifies SMF default options to read during SMF initialization.

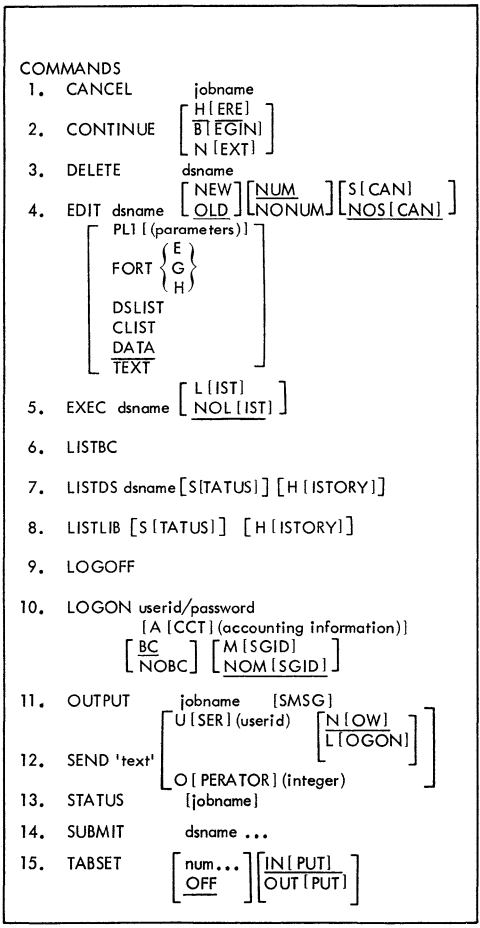
CRJE Macros

Name	Macro	Operands
[name]	CRJELINE	DDLINE=ddname, DDSYSIN=ddname [,RLN= {integer / 1}] [,LERB=([integer1 / 255][,integer2 / 10][,integer3 / 5][,ingeger4 / 5])] ,TYPE= {1050,ADDR=chars / 2740-1 / 2741} [,CODE= {BCD / CORRES / EBCD}] [,FEATURE=([DIAL],[INTERRUPT])] [,ONLNT={NO / YES}]

Name	Operation	Operands
name	CRJETABL	JOB=integer, USERS=integer, SYSCRJE=character [,JOBEXIT=routine name] [,ONEXIT=routine name] [,OFFEXIT=routine name] [,BUFNO={integer / 1}] [,MSGNO={integer / 100}] [,BRDCST={integer / 100}] [,OUTNO={integer / 10}] [,MSGRC={integer / 8}] [,ALIAS=(command name,alias,...)] [,USRMCMD=(command,...)] [,USRSCMD=(subcommand,...)] [,CMDEXIT=routine name] [,PL1LNO={integer / 2}] [,FORTLNO={integer / 2}]

Name	Macro	Operands
[name]	CRJEUSER	[userid,password,...]

CRJE Terminal Command Formats



CRJE Terminal Command Formats (cont'd)

EDIT SUBCOMMANDS

1. linenum [Δ text]
2. CA [NCEL] jobname
3. C [HANGE] linenum [linenum]
   Δ text1 Δ text2 Δ [A [LL]]
4. D [ELETE] [linenum [linenum]]
5. END
6. I [NPUT] [linenum [[increment] [1] / R]] [P [ROMPT] / NOP [ROMPT]]
7. L [IST] [linenum [linenum]] [NUM / NONUM]
8. M [ERGE] { dsname / * } [linenum linenum] [linenum]
9. REN [UMBER] [linenum [increment] / 10 [10]]
10. S [AVE] [dsname] [K [EY] (key)]
11. SC [AN] [linenum [linenum] [ON / OFF]
12. SEND 'text' [U [SER] (userid) [N [OW] / L [OGON]] / O [PERATOR] (integer)]
13. SUB [MIT] { dsname / * } ...
14. TAB [SET] [num... / OFF] [IN [PUT] / OUT [PUT]]

CRJE Terminal Commands and Functions

SESSION MANAGEMENT COMMANDS

Command	Function
LOGON	To identify the user and initiate his session.
LOGOFF	To terminate a session.

DATA MANAGEMENT COMMANDS

General

Command	Function
DELETE	To scratch an VS data set or to remove a CRJE data set from the user's library.
EDIT	To initiate creating or updating operations.

EDIT Subcommands

Subcommand	Abbreviation	Function
INPUT	I	To insert and/or replace lines in the active set.
DELETE	D	To remove lines in the active data set.
Implicit		To enter or delete lines in the active data set.
CHANGE	C	To replace character strings within lines of the active data set.
MERGE	M	To combine another data set with the active data set or to copy lines from one place to another within the active data set.
RENUMBER	REN	To reassign line numbers to the lines in the active data set.
LIST	L	To display lines of the active data set.
SCAN	SC	To request a syntax analysis of PL/1 or FORTRAN source language statements in the active data set.
SAVE	S	To store the active data set in the user's library.
END		To terminate creating and updating operations and to delete the active data set.

JOB PROCESSING COMMANDS

Command	Function
SUBMIT	To enter a job into the VS job input stream. (Can also be used as an EDIT subcommand; it can be abbreviated SUB when used as a subcommand.)
OUTPUT	To request CRJE SYSOUT output of a conversationally-submitted job.
CONTINUE	To resume output listing that was previously interrupted.
CANCEL	To remove a job from the CRJE system and to delete any CRJE SYSOUT output of that job. (Can be used as an EDIT subcommand; it may also be abbreviated CA when used as a subcommand.)

STATUS INFORMATION COMMANDS

Command	Function
LISTLIB	To obtain the name and characteristics of every CRJE data set in the user's library.

STATUS INFORMATION COMMANDS (cont.)

Command	Function
LISTDS	To obtain information about a particular CRJE data set in the user's library.
STATUS	To obtain information about jobs the user has submitted.

MESSAGE COMMANDS

Command	Function
SEND	To send a message to the central operator or to another terminal user. (Can also be an EDIT subcommand.)
LISTBC	To request the broadcast messages.

TABSET COMMAND

Command	Function
TABSET	To indicate the tab settings at the terminal. This command affects all input and output and can be either a command or an EDIT subcommand. (Can only be abbreviated - TAB - as a subcommand.)

EXEC COMMAND

Command	Function
EXEC	To execute a sequence of commands contained in a CRJE data set.

CRJE INSTALLATION VARIABLES

The following functions, restrictions, and assignments are determined by the central installation when the system is generated.

ADDITIONAL COMMANDS AND SUBCOMMANDS

The installation may add commands and subcommands to the system by providing the routines to process them.

COMMAND ALIASES

The installation may assign alternate verbs (aliases) for the CRJE commands and subcommands. Duplication of aliases is allowed between modes but not within the same mode; i.e., the same alias may be used for a command and a subcommand, but it cannot be used for two commands (if in command mode) nor for two subcommands (if in edit mode). Either the CRJE name or the installation alias is recognized when entered from a terminal.

EXIT ROUTINES

Routines may be provided by the installation to check the accounting information on LOGON commands, to check JCL statements of jobs submitted for batch processing, and to obtain accounting information when a user logs off the system. An installation routine may reject a LOGON command and may terminate a job submission.

CRJE Installation Variables (cont'd)

SYNTAX CHECKERS

The installation selects what syntax checkers, if any, are provided in the system and the kind of checking performed (i.e., level of checking or language level supported).

NUMBER OF LINES PER SYNTAX SCAN

The installation can impose a limit on the number of lines one statement can span and still be scanned as a complete statement by the syntax checker.

USERID/PASSWORD

The installation assigns userids and passwords to authorized terminal users.

CRJE SYSOUT CLASS

The system output class used for remote job output to be returned to terminal users is assigned by the installation.

NUMBER OF LINES PER OUTPUT GROUP

The installation specifies how many lines of output are sent to terminal before allowing the terminal user to interrupt the output. This only applies to terminals without a special interrupt feature.

MAXIMUM NUMBER OF JOBS

The maximum number of jobs that can reside in the central system at one time is determined by the installation. When this maximum is reached, no more jobs are accepted until some of the existing jobs are cancelled or their output is returned.

MAXIMUM NUMBER OF MESSAGES

The installation determines the number of messages that can be maintained by the system at any one time. This includes messages waiting for delivery at logon time and messages currently being processed.

ROUTING CODES FOR MULTIPLE CONSOLES

If the central system supports multiple consoles, the installation specifies a routing code for each console. A user may direct a message to an operator at a particular console by specifying the routing code for that console.

ON-LINE TERMINAL TEST

The installation determines whether or not the BTAM On-Line Terminal Test facility is provided. This facility provides tests that can be used by the terminal user as a start-up procedure or by the customer engineer for terminal checkout and diagnosis of terminal failure.

Operation	Operand
BRDCST	C { nnnn,'text' / 'text' / nnnn / DELETE }

Operation	Operand
CENOUT	C, J=jobname, C=class

Operation	Operands
{ MODIFY / F }	[procname.] identifier, { D / A } = (address,...)

Operation	Operand
MSG	C { M='text' [, U=userid [, Q]] / D=userid }

Operation	Operands
SHOW	C { JOBS [,jobname] / USERS [,userid] / ACTIVE [,NUMBER] / BRDCST / MSGS [,userid] / LERB [,lineaddress] / SESS [,userid] / SESSREL [,userid] }

Operation	Operands
{ START / S }	procname.identifier,,,({ FORM / NFMT / NONE } { ,ABNO / ,NORM })

Operation	Operand
{ STOP / P }	[procname,] identifier

Operation	Operands
USERID	C, { { A [DD] / D [ELETE] } = (userid,password) / S [UPPRESS] / R [ESUME] }

This section contains information about the Linkage Editor and the Loader.

Source Publication

Detailed information about the linkage editor and loader is contained in *OS/VS Linkage Editor and Loader,* GC26-3813.

Linkage Editor JCL Statements - Optional/Required

Required and Optional Statements				Notes
Optional See notes.	//jobname	JOB		
	//stepname	EXEC	{ PBM=HEWL } { PGM=LINKEDIT } ,PARM='options'	These names can also be used: IEWL, IEWLF440, IEWLF880, IEWLF128 or as a subprogram: LOAD/CALL, LINK, XCTL or as a subtask: ATTACH
	//SYSLIN	DD	dataset reference	Primary input data set: *-for an immediately following data set. DSNAME =data set. DISP=(OLD, DELETE) for a cataloged data set
opt.	//SYSLIB	DD	dataset reference	For automatic call: DSNAME=library, DISP=SHR libraries are; SYS1.ALGLIB, SYS1.COBLIB, SYS1.FORTLIB, SYS1.PL1LIB, SYS1.SORTLIB
	//	DD	DDNAME=SYSIN	Reference to linkage editor control statements if not included with SYSLIN data
	//SYSUT1	DD	dataset reference	Intermediate data set
	//SYSPRINT	DD	dataset reference	Diagnostic output data set
	//SYSLMOD	DD	dataset reference	Output module library
opt.	//SYSTERM	DD	dataset reference	Required only if PARM=TERM specified on EXEC statement
	//ddname	DD	dataset reference	One for each INCLUDE or LIBRARY reference
opt.	//Linkage Editor Control Statements			In addition to or if not defined as a data set by the //SYSLIN DD statement
opt.	//Object Module(s)			In addition to or if not defined as a data set by the //SYSLIN DD statement
	/*			End of linkage editor input and job step.
	//			End of job

Linkage Editor Execute Statement

Execute Statement:

PARM='options'

options are:

AC(1)	-	assign an authorization code
NE	-	not editable; no ESD produced in load module. NE is ignored if MAP or XREF specified.
OL	-	only load; a LOAD and branch instruction or CALL required to load and enter module.
OVLY	-	overlay; must be present if OVERLAY or INSERT statements are used. Not for use with refreshable, re-enterable, or serially reusable programs.
RENT	-	re-enterable; all CSECTs must be re-enterable or RENT is ignored.
REUS	-	reusable; all CSECTs must be re-enterable or serially reusable or REUS is ignored.
REFR	-	refreshable; all CSECTs must be refreshable or REFR is ignored.
XCAL	-	exclusive call; must be specified with OVLY.
LET	-	allow execution; execution of the module may be attempted even if severity 2 errors have occurred during linkage editing.
NCAL	-	no automatic library call; library members are not called to resolve external references. A SYSLIB DD statement need not be supplied.
ALIGN2	-	align on page boundary; used with PAGE or ORDER with P operand statements to cause alignment of CSECTs on 2K page boundary. Default is 4K alignment.
SIZE= (value1, value2)	- -	size; value1 is virtual storage available for linkage editor with minimum of 65,536 and default of 196,608. Value2 is load module buffer with minimum of 6144, maximum of 102,400, and default of 65,536.
DCBS	-	allow specification of DCB for SYSLMOD - block size must be specified in DCB parameter of SYSLMOD DD statement.
LIST	-	list linkage editor control statements; statements appear in card-image format on diagnostic output data set.
MAP	-	map the output module; the map appears on the diagnostic output data set.
XREF	-	produce cross reference table; cross reference table, including map, appear on diagnostic output data set. MAP need not be used with XREF.
TERM	-	print diagnostics on data set specified by SYSTERM DD statement; if SYSTERM DD statement is not included, TERM is ignored.

Execute Statement (cont'd) - SIZE and REGION Parameter Guidelines

PARM default attributes for the linkage editor:

not overlay
not tested
block format
not refreshable
not re-enterable
not serially reusable

Execute Statement:

REGION parameter

REGION=value - if SIZE= was specified in PARM, partition size must be 10K larger than $value_1$.

SIZE AND REGION PARAMETER GUIDELINES

Guidelines for determining an appropriate REGION parameter value and SIZE parameter values for a linkage editor job step:

First - determine $Value_2$ of the SIZE parameter.

$$Value_2 = \begin{bmatrix} 6K \\ 6144 \\ J \\ \ell \end{bmatrix} \leq \begin{bmatrix} a+b \\ c \times d \\ c \times e \end{bmatrix} \leq a+b$$

where: a is the length of the load module to be built
b is 0, if the length of the load module to be built is $< \begin{bmatrix} 40K \\ 40960 \end{bmatrix}$ or $\begin{bmatrix} 4K \\ 4096 \end{bmatrix}$ if the length of the load module to be built $\geq \begin{bmatrix} 40K \\ 40960 \end{bmatrix}$
c is an integer ≥ 2
d is the track capacity of the SYSLMOD device
e is the block size of the SYSLMOD data set
J is the length of the largest text record in load module input
ℓ is the track capacity of the SYSUT1 device

Second - determine $Value_1$ of the SIZE parameter

$Value_1 = f + g + h$. $Value_1$ must range between f and $\begin{bmatrix} 999K \\ 999999 \end{bmatrix}$

where: f is the design point of the Linkage Editor being used:

$f = \begin{bmatrix} 64K \\ 65536 \end{bmatrix}$

g is the excess of $Value_2$ over $\begin{bmatrix} 6K \\ 6144 \end{bmatrix}$

$g = Value_2 - \begin{bmatrix} 6K \\ 6144 \end{bmatrix}$

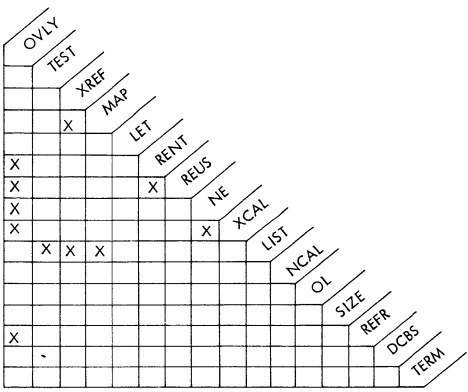
h is the additional storage required to support the blocking factor for SYSLIN, any object module libraries, and SYSPRINT:

	5 to 1	10 to 1	40 to 1
F64	-	$\begin{bmatrix} 18K \\ 18432 \end{bmatrix}$	$\begin{bmatrix} 28K \\ 28672 \end{bmatrix}$

Third - determine the REGION parameter.

$$REGION = Value_1 + \begin{bmatrix} 10K \\ 10240 \end{bmatrix}$$

Linkage Editor Incompatible Job Steps - Return Codes



Incompatible Job Step Options for the Linkage Editor

Linkage Editor Return Codes

Return Code	Severity Code	Description
00	0	Normal conclusion.
04	1	Warning messages have been listed, execution should be successful. For example, if the overlay option is specified and the overlay structure contains only one segment, a return code of 04 is issued.
08	2	Error messages have been listed, execution may fail. The module is marked not executable unless the LET option is specified. For example, if the block size of a specified library data set cannot be handled by the linkage editor, a return code of 08 is issued.
12	3	Severe errors have occurred, execution is impossible. For example, if an invalid entry point has been specified, a return code of 12 is issued.
16	4	Terminal errors have occurred, the processing has terminated. For example, if the linkage editor cannot handle the blocking factor requested for SYSPRINT, a return code of 16 is issued.

Linkage Editor Control Statements

Operation	Operand
ALIAS	{ symbol / external name } [, symbol / , external name] ...
CHANGE	external symbol(newsymbol)[, external symbol(newsymbol),...]
ENTRY	externalname
EXPAND	name (xxxx) [,name(xxxx)]...
IDENTIFY	csectname('data')[,csectname('data')]...
INCLUDE	ddname[(membername[,membername]...)] [,ddname[(membername[,membername]...)]...]
INSERT	csectname[,csectname,...]
LIBRARY	{ ddname(membername[,membername]...) / (externalreference[,externalreference]...) / *(externalreference[externalreference]...) } ,...
NAME	membername[(R)]
ORDER	{ common area name / csectname } [(P)] [, { common area name / csectname } [(P)]] ...
OVERLAY	symbol[(REGION)]
PAGE	{ common area name / csectname } [, { common area name / csectname }]...
REPLACE	{ csectname-1[(csectname-2)] / entry name } ,...
SETCODE	AC(1)
SETSSI	xxxxxxxx

Linkage Editor Record Formats - Capacities

The following record formats are used with the linkage editor:

F -- The records are fixed length.

FB -- The records are fixed length, and blocked.

FBA -- The records are fixed length, blocked, and contain ANSI control characters.

FBS -- The records are fixed length, blocked, and written in standard blocks.

FA -- The records are fixed length and contain ANSI control characters.

FS -- The records are fixed length and written in standard blocks.

U -- The records are undefined length.

UA -- The records are undefined length and contain ANSI control characters.

Capacities

Function		Capacity
Virtual storage allocated (in bytes)		64K
Maximum number of entries in composite external symbol dictionary (CESD)		558
Maximum number of intermediate text records		372
Maximum number of relocation dictionary (RLD) records		192
Maximum number of segments per program		255
Maximum number of overlay regions per program		4
Maximum blocking factor for input object modules (number of 80-column card images per physical record)		10[1]
Maximum blocking factor for SYSPRINT output (number of 121-character logical records per physical record)		10[1]
Output text record length (in bytes)	On IBM 2314, 2319 Storage Facility	3072[2]
	On IBM 2305-2 Fixed Head Storage Facility	3072[2]
	On IBM 3330, 3340, or 3350 Disk Storage Facility	3072[2]

[1] From 74K to 9999K for value$_1$ of the SIZE option, the blocking factor for input object modules and SYSPRINT output is 40.

[2] The maximum output text record length is achieved when value$_2$ of the SIZE parameter is at least twice the record length size. For example, on a 3330, 12288 byte records are written when value$_2$ is at least 24576.

Loader JCL Statements

```
//name      JOB   parameters   (optional)
//name      EXEC  PGM=LOADER,PARM=(parameters)
//SYSLIN    DD    parameters
//SYSLIB    DD    parameters   (optional)
//SYSLOUT   DD    parameters   (optional)
//SYSTERM   DD    parameters   (optional)
//      (optional DD statements and data required for loaded program)
```

Input Deck for the Loader -- Basic Format

Loader EXEC Statement

The two loader names are:

1. LOADER
2. IEWLDRGO

Loader Execute Statement

MAP. The loader produces a map of the loaded program that lists external names and their absolute storage addresses on the SYSLOUT data set. (If the SYSLOUT DD statement is not used in the input deck, this option is ignored.)

NOMAP. A map is not produced.

RES. An automatic search of the link pack area queue is to be made. This search is always made after processing the primary input (SYSLIN), and before searching the SYSLIB data set. When this option is specified, the CALL option is automatically set.

NORES. No automatic search of the link pack area queue is to be made.

CALL. An automatic search of the SYSLIB data set is to be made. (If the SYSLIB DD statement is not included in the input deck, this option is ignored.)

NOCALL or NCAL. An automatic search of the SYSLIB data set will not be made. When this option is specified, the NORES option is automatically set.

LET. The loader will try to execute the object program even though a severity 2 error condition is found. (A severity 2 error condition is one that could make execution of the loaded program impossible.)

NOLET. The loader will not try to execute the loaded program if a severity 2 error condition is found.

SIZE=size. Specifies the size, in-bytes, of dynamic virtual storage that can be used by the loader.

EP=name. Specifies the external name to be assigned as the entry point of the loaded program. This parameter must be specified if the entry point of the loaded program is in an input load module. For FORTRAN, ALGOL, and PL/1, these entry points must be MAIN, IHIFSAIN, and IHENTRY, respectively.

NAME=name. Specifies the name to be used to identify the loaded program to the system. If this parameter is not used, the loaded program will be named **GO.

PRINT. Informational and diagnostic messages are produced on the SYSLOUT data set.

NOPRINT. Informational and diagnostic messages are not produced on the SYSLOUT data set. SYSLOUT is not opened.

TERM

Numbered diagnostic messages are to be sent to the SYSTERM data set. The SYSTERM data set can be used to replace or supplement the SYSLOUT data set at any time. (If the SYSTERM DD statement is not included in the input deck, this option is ignored.)

NOTERM

Numbered diagnostic messages are not to be sent to the SYSTERM data set.

Unless otherwise specified with the LOADER macro instruction during system generation, the default options are: NOMAP, RES, CALL, NOLET, SIZE=100K, and PRINT. The default options NAME=**GO and NOTERM cannot be changed during system generation.

DD Statements - Loader Macros

The following considerations apply to the DCB parameter of SYSLIN, SYSLIB, and SYSLOUT.

- For better performance, BLKSIZE and BUFNO can be specified.
- If BUFNO is omitted, BUFNO=2 is assumed.
- Any value given to BUFNO is assumed for NCP (number of channel programs).
- If RECFM=U is specified, BUFNO=2 is assumed, and BLKSIZE and LRECL are ignored.
- RECFM=V is not accepted.
- RECFM=FBSA is always assumed for SYSLOUT.
- If RECFM is omitted, RECFM=F is assumed for SYSLIN and SYSLIB.
- If BLKSIZE is omitted, the value given to LRECL is assumed.
- LRECL=121 is assumed for SYSLOUT.
- If LRECL is omitted, LRECL=80 is assumed for SYSLIN and SYSLIB.
- If OPTCD=C is used to specify chained scheduling, an additional 2K (2048 bytes) of virtual storage is needed in the user's region if the necessary data management routines are not resident.

Note: The SYSTERM data set will always consist of unblocked 81-character records with BUFNO=2 and RECFM=FSA. Because these values are fixed, the DCB parameter need not be used.

In addition to the DD statements used by the loader, any DD statements and data required by the loaded program must be included in the input deck.

Loader Macros

Name	Operation	Operand
[symbol]	{LINK / ATTACH}	EP=loadername PARAM=(optionlist[,ddname list]) VL=1
	{LOAD / XCTL}	EP=loadername

Macro Instruction Basic Format

Loader Macro Parameters

EP

specifies the symbolic name of the loader. The entry point at which execution is to begin is determined by the control program from the library directory entry.

PARAM

specifies, as a sublist, address parameters to be passed to the loader. The first fullword in the address parameter list contains the address of the option list for the loader and/or loaded program. The second fullword contains the address of the ddname list. If standard ddnames are to be used, this list may be omitted.

option list

specifies the address of a variable length list containing the loader and loaded program options. This address must be written even though no list is provided.

The option list must begin on a halfword boundary. The two high-order bytes contain a count of the number of bytes in the remainder of the list. If no options are specified, the count must be zero.

The option list is free form, with the loader and loaded program options separated by a slash (/), and with each option separated by a comma. No blanks or zeros should appear in the list.

ddname list

specifies the address of a variable length list containing alternative ddnames for the data sets used during loader processing. If the standard ddnames are used, this operand may be omitted.

The format of the ddname list is identical to the format of the ddname list for invoking the linkage editor; the 8-byte entries in the list are as follows:

Entry	Alternate Name For:
1	SYSLIN
2	not applicable
3	not applicable
4	SYSLIB
5	not applicable
6	SYSLOUT
7-11	not applicable
12	SYSTERM

VL

specifies that the sign bit is to be set to 1 in the last fullword of the address parameter list.

Loader Return Codes

Return Code	Loader Return Code[1]	Loaded Program Return Code	Conclusion or Meaning
0	0	0	Program loaded successfully, and execution of the loaded program was successful.
	4	0	The loader found a condition that may cause an error during execution, but no error occurred during execution of the loaded program.
	8 (LET)	0	
4	0	4	Program loaded successfully, and an error occurred during execution of the loaded program.
	4	4	The loader found a condition that may cause an error during execution, and an error did occur during execution of the loaded program.
	8 (LET)	4	
8	0	8	Program loaded successfully, and an error occurred during execution of the loaded program
	4	8	The loader found a condition that may cause an error during execution, and an error did occur during execution of the loaded program.
	8 (LET)	8	
	8		The loader found a condition that could make execution impossible. The loaded program was not executed.
12	0	12	Program loaded successfully, and an error occurred during execution of the loaded program.
	4	12	The loader found a condition that may cause an error during execution, and an error did occur during execution of the loaded program.
	8(LET)	12	
	12		The loader could not load the program successfully, execution impossible.
16	0	16	Program loaded successfully, and the loaded program found a terminating error.
	4	16	The loader found a condition that may cause an error during execution, and a terminating error was found during execution of the loaded program.
	8(LET)	16	
	16		The loader could not load program, execution impossible.

[1]Error diagnostics (SYSLOUT and/or SYSTERM data set) for the loader will show the severity of errors found by the loader.

Loader Virtual Storage Requirements

Consideration		Approximate Virtual Storage Requirements (in bytes)	Comments
Loader Code	Control	700	--
	Processing	13664	--
Data Management		6K	BSAM
Object Module Buffers and DECBs		BUFNO(BLKSIZE+24)	Concatenation of different BLKSIZE and BUFNO must be considered. (Minimum BUFNO=2)
Load Module Buffer and DECBs.		304	--
SYSTERM DCB Buffers, and DECBs		312	Allocated if TERM option is specified
SYSLOUT Buffers and DECBs		BUFNO (BLKSIZE + 24)	Buffer size rounded up to integral number of double words. (Minimum BUFNO=2)
Size of program being loaded		Program Size	Program size is restricted only by available virtual storage
Each external relocation dictionary entry		8	--
Each external symbol		20	--
Largest ESD number		4n n is the largest ESD number in any input module	Allocated in increments of 32 entries
Fixed Loader Table Size		1260	Subtract 88 if NOPRINT is specified
System Requirements		1600	--

This section contains information concerned with telecommunication access methods and devices.

Source Publications

Detailed information about BTAM, TCAM, and VTAM is contained in these publications:

- *OS/VS BTAM*, GC27-6980
- *OS/VS TCAM Programmer's Guide*, GC30-2054
- *OS/VS Virtual Telecommunications Access Method (VTAM) Macro Language Reference Manual*, GC27-6995
- *VTAM Concepts and Planning*, GC27-6998

Name	Operation	Operand	
[symbol]	AS	ID=absexp[,ASGROUP=symbol] [,DEGROUP=(symbol,absexp)]	2715 ONLY
[symbol]	ASCTR	ID=absexp, HIGHCTR=absexp, ROUTE=([{CPU / DISK}] [,LOG] [,ASLOG] [,EXTRALRM]) [,NEXTAS=absexp]	2715 ONLY
[symbol]	ASLIST	device-code,NORM=absexp [,LENGTH=(absexp1,absexp2)] [,DIGIT=(absexp1,absexp2,absexp3)] [,ENTRY={1 / M}][,MSG='text'] [,INQDISP=absexp] [,MODULUS=(absexp1,absexp2, absexp3)] [,SELTRAN={NO / YES}]	2715 ONLY
(Omit)	ASMTRTAB	tablename,...	
[symbol]	CHGNTRY	list:addr,listype,listposition, numchars,action	
[symbol]	{OPEN / CLOSE}	({dcb,,}...), [MF=L / MF=(E,listname)]	
[symbol]	CONFIGUR	[CORE={16 / 32}] [,PC={NO / YES}] [,GDU={NO / YES}] [,FUNCERR=(absexp,...)] [,ENDERR=(absexp,...)] [,MONERR=(absexp,...)] [,GETID=absexp] [,STORID=absexp] [,IDCOUNT=absexp] [,INQDISP={NO / YES}]	2715 ONLY
[symbol]	CTRGROUP	ctrno, [sro], [cttest], ID=absexp [,SROENAB={NO / YES}] [,CTINIT={NULL / NCT / UNASP}]	2715 ONLY

BTAM Macros (cont'd)

Name	Operation	Operand	
[symbol]	CTRLIST	DEVCOD={B / C / M}, CTRADR={IMP / EXP}, CTRRD={SINGLE / GROUP}, CTTEST=(NULL / SETNCT / SETUNAS / RESET) CTROP=(READ / SET / READSET / READRST / RDRESID / NULL) [,MSG='text']	2715 ONLY
[symbol]	CTRSCHED	sched,...	2715 ONLY
	DATAMGT	ACSMETH=BTAM	
symbol	DCB	keyword operands	
[symbol]	DEULIST	[DIGIT=(absexp1,absexp2)] [,LENGTH=absexp1] [,MSG='text'] [,MODULUS=(absexp1,absexp2)] [,DIGIT2=(absexp1,absexp2)]	2715 ONLY
symbol	DFTRMLST	list type, device-dependent operands	
[symbol]	DISPGUID	DISPMSG='text' [,SUPPRES={YES / NO}]	2715 ONLY
[symbol]	GDUAS	ID=absexp, GDUNUMB=absexp	
[symbol]	GDULIST	PARAMNO=absexp [,(NORGUID=absexp,...)] [{DISPMSG=symbol / IDENT=absexp}] [,MSG='text'] [,ENTRY={I / M}]	
[symbol]	GDUTRANS	TRCODE=absexp, TRLIST=symboln	
[symbol]	IODEVICE	UNIT=type, ADDRESS=address, ADAPTER=type, TCU=type, MODEL=model, [,FEATURE=(feature1,feature2,...)] [,SETADDR=type] [,OBRCNT=n]	

Name	Operation	Operand	
symbol	LERB	nlines[, { ([transmct] [,datack], [,intreq] [,nontto]) } ...]	
[symbol]	LERPRT	dcbaddr[,rln][,cid] [,CLEAR=YES / ,CLEAR=NO]	
[symbol]	LOPEN	decbaddr	
[symbol]	ONLTST	DECB=decb address, X=type of test, Y=no. of transmissions, DCB=dcb address, AREA=rft message area [,TEXT=user text area, LENGTH=user text length] [,ENTRY=list address] [,RLN=line number]	
[symbol]	OPEN	See CLOSE	
[symbol]	PARAMNUM	PLN=absexp, PARMLST=symboln	2715 ONLY
[symbol]	PARMLIST	[CKLNGTH=(length-absexp, errguidance-absexp,...)] [,CKMONKY={NO / YES}] [,CKMOD11=(length-absexp, position-absexp, errguidance-absexp,...)] [,CKRANGE=(position1-absexp, position2-absexp, hilowchars-absexp,...)] [,LOWGUID=(absexp,...)] [,HIGUID=(absexp,...)] [,RNGETST={ERROR / DATA}] [,CKMOD10=(length-absexp, position-absexp, errguidance-absexp,...)] [,CKOR=(position-absexp, checkchar1-hexchar,... checkcharn-hexchar)] [,ORGUID=(absexp,...)] [,CKAND=(position1-absexp, position2-absexp, checkchar1-hexchar, checkcharn-hexchar)] [,ANDGUID=(absexp,...)] [,CKNONUM=(position1-absexp, position2-absexp, errguidance-absexp,...)] [,CKNUM=(position1-absexp, position2-absexp, errguidance-absexp,...)] [,TRANSL={NO / YES}] [,IDENT={NO / YES}]	

BTAM Macros (cont'd)

Name	Operation	Operand	
[symbol]	{READ / WRITE}	decbaddr,optype,dcbaddr, {[inoutarea] / ([inarea],[outarea])}, {[inoutlength] / ([inlength],[outlength])}, [entry], [rln] [,MF=L / ,MF=E]	
[symbol]	RELBUF	dcbaddr,bufferaddr	
[symbol]	REQBUF	dcbaddr,returnreg,[count]	
[symbol]	RESETPL	decbaddr [,POLLING / ,ANSRING]	
[symbol]	RESETPL	decbaddr [,ATTENT]	LOCAL 3270 ONLY
[symbol]	STEND		2715 ONLY
symbol	TGROUP	[TCn=(symboln [,E])]	2715 ONLY
[name]	TPEDIT	MINLN=n[,REPLACE={X'19' / X'xx'}] [,EDIT={EDITD / EDITR}] [,RECFM={V / U}] [,ERROPT={IGNORE / name}] [,VERCHK={NOCHK / VOKCHK}] [,BUFFER={NO / YES}]	IBM 50 MAGNETIC DATA INSCRIBER ONLY
[symbol]	TRANSLAT	TRANSCH=hexchar,TRANTXT='text'	
symbol	TRLIST	TRID=absexp1 [,ROUTE=[{DISK / CPU}]] [,LOG] [{,NULL / ,absexp2}] [,TEXT={NO / YES}] [,INQDISP={NO / YES}] [,DEMOD10={NO / YES}] [,DEMOD11={NO / YES}] [,GDU={NO / YES}]	2715 ONLY
[symbol]	TRNSLATE	[dcbaddr], tablename, area, length	
symbol	{TRSLRCTW / TRSLRCT3}	Fx=code,...	World Trade Telegraph Terminal
symbol	{TRSLSCTW / TRSLSCT3}	Xyy=Fx,...	World Trade Telegraph Terminal
[symbol]	TWAIT	(returnreg), ECBLIST=ecb list addr	

BTAM Macros (cont'd)

Name	Operation	Operand
[symbol]	WAIT	[count] ECB=ecb address, ECBLIST=ecb list addr
[symbol]	WRITE	See READ

BTAM Macro Instruction Format

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
ASMTRTAB	tablename												x
CHGNTRY	listaddr**			x				x					
	dcbaddr***			x				x					
	listype												x
	listposition			x					x				
	numchars**			x					x				
	action												x
CLOSE	dcb							x					
	MF=												x
	listname			x	x		x						
DCB	DSORG=												x
	MACRF=												x
	DDNAME=	x											
	BUFNO=								x				
	BUFL=								x				
	BUFCB=							x					
	EXLST=							x					
	BFTEK=												x
	LERB=							x					
	EROPT=												x
	DEVD=												x
	MODE=												x
	CODE=												x
	READYQ=							x					x
DFTRMLST	listype												x
	xx											x	
	yy											x	
	dialcount		x										
	dialchars										x		
	numsent		x										
	sentchar											x	
	numcnsent		x										
	cntrlseq											x	
	tidseq											x	
	numrec		x										
	ridseq											x	
	AN												As Shown
	MD												As Shown
	AD												As Shown

* See macro description for allowable values.
** Does not apply to local 3270 display.
*** Applies only to local 3270 display.

BTAM Macro Instruction Format (cont'd)

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-Type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
DFTRMLST (cont'd)	entrylength		x										
	userlength		x										
	idcount		x										
	idsent											x	
	authsequence											x	
	controlvalue		x										
	userdata							x					
LERB	nlines								x				
	transmct								x				
	datack								x				
	intreq								x				
	notto								x				
LERPRT	dcbaddr	x		x	x								
	rln			x		x			x				
	cid			x				x					
	CLEAR=												x
LOPEN	decbaddr	x		x									
ONLTST	DECB=			x	x		x						
	X=			x					x				
	Y=			x					x				
	DCB=			x			x						
	AREA=			x			x						
	TEXT=			x			x						
	LENGTH=			x					x				
	ENTRY=			x			x						
	RLN=			x					x				
OPEN	dcb							x					
	MF=												x
	listname			x	x		x						
READ (list form, MF=L)	decbaddr	x											
	optype												x
	dcbaddr							x					
	inoutarea							x					
	inarea							x					
	outarea							x					
	inoutlength								x				
	inlength								x				
	outlength								x				
	entry							x					
	rln								x				
	MF=L												As Shown

* See macro description for allowable values.

BTAM Macro Instruction Format (cont'd)

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-Type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code *
READ (Execute form, MF=E)	decbaddr			x	x		x						
	optype												x
	dcbaddr			x			x						
	inoutarea			x			x						's'
	inarea			x			x						's'
	outarea			x			x						
	inoutlength			x					x				's'
	inlength			x					x				's'
	outlength			x					x				
	entry			x			x						's'
	rln			x					x				
	MF=E												As Shown
READ (Standard form)	decbaddr	x											
	optype												x
	dcbaddr			x				x					
	inoutarea			x				x					's'
	inarea			x				x					's'
	outarea			x				x					
	inoutlength			x					x				's'
	inlength			x					x				's'
	outlength			x					x				
	entry			x				x					
	rln			x					x				
RELBUF	dcbaddr			x	x		x						
	bufferaddr			x									
REQBUF	dcbaddr			x	x		x						
	returnreq			x									
	count			x		x			x				
RESETPL	decbaddr			x	x			x					
	POLLING												As Shown
	ANSRING												As Shown
	ATTENT												As Shown
TRNSLATE	dcbaddr			x			x						
	tablename			x			x						
	area			x			x						
	length			x		x			x				's'
TRSLRCTW	Pnn=												x
TRSLRCT3	Pnn=												x
TRSLSCTW	Xyy=												x
TRSLSCT3	Xyy=												x
TWAIT	Returnreg			x									
	ECBLIST=			x			x						
WAIT	count			x		x	x		x				
	ECB=			x	x		x						
	ECBLIST=			x	x								

* See macro description for allowable values.

BTAM Macro Instruction Format (cont'd)

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-Type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code *
WRITE (List form, MF=L)	decbaddr	x											
	optype												x
	dcbaddr							x					
	inoutarea							x					
	inarea							x					
	outarea							x					
	inoutlength								x				
	inlength								x				
	outlength								x				
	entry							x					
	rln								x				
	MF=L												As Shown
WRITE (Execute form, MF=E)	decbaddr			x	x		x						
	optype												x
	dcbaddr			x			x						
	inoutarea			x			x						
	inarea			x			x						's'
	outarea			x			x						
	inoutlength			x					x				's'
	inlength			x					x				's'
	outlength			x					x				
	entry			x			x						
	rln			x					x				
	MF=E												As Shown
WRITE (Standard form)	decbaddr	x											
	optype												x
	dcbaddr	x						x					
	inoutarea	x						x					
	inarea	x						x					's'
	outarea	x						x					
	inoutlength	x							x				's'
	inlength	x							x				's'
	outlength	x							x				
	entry	x						x					
	rln	x							x				

* See macro description for allowable values.

2715 User Table Macro Instructions

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
AS	ID=								x				
	ASGROUP=	x											
	DEGROUP= tgroupname deunumber	x							 x				
ASCTR	ID=								x				
	HIGHCTR=								x				
	ROUTE=												x
	LOG												As Shown
	ASLOG												As Shown
	EXTALRM												As Shown
	NEXTAS=								x				
ASLIST	device												x
	NORM=								x				
	LENGTH= data length gdlight2								 x x				
	DIGIT= entrypos compvalue gdlight3								 x x x				
	ENTRY=												x
	MSG=									x			
	INQDISP=								x				
	MODULUS= entry pos fld length gdlight3								 x x x				
	SELTRAN=												x
CONFIGUR	CORE=												x
	PC=												x
	GDU=												x
	FUNCERR=								x				
	ENDERR=								x				
	MONERR=								x				
	GETID=								x				
	STORID=								x				
	IDCOUNT=								x				
	INQDISP=												x

*See macro description for allowable values.

2715 User Table Macro Instructions (cont'd)

Macro Instruction	Operand	Sym	Dec Dig	Register (2-12)	Register (1)	Register (0)	RX-type	Rel Exp	Abs Exp	Char	Dec Char	Hex Char	Code*
CTRGROUP	ctrno								x				
	sro								x				
	cttest								x				
	ID=								x				
	SROENAB=												x
	CTINIT=												x
CTRLIST	DEVCOD=												x
	CTRADR=												x
	CTRRD=												x
	CTTEST=												x
	CTROP=												x
	MSG=									x			
CTRSCHED	sched								x				
DEULIST	LENGTH=		x										
	DIGIT= entrypos compvalue								 x x				
	MSG=									x			
	MODULUS= entrypos fld length								 x x				
	DIGIT2= value pos comp value								 x x				
STEND	no operands												
TGROUP	TCn= tcode E	x											As Shown
TRLIST	TRID=								x				
	ROUTE=												x
	LOG												As Shown
	NULL												As Shown
	TEXT=												x
	INQDISP=												x
	DEMOD10=												x
	DEMOD11=												x
	GDU=												x

*See macro description for allowable values.

Line and Station Configuration Supported by BTAM

Start-Stop Communications

1. Nonswitched lines (point-to-point or multipoint), using programmed polling:

   IBM 1030 Data Collection System
   IBM 1050 Data Communications System
   IBM 1060 Data Communications System
   IBM 2260 Display Station --
      IBM 2848 Display Control
      (Remote -- 2701 only)
   IBM 2265 Display Station -- IBM 2845
      Display Control (Remote -- 2701 only)
   IBM 2740 Communications Terminal (Model 1):
      Basic: with checking[1]; with Station Control[2]; with Checking and Station control[2]; or with Checking and IBM 2760 Optical Image Unit features (point-to-point only, if 2740 is equipped with 2760 Optical Image Unit)
      (Model 2): Basic or with Checking[1]
   IBM 2741 Communications Terminal
   Western Union Plan 115A Outstations
   AT&T 83B3 Selective Calling Stations

2. Switched lines:

   IBM 1050 Data Communications System
   IBM 2740 Communications Terminal
      (Model 1): Dial; Dial, with Checking;
      Dial, with Transmit Control; Dial, with Checking and
      Transmit Control, or Dial, with Checking and IBM 2760
      Optical Image Unit features
   IBM 2741 Communications Terminal
   WU Model 33/35 Teletypewriter
      Exchange Terminal (TWX)

3. Nonswitched multipoint lines using the Auto Poll facility (IBM 2702 or 2703 only);

   IBM 1030 Data Collection System
   IBM 1050 Data Communications System
   IBM 1060 Data Communications System
   IBM 2740 (Model 1 and 2): with Station Control[2]
      or with Station Control[2] and Checking features

[1]Used as a regular terminal or as an operator's console, when the operating system includes the Multiple Console Support.

[2]Station Control feature cannot be used if the 2740 is also used as a console under Multiple Console Support.

Line and Station Configuration Supported by BTAM (cont'd)

Binary Synchronous Communications

1. Nonswitched point-to-point and switched point-to-point lines:

   IBM System/370[3]

   IBM System/360 Model 20

   IBM System/3

   IBM 1130 Computing System

   IBM 1800 Data Acquisition and Control System

   IBM 2715 Transmission Control Unit (Model 1 attaches directly to multiplexer channel of central computer; Model 2 communicates with central computer via IBM 2701 or 2703)

   IBM 2770 Data Communications System

   IBM 2780 Data Transmission Terminal

   IBM 3735 Programmable Buffered Terminal
   IBM 3741 Data Station
   IBM 3747 Data Converter
   IBM 3750 Switching System (World Trade users only; nonswitched point-to-point line only)
   IBM 3780 Data Communication Terminal

2. Switched point-to-point

   IBM 3275 Equipped with dial feature
   IBM 5275 Direct Numerical Control Station

3. Nonswitched multipoint lines:

   IBM System/360 Model 20

   IBM System/3

   IBM 1130 Computing System

   IBM 1800 Data Acquisition and Control System

   IBM 2715 Transmission Control Unit (Model 1 attaches directly to multiplexer channel of central computer; Model 2 communicates with central computer via IBM 2701 or 2703)

   IBM 2770 Data Communications System

   IBM 2780 Data Transmission Terminal

   IBM 2972 (Models 8 and 11) General Banking Terminal System

   IBM 3270 Display System (remote)
   IBM 3735 Programmable Buffered Terminal (requires special feature)
   IBM 3780 Data Communication Terminal
   IBM 5275 Direct Numerical Control Station

[3]The remote System/370 may be a Model 135, 145, 155, 158, 165, 168, or 195.

Name	Operation	Operands
[symbol]	CANCELMG	[mask] [,CONNECT= { AND / OR }] [,LEVEL= { BLK / MSG }]
[symbol]	CHECK	decbname
[symbol]	CHECKPT	(no operands)
[symbol]	CKREQ	(no operands)
[symbol]	CLOSE	(dcbname,,...) MF = { L / (E, listname) }
[symbol]	CLOSE (MCP)	(dcbname,,...)
[symbol]	CODE	[{ tablename / NONE / (register) }]
[symbol]	COMMBUF	LIST=name, MAXDEEP=integer
[symbol]	COUNTER	opfield
[symbol]	CTBFORM	[opfield][,DVCID= { NO / YES }][,ENDCHAR= { NO / YES }] [,INSERT= { NO / YES }]
[symbol]	CUTOFF	integer
[symbol]	DATETIME	[DATE= { NO / YES }] [,TIME= { NO / YES }]
[symbol]	ERRORMSG	[mask] [,CONNECT= { AND / OR / NAND }] [,DEST= { destination name / opfield / ORIGIN / DESTIN }] ,DATA= { message / fieldname } [,EXIT=name of routine]
[symbol]	FORWARD	[DEST= { destname / opfield / (number) / PUT / * * / ORIGIN / REG(number) }] [,EOA=characters] [,EXIT=name] [,THRESH=integer]
[symbol]	GET	dcbname [,areaname]

Name	Operation	Operands
grpname	GROUP	MH=mhname [,BUFSIZE=integer] [,BUFMAX={integer / 2}] [,BUFOUT={integer / 2}] [,PCI={(,A) / (,N)}] [,TRANS={tablename / EBCD}] [,INVLIST=(listname,...)] [,RESERVE=(integer1,integer2)]
[symbol]	HOLD	[mask] [,RELEASE] [,INTVL=integer] [,CONNECT={AND / OR}] [,LEVEL={BLK / MSG}]
[symbol]	ICHNG	grpname,rln,{areaname / ACT / DEACT} [,PSSWRD=chars]
[symbol]	ICOPY	grpname,rln,areaname
[symbol]	IEDHALT (outmessage or inmessage subgroup)	mask,CONNECT={AND / OR}
[symbol]	IEDHALT (Inheader subgroup)	CHAR={chars / opfield}
[symbol]	IEDOPCTL	(no operands)
[symbol]	INBLOCK	[PATH=(opfield,switch)]
[symbol]	INBUF	[PATH=(opfield,switch)]
[symbol]	INEND	(no operands)
[symbol]	INHDR	[PATH=(opfield,switch)]
[symbol]	INITIATE	[conchars [,BLANK={char / NO / YES}]]
[symbol]	INMSG	[PATH=(opfield,switch)]
[symbol]	INTRO	KEYLEN=integer UNITSZ=integer ,LNUNITS=integer ,STARTUP={C / W} [{Y / E}] [(I)] [,CIB={integer / 2}] [,CKREQS={integer / 0}] [,COMMBUF=(integer1,integer2,integer3)] [,COMWRTE={YES / NO}] [,CONTROL={characters / 0}] [,CPB={integer / 0}] [,CPINTVL={integer / 1800}] [,CPRCDS={integer / 2}]

Name	Operation	Operands
		[,CROSSRF={integer / 0}] [,DISK={YES / NO}] [,DLQ={entry / 0}] [,DTRACE={integer / 0}] [,FEATURE=({NODIAL / DIAL},{NO2741 / 2741},{NOTIMER / TIMER},{NOCONC / CONCO / CONC},{MIXDVTAM / ONLYVTAM / NOVTAM})] [,INTVAL={integer / 0}] [,LINETYPE={BISC [STSP | MINI | BOTH]}] [,MSMAX={integer / 70}] [,MSMIN={integer / 50}] [,MSUNITS={integer / 0}] [,OLTEST={integer / 18}] [,PASSWRD=characters] [,PLCBNO=integer] [,PRIMARY={termname / SYSCON}] [,PROGID=characters] [,RESTART={integer / 0}] [,THRESH=({integer1 / 255},{integer2 / 1},{integer3 / 1},{integer4 / 1})] [,TOPMSG={YES / NO}] [,TRACE={integer / 0}] [,USEREG={integer / 0}] [,VM={PROMPT / YES / NO}] [,WTTONE={integer / 0}]
symbol	INVLIST	ORDER=(entry,...)[,EOT=hexchars][,CPUID=addr][,MASTER={YES / NO}]
[symbol]	LOCK	{EXTEND / MESSAGE} [,conchars [,BLANK={YES / NO / char}]]
[symbol]	LOCOPT	opfield,{(register) / (15)}
[symbol]	LOG	{dcbname / typename}

TCAM Macros (cont'd)

Name	Operation	Operands
typename	LOGTYPE	dcbname,BUFSIZE=size,QUEUES=form
[symbol]	MCOUNT	DCB= { name / (r) }
[symbol]	MCPCLOSE	{ QUICK / FLUSH } [,PASSWRD=chars]
[symbol]	MHGET	{ WORK= { (register) / name } [,RESERVE= { YES / NO }] / REG=(register) }
[symbol]	MHPUT	WORK= { (register) / name } [,RESERVE=integer]
[symbol]	MRCHECK	(no operands)
[symbol]	MRELEASE	statname [,PASSWRD=chars]
[symbol]	MSGEDIT	((group1),(group2),...),BLANK= { NO / char / YES }
[symbol]	MSGFORM	[BLOCK=integer] [,SUBBLCK=integer] [,COUNT=integer] [,SENDTRP= { YES,PAD / YES,NOPAD / NO }] [,ENDCHAR=subblock delimiter] [,LC= { IN / OUT }]
[symbol]	MSGGEN	[mask],{ message / fieldname } [,CONNECT= { AND / OR }] [,CODE= { tablename / NO }]
[symbol]	MSGLIMIT	{ integer / opfield }
[symbol]	MSGTYPE	[{ conchars / TABLE=name,EXIT=name }] [,BLANK= { YES / NO / char }]
[symbol]	OPEN	(dcbname,,...) [,MF= { L / (E,listname) }]
[symbol]	OPEN (MCP)	(dcbname, [{ OUTPUT / INOUT / INPUT } [,IDLE]],...) [{ MF=L / MF=(E,listname) }]
opfldname	OPTION	typelength
[symbol]	ORIGIN	[integer / X'FF'] [,FORM= { ID / NAME }]
[symbol]	OUTBUF	[PATH=(opfield,switch)]

Name	Operation	Operands
[symbol]	OUTEND	(no operands)
[symbol]	OUTHDR	[PATH=(opfield,switch)]
[symbol]	OUTMSG	[PATH=(opfield,switch)]
[symbol]	PATH	switch,opfield[,conchars[,BLANK={YES / NO / char}]]
pcbname	PCB	MH=mhname,BUFSIZE=integer [,BUFIN={number / 2}] [,BUFOUT={number / 2}] [,RESERVE=(integer1,integer2)] [,SFLAG={YES / NO}] [,DATE={YES / NO}]
[symbol]	POINT	dcbname,address
[symbol]	PRIORITY	[integer] [,conchars[,BLANK={YES / NO / char}]]
[symbol]	PUT	dcbname[,areaname]
[symbol]	QACTION	TYPE={A / V},EXIT=name
[symbol]	QCOPY	termname,areaname[,LIMIT={integer / (register)}]
[symbol]	QRESET	dcbname,MAX=integer
[symbol]	QSTART	(no operands)
[symbol]	READ	decbname,SF,dcbname,areaname, {length / 'S'} [MF={L / (E,listname)}]
[symbol]	READY	[GMMSG=routine] [,RSMSG=routine]
[symbol]	REDIRECT	[mask][,CONNECT={AND / OR}] [,DEST={destname / opfield / ORIGIN}]
[symbol]	RETRY	INTVL=integer
[symbol]	SCREEN	{WRE / WLA / WDC / XRE / XLA / XDC} [conchars[,BLANK={YES / NO / char}]]
[symbol]	SEQUENCE	(no operands)

TCAM Macros (cont'd)

Name	Operation	Operands
[symbol]	SETEOF	[conchars [,BLANK= {YES / NO / char}]]
[symbol]	SETEOM	[ENDCHAR= {chars / opfield}] [EOM=ETB] [,LENGTH=({integer / opfield1} ,opfield2)] [,PROCESS= {YES / NO}] [,REMOVE= {YES / NO}]
[symbol]	SETSCAN	{skipchars / integer} [,BLANK= {YES / NO / char}] [,POINT= {BACK / FORWARD}] [,MOVE= {RETURN / KEEP}] [,RESULT= {(register) / (15)}]
[symbol]	SLOWPOLL	[mask] [,CONNECT= {AND / OR}] [,SECONDS= {integer} / {60}]
mhname	STARTMH	LC= {IN / OUT}[,{STOP / CONT}] = {YES / (opfield,switch)} [,CONV= {YES / (opfield,switch) / NO}] [,LOGICAL= {(opfield) / (opfield1,switch,opfield2)}] [,BREG= {integer / 1}] [,LMD= {YES / NO / (opfield,switch)}]
[symbol]	TCHNG	termname,areaname [,PASSWRD=chars]
[symbol]	TCOPY	statname,areaname
[symbol]	TERMINAL	QBY= {T / L}[,DCB=dcbname ,GROUP=grpname] ,RLN=integer,TERM=type ,QUEUES=form[,DIALNO= {REMOTE / chars / NONE}] [,ADDR=chars] [,LEVEL=(integer,...)] [,CLOCK=time] [,CINTVL=integer] [,BUFSIZE=integer] [,ALTDEST=entry] [,BFDELAY=integer] [,NTBLKSZ=(blocksize,subblocksize)] [,TBLKSZ=integer] [,OPDATA=(data,...)] [,RETRY=integer] [,LMD= {YES / NO}] [,MB= {YES / NO}] [,SECTERM= {YES / NO}] [,FEATURE= {ATTN / NOATTN}] [,COMP= {YES / NO}] [,UTERM= {YES / IDLE / NO}] [,SPECOUT= {YES / NO}] [TERMID=n]

TCAM Macros(cont'd)

Name	Operation	Operands
[symbol]	TERMINAL (continued)	[,DVCID={(CONC[,integer])} {chars} {NONE}] [,QCNTRL={(ALL / (MSG[,{msgcount}]) / bytecount [,L][,STATUS][,char])}] [,CTBMAX=integer]
[symbol]	TERRSET	(no operands)
[symbol]	TGOTO	MH={name of MH / opfield}
[symbol]	TLIST	TYPE={D / C},LIST=(entry,entry,...)
[symbol]	TPDATE	DCB={name / (r)}[,RECDLM={YES / NO}] [,DTSAREA={area / (r)}][,DELETE={YES / NO}]
[name]	TPEDIT	MINLN=n,EDIT={EDITR / EDITD},RECFM={U / V}, ERROPT={name / IGNORE},VERCHK={VOKCHK / NOCHK}, REPLACE={X'xx' / X'19'},BUFFER={YES / NO}
procname	TPROCESS	PCB=pcbname[,QUEUES=form][,ALTDEST=entry] [,CKPTSYN={YES / NO}][,DATE={YES / NO}] [,SECTERM={YES / NO}][,RECDEL=delimiter] [,LEVEL=(integer,...)][,OPDATA=(data,...)] [,QBACK={YES / NO}] [,SECURE={YES / NO}]
[symbol]	TTABLE	LAST=name[,MAXLEN=integer][,OLTERM=n]
[symbol]	TYPETABL	conchar,ROUTINE=name
[symbol]	UNLOCK (Outheader or inheader subgroup)	[conchars [,BLANK={YES / NO / char}]]
[symbol]	UNLOCK (Outmessage or inmessage subgroup)	[mask] [,CONNECT={OR / AND}] [,DISABLE={NO / YES}]
[symbol]	WRITE	decbname,SF,dcbname,areaname,{length / '5'}

TCAM Operator Commands

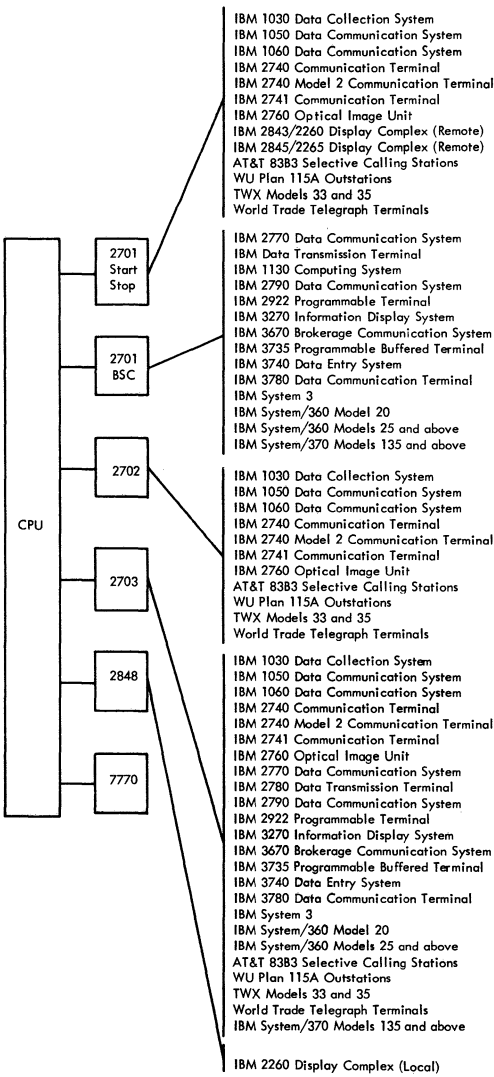
Control Chars	Operation	Operands
control chars	{DISPLAY} {D}	TP,ACT,{grpname,rln} {address} TP,ADDR,statname TP,INACT,{grpname,rln} {address} TP,INTER TP,LINE,{grpname,rln} {address} TP,LINE,INACTIVE TP,LIST,{grpname,rln} {address} TP,OPTION,statname,opfldname,{X} {C} {D} TP,PRITERM TP,QUEUE,statname TP,SECTERM TP,STATUS,{ddname} {address} TP,STORE,{ddname} {address},aaaaa TP,TERM,statname
control chars	{HALT} {Z}	TP,{QUICK} {FLUSH}
control chars	{HOLD} {H}	TP=statname
control chars	{MODIFY} {F}	procname.id :id jobnama procname ,ACTIV={ddname}[,IDLE] {address} ,AUTOPOLL={grpname,rln},{ON} {address}{OFF} ,BACKUP={ddname} {address} ,BHSET=statname,{C}[,aaa] {A} {D} ,BTRACE={linename},{ON} {grpname,rln}{OFF} ,CHANL={ddname} {address} ,CHNGMODE={grpname,rln}[,AUTO] {grpname[,ALL]}[,MANUAL]

Control Chars	Operation	Operands
control chars	{MODIFY} {F} (cont'd)	procname.id id jobname procname ,DEACT= {ddname} ,[QUICK] {address} [FLUSH] ,DEBUG= {L} , {IEDQFE10} {D} {IEDQFE20} {IEDQFE30} ,DUMP= {ddname} {address} ,INTENSE= {LINE, {grpname,rln} } ,sense,[count] {address} [15] {TERM,statname} ,INTERVAL=POLL,statname,data, {S} {N} ,INTERVAL=SYSTEM[,data] ,IPL= {ddname} {address} ,LNSWITCH=grpname,rln, {EP} {NCP} ,LOAD= {ddname}.txt {address} ,OPERATOR= {statname} {SYSCON} ,OPT=statname,opfldname,data ,SESSION=grpname,rln,aaa ,SPEED=grpname,rln, {H} {L} ,SWAP=concname,statname1,statname2 ,SWBACK= {ddname1}, {ddname2} {address1} {address2} ,SWDEVICE=statname,[P] [B[grpname,rln]] ,SWITCH= {ddname} {address} ,TIMEDAT= {ddname} {address} ,TRACE= {grpname,rln}, {ON} ,aaa,bbbbb {address} {OFF} {addr3705/addrline} ,TRANLMT=statname,aaa

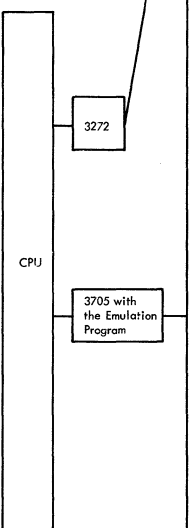
TCAM Operator Commands(cont'd)

Control Chars	Operation	Operands
control chars	{ RELEASE { A }	TP=statname
control chars	{ VARY } { V }	{ gpstatname, ONTP, { E / B } statname, ONTP, B statname, ONTP, E gpstatname, OFFTP, { E / B } statname, OFFTP, { B / BM } statname, OFFTP, { E / EM } { (grpname, rln) } , OFFTP, { C / I } { (grpname,) } { address } { (grpname, ALL) } { (grpname, rln) } , ONTP { (grpname,) } { address } { (grpname, ALL) } }

Device Configurations Supported by TCAM



Device Configurations Supported by TCAM (cont'd)



IBM 3270 Information Display System (Local)

Start-Stop Terminals

IBM 1030 Data Collection System
IBM 1050 Data Communication System
IBM 1060 Data Communication System
IBM 2740 Communications Terminal, Models 1 and 2
IBM 2741 Communications Terminal
IBM 2760 Optical Image Unit (via the IBM 2740 Communications Terminal, Model 1)
IBM 2845/2265 Display Complex
IBM 2848/2260 Display Complex
IBM System 7 (supported as an IBM 2740 Communications Terminal, Model 1, with the Record Check feature)

Binary Synchronous Terminals

IBM 2270 Data Communications System
IBM 2780 Data Transmission Terminal (no support for Transcode)
IBM 2922 Programmable Terminal
IBM 3270 Information Display System
IBM 3670 Brokerage Communication System (supported only in the United States and Canada)
IBM 3735 Programmable Buffered Terminal
IBM 3780 Data Communication Terminal

Transmission Control Units (BSC support only)

IBM 2701 Data Adapter Unit (with Synchronous Data Adapter Type II)
IBM 2703 Transmission Control (with Synchronous Terminal Control)
IBM 2715 Transmission Control, Mode! 2
IBM 3705 Communications Controller

Note: The IBM 2701, IBM 2703, and IBM 3705 control units are supported only when attached locally to an IBM System/360 or System/370.

Computers (BSC support only)

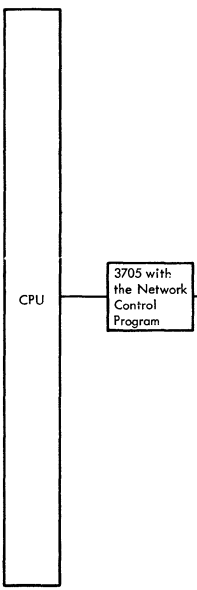
IBM 1130 Computing System (with Synchronous Communications Adapter)
IBM System 3
IBM System/360, Model 20 (with Binary Synchronous Communications Adapter)
IBM System/360, Model 25 and above (with Integrated Communications Attachment with Synchronous Data Adapter II)
IBM System/370, Model 135 and above (with Integrated Communications Attachment with Synchronous Data Adapter II)

Other Devices

- World Trade teleprinters that use CCITT=2 or =5 code leased point-to-point, leased multipoint, or switched network lines.
- Terminals using the following line control disciplines: AT&T 83B3 or WU 115A start-stop code, over point-to-point or multipoint leased telegraph lines: AT&T CPT-TWX (33/35) start-stop code over switched lines.

Attachment of non-IBM device is under the provisions of the IBM Multiple Supplier Systems Policy.

Device Configurations Supported by TCAM (cont'd)



Start-Stop Terminals

IBM 1050 Data Communication System
IBM 2740 Communications Terminal, Models 1 and 2
IBM 2741 Communications Terminal
IBM System/7 (supported as an IBM 2740 Communications Terminal Model 1, with the Record Check feature)

Binary Synchronous Terminals

IBM 2770 Data Communication System
IBM 2780 Data Transmission Terminal (no support for Transcode)
IBM 2972 General Banking Terminal System, Models 8 and 11)
IBM 3270 Information Display System
IBM 3735 Programmable Buffered Terminal
IBM 3780 Data Communication Terminal

Transmission Control Units (BSC support only)

IBM 2701 Data Adapter Unit (with Synchronous Data Adapter Type II)
IBM 2703 Transmission Control (with Synchronous Terminal Control)
IBM 2715 Transmission Control Model 2
IBM 3705 Communications Controller

Note: The IBM 2701, IBM 2703, and IBM 3705 control units are supported only when attached locally to an IBM System/360 or System/370.

Computers (BSC support only)

IBM System/3
IBM System/360, Model 20 (with Binary Synchronous Communications Adapter)
IBM System/360, Model 25 and above (with Integrated Communications Attachment with Synchronous Data Adapter)
IBM System/370, Model 135 and above (with Integrated Communications Attachment with Synchronous Data Adapter II)
IBM 1130 Computing System (with Synchronous Communications Adapter)
IBM 1800 Data Acquisition and Control System (with IBM 1826 Data Adapter Unit with Communication Adapter)

Other Devices

- World Trade teleprinters that use CCITT No. 2 or No. 5 code on leased point-to-point lines.
- Terminals using the following line control disciplines: AT&T 83B3 or WU 115A start-stop code, over point-to-point or multipoint leased telegraph lines: AT&T CPT-TWX (33/35) start-stop code over switched lines.

Attachment of non-IBM terminals is under the provisions of the IBM Multiple Supplier Systems Policy.

Device Configurations Supported by TCAM

Station Type		Channel Type		TCU					Audio Response Unit	Line Type		Notes
		Multi-plexer	Selector	IBM 2701 Data Adapter Unit	IBM 2702 Trans-mission Control	IBM 2703 Trans-mission Control	IBM 3705 EP Comm Con-troller	IBM 3705 NCP Comm Con-troller	IBM 7770 Model 3	Switched	Non-Switched	
IBM 1030 Data Collection System	Auto Poll	X			X	X	X				X	The IBM Digital Time Out feature cannot be attached through an IBM 2701 TCU.
		X		X	X	X	X				X	
IBM 1050 Data Communication System	Auto Poll	X			X	X	X				X	
		X		X	X	X	X	X		X	X	
IBM 1060 Data Communication System	Auto Poll	X			X	X	X				X	
		X		X	X	X	X				X	
IBM 2260-2848 Display Complex (Remote)		X		X			X				X	
IBM 2260-2848 Display Complex (Local)		X	X									
IBM 2265-2845 Display Complex (Remote)		X		X			X				X	
IBM 2740 Model 1 Communication Terminal	Auto Poll	X			X	X	X				X	Two Types: 2740 with station control 2740 with station control and record checking
		X		X	X	X	X	X			X	Four Types: 2740 basic 2740 with station control 2740 with record checking 2740 with station control and record checking
		X		X	X	X	X			X		Four Types, all with dial: 2740 2740 with transmit control 2740 with record checking 2740 with transmit control and record checking
IBM 2740 Model 2 Communication Terminal	Auto Poll	X			X	X	X	X			X	Four Types: 2740 2740 with record checking 2740 with buffer receive 2740 without buffer receive (requires lines slowdown feature)
		X		X	X	X	X				X	Four Types: 2740 2740 with record checking 2740 with buffer receive 2740 without buffer receive

Device Configurations Supported by TCAM(cont'd)

Station Type	Channel Type		TCU					Audio Response Unit	Line Type		Notes
	Multi-plexer	Selector	IBM 2701 Data Adapter Unit	IBM 2702 Trans-mission Control	IBM 2703 Trans-mission Control	IBM 3705 EP Comm Con-troller	IBM 3705 NCP Comm Con-troller	IBM 7770 Model 3	Switched	Non-Switched	
IBM 2741 Communication Terminal	X		X	X	X	X	X		X	X	
IBM 1130 Computing System	X		X		X	X	X		X	X	BSC transmission
IBM 1800 Data Acquisition System							X				
IBM 2760 Optical Image Unit									X	X	Attached to a 2740 Model 1 with record checking
IBM 2770 Data Communication System	X		X		X	X	X		X	X	BSC transmission using either ASCII or EBCDIC code
IBM 2780 Data Transmission Terminal	X		X		X	X	X		X	X	BSC transmission ASCII, EBCDIC, or 6-bit code
IBM 2790 Data Communications System	X		X		X	X	X		X	X	
IBM 2972 General Banking Terminal							X				
IBM 3270 Information Display System	X		X		X	X	X			X	
IBM 3670 Brokerage Communication System	X		X		X	X				X	BSC transmission using EBCDIC code
IBM 3735 Program-mable Buffered Terminal	X		X		X	X	X		X	X	Either ASCII or EBCDIC
IBM 3740 Data Entry System	X		X		X	X			X	X	BSC Transmission code TERM= BSC1 or TERM=BSC2 on TERMINAL macro
IBM 3780 Data Communication Terminal	X		X		X	X	X		X	X	BSC transmission using either ASCII or EBCDIC code
IBM System 3	X		X		X	X	X		X	X	Code TERM=202A or TERM=202B on TERMINAL. Macro inquiry/ response not supported
IBM System 7							X				
IBM System/360 Model 20	X		X		X	X	X		X	X	BSC transmission using either ASCII or EBCDIC code

Station Type	Channel Type: Multiplexer	Channel Type: Selector	TCU: IBM 2701 Data Adapter Unit	TCU: IBM 2702 Transmission Control	TCU: IBM 2703 Transmission Control	TCU: IBM 3705 EP Communication Controller	TCU: IBM 3705 NCP Communication Controller	Audio Response Unit: IBM 7770 Model 3	Line Type: Switched	Line Type: Non-Switched	Notes
IBM System/360 Model 25 and above	X		X		X	X	X		X	X	BSC transmission and point-to-point lines only
IBM System/370 Model 135 and above	X		X		X	X	X		X	X	BSC transmission and point-to-point lines only
IBM System/370 Model 135 Integrated Communication Attachment							X				
AT & T 83B3 Selective Calling Station	X		X	X	X	X	X			X	
Western Union Plan 115A Outstations	X		X	X	X	X	X			X	
TWX Models 33 and 35	X		X	X	X	X	X		X		Teletype terminals, dial service (8 level code)
World Trade Telegraph Terminals	X		X	X	X	X	X			X	Control unit must incorporate a WTTA
Audio terminals	X							X	X		Example IBM 2721 Portable Audio Terminal

VTAM Macros

ACB -- Create an Access Method Control Block

Name	*Operation*	*Operands*
[symbol]	ACB	AM=VTAM [,APPLID=address of application's symbolic name] [,PASSWD=password address] [,EXLST=exit list address] [,MACRF={<u>LOGON</u>|NLOGON}]

CHANGE -- Change a NIB's PROC Option or USERFLD Data

[symbol]	CHANGE	RPL=rpl address [,rpl field name=new value] ...

CHECK -- Check Request Status

[symbol]	CHECK	RPL=rpl address

CLOSE -- Close One or More ACBs

[symbol]	CLOSE	(acb address[,,acb address] ...)

CLSDST -- Disconnect a Terminal from the Application Program

[symbol]	CLSDST	RPL=rpl address [,rpl field name=new value] ...

DO -- Initiate LDO-specified I/O Operations

[symbol]	DO	RPL=rpl address [,rpl field name=new value] ...

EXECRPL -- Execute a Request

[symbol]	EXECRPL	RPL=rpl address [,rpl field name=new value] . . .

EXLST -- Create an Exit List

[symbol]	EXLST	AM=VTAM [,DFASY= / [,RESP= / [,SCIP= / [,LERAD= } exit routine address] [,SYNAD= / [,TPEND= / [,RELREQ= / [,LOGON= / [,LOSTERM= / [,ATTN= } exit routine address]

GENCB -- Generate a Control Block

[symbol]	GENCB	BLK={ACB|EXLST|RPL|NIB} [,keyword=value] ... [,COPIES={<u>1</u>|quantity}] [,WAREA=work area address ,LENGTH=work area length] [,MF={L|({ {L|G},{address|(R)}[,label] | E,parameter list address })}]

INQUIRE -- Obtain Terminal Information or Application Program Status

Name	*Operation*	*Operands*
[symbol]	INQUIRE	RPL=rpl address [,rpl field name=new value]...

INTRPRET -- Interpret a Logon Message

[symbol]	INTRPRET	RPL=rpl address [,rpl field name=new value]...

LDO -- Create a Logical Device Order

[symbol]	LDO	CMD-command [,ADDR=data address or data area address] [,LEN=data length or data area length] [,FLAGS=C|D]

MODCB -- Modify the Contents of Control Block Fields

[symbol]	MODCB	AM=VTAM {,ACB=acb address ,EXLST=exit list address ,RPL=rpl address ,NIB=nib address} {,field name=new value}... [,MF={L|(E|{L|G},address[,label])}]

NIB -- Create a Node Initialization Block

[symbol]	NIB	[NAME=name in resource definition table] [,USERFLD=fullword of terminal data] [,SDT=APPL|SYSTEM] [,EXLST=exit list address] [,RESPLIM=response limit] [,LISTEND={YES|NO}] [,MODE=BASIC] ,PROC=[,{BLOCK|MSG|TRANS|CONT}] [,{LGOUT|NLGOUT}] [,{LGIN|NLGIN}] [,{DFASYX|NDFASYX}] [,{RESPX|NRESPX}] [,{CONFTXT|NCONFIXT}] [,{TMFLL|NTMFLL}] [,{EIB|NEIB}] [,{TIMEOUT|NTIMEOUT}] [,{ERPIN|NERPIN}] [,{ERPOUT|NERPOUT}] [,{MONITOR|NMONITOR}] [,{ASYIPX|NASYIPX}] [,{ELC|NELC}] [,{TRUNC|KEEP}] [,{BINARY|NBINARY}]

OPEN -- Open one or more ACBs

[symbol]	OPEN	(acb address[,,acb address]...)

OPNDST -- Establish Connection with Terminals

Name	*Operation*	*Operands*
[symbol]	OPNDST	RPL=rpl address [,rpl field name=new value] ...

READ -- Read Data into Program Storage

[symbol]	READ	RPL=rpl address [,rpl field name=new value] ...

RECEIVE -- Receive Input from a Logical Unit

[symbol]	RECEIVE	RPL=rpl address [,rpl field name=new value] ...

RESET -- Cancel an I/O Operation

[symbol]	RESET	RPL=rpl address [,rpl field name=new value] ...

RESETSR -- Cancel an I/O Operation or Switch CS-CA MODE

[symbol]	RESETSR	RPL=rpl address [,rpl field name=new value] ...

RPL -- Create a Request Parameter List

Name	*Operation*	*Operands*
[symbol]	RPL	AM=VTAM [,ACB=acb address] [,NIB=nib address] [,AREA=data area address] [,AREALEN=data area length] [,RECLEN=data length] [,AAREA=alternate data area address] [,AAREALN=alternate data area length] [{,ECB=event control block address / ,EXIT=rpl exit-routine address}] [,SEQNO=sequence number] [,POST=SCHED|RESP] [,RESPOND=(EX|NEX,FME|NFME,RRN|NRRN)] [,CONTROL={DATA|QEC|RELQ|QC|CANCEL|CHASE|SHUTD|BID|LUS|SDT|CLEAR|STSN}] [,CHAIN=FIRST|MIDDLE|LAST|ONLY] [,CHNGDIR=(CMD|NCMD,REQ|NREQ)] [,BRACKET=(BB|NBB,EB|NEB)] [,RTYPE=(DFSYN|NDFSYN,DFASY|NDFASY,RESP|NRESP)] [,STYPE=REQ|RESP] [,SSENSEQ=CPM|STATE|FI|RR] [,SSENSMO=system sense modifier value] [,USENSEQ=user sense value] [,IBSQAC=SET|TESTSET|INVALID|IGNORE] [,OBSQAC=SET|TESTSET|INVALID|IGNORE] [,IBSQVAL=inbound sequence number] [,OBSQVAL=outbound sequence number] [,OPTCD=[,{CONALL / CONANY}] [,NIBTK|TRUNK|KEEP] [,FMHDR|NFMHDR] [,{ACCEPT|ACQUIRE}] [,{SPEC|ANY}] [,{QUIESCE|STOP|START}] [,{PASS|RELEASE}] [,{LOGONMSG|DEVCHAR|COUNTS|TERMS|APPSTAT|CIDXLATE|TOPLOGON|BSCID}] [,{SYN|ASY}] [,{CS|CA}] [,{BLK|LBM|LBT}] [,{CONV|NCONV}] [,{COND|UNCOND|LOCK}] [,{ERASE|EAU|NERASE}] [,{RELRQ|NRELRQ}] [,{Q|NQ}]]

VTAM Macros (cont'd)

SEND -- Send Output to a Logical Unit

Name	*Operation*	*Operands*
[symbol]	SEND	RPL=rpl address [,rpl field name=new value] ...

SESSIONC -- Send an STD, Clear, or STSN Indicator to a Logical Unit

[symbol]	SESSIONC	RPL=rpl address [,rpl field name=new value] ...

SETLOGON -- Reset an ACB's Logon Status

[symbol]	SETLOGON	RPL=rpl address [,rpl field name=new value] ...

SHOWCB -- Extract the Contents of Control Block Fields

[symbol]	SHOWCB	AM=VTAM {,ACB=acb address ,EXLST=exit list address ,RPL=rpl address ,NIB=nib address} ,FIELDS={field name | (field name,...)} ,AREA=data area address ,LENGTH=data area length [,MF={L | ({{L},{address} [,label] {G} {(R)} E,parameter list address})}]

SIMLOGON -- Generate a Simulated Logon Request

[symbol]	SIMLOGON	RPL=rpl address [,rpl field name=new value] ...

SOLICIT -- Obtain Data from a Terminal

[symbol]	SOLICIT	RPL=rpl address [,rpl field name=new value] ...

TESTCB -- Test the Contents of a Control Block Field

[symbol]	TESTCB	AM=VTAM {,ACB=acb address ,EXLST=exit list address ,RPL=rpl address ,NIB=nib address} ,field name=test value [,ERET=error exit routine address] [,MF={L | ({{L},{address} [,label] {G} {(R)} E,parameter list address})}]

WRITE -- Write a Block of Data from Program Storage to a Terminal

[symbol]	WRITE	RPL=rpl address [,rpl field name=new value] ...

Devices Supported by VTAM; Processing Options

Start–Stop Devices:	PROC operand value																	
	BLOCK	MSG	TRANS	CONT	LGOUT	CONFTXT	NTMFLL	EIB	NTIMEOUT	NERPIN	NERPOUT	MONITOR	ASYIPX	ELC	TRUNC	CA-CS-RPLC
IBM 1050 Data Communication System	X		X	X		X	X		X	X	X	X	X	X	X	X
IBM 2740 Communication Terminal, Model 1			X	X		X	X		X				X	X	X	X
IBM 2740 Communication Terminal, Model 1 with checking	X		X	X		X	X		X	X	X		X	X	X	X
IBM 2740 Communication Terminal, Model 1, with station control			X	X		X	X		X				X	X	X	X
IBM 2740 Communication Terminal, Model 1, with checking and station control	X		X	X		X	X		X	X	X		X	X	X	X
IBM 2740 Communication Terminal, Model 2			X	X		X				X	X		X		X	X
IBM 2741 Communication Terminal			X	X		X	X		X			X	X	X	X	X
IBM Communicating Magnetic Card Selectric Typewriter			X	X		X	X		X	X	X	X	X	X	X	X
IBM World Trade Telegraph Station			X	X		X							X	X	X	X
IBM SYSTEM/7			X	X		X			X				X	X	X	X
AT and T 8383 Selective Calling Station			X	X		X							X	X	X	X
CPT–TWX Models 33 and 35			X	X		X	X		X				X	X	X	X
Western Union Plan 115A Station			X	X		X							X	X	X	X
Binary Synchronous Devices:																
IBM 2770 Data Communication System	X	X	X	X		X		X		X	X		X	X	X	X
IBM 2780 Data Transmission Terminal	X	X	X	X		X		X		X	X		X	X	X	X
IBM 2972 General Banking Terminal, Models 8 and 11			X	X		X		X		X	X		X	X	X	X
IBM 3270 Information Display System, locally attached to controller			X			X							X		X	X
IBM 3270 Information Display System, remotely attached to controller			X			X							X		X	X
IBM 3735 Programmable Buffered Terminal	X	X	X	X	X	X		X		X	X		X	X	X	X
IBM 3740 Data Entry System	X	X	X	X		X		X		X	X		X	X	X	X
IBM 3780 Data Transmission Terminal	X	X	X	X		X		X		X	X		X	X	X	X
IBM SYSTEM/3	X	X	X	X	X	X		X		X	X		X	X	X	X
IBM SYSTEM/370	X	X	X	X	X	X		X		X	X		X	X	X	X
SNA Devices:																
IBM 3600 Finance Communication System						X									X	X
IBM 3650 Retail Store System						X									X	X
IBM 3660 Supermarket System						X									X	X
IBM 3767 Communications Terminal						X									X	X
IBM 3770 Data Communications System						X									X	X
IBM 3790 Communication System						X									X	X
IBM 3270 SNA Information Display System						X									X	X
IBM System 32 Batch Work Station						X									X	X

Terminals Operating in a VTAM Network

Summary of Terminals That Can Operate in a VTAM Network (GC27-6998)

Device	Type of Network	Comments
Local		
3270		3277, 3284, 3286 require 3272 Control Unit
SDLC		SDLC terminals are logical units
3601	b, c	3600 Finance Communications System
3651 Models 50, 60	b, c	3650, 3660 Retail Store System
3767	a,b,c	3767 Communications Network
3770	a,b,c	3770 Data Communications System
3791	b, c	3790 Communications System
Start-Stop		
1050	a, c	105x, 109x Terminals
2740 Model 1	a, b, c	
2740 Model 2	b, c	
2741	a, b	
AT&T 33/35 TWX	a	
AT&T 83B3	c	
WU 115A	c	
Communicating Mag Card SELECTRIC Typewriter	a	
System/7	a, b, c	
World Trade Telegraph	b	
BSC		
2770	a, b, c	Uses 2772 Control Unit
2780	a, b, c	
2972 Models 8, 11	c	Terminals are 2980, 2971
3270	c	Terminals 3277, 3284, 3286 require 3271/3275 Control Unit
3735	a, c	5496, 3286
3740	a, b, c	3741, 3747
3780	a, b, c	
System/3	a, b, c	
System/7	a, b, c	
System/370	a, b	2701, 2703, 3704, 3705; ICA

Network Types
a - Point-to-Point Switched
b - Point-to-Point Nonswitched
c - Multipoint Nonswitched

Command **Operand Field**

{DISPLAY / D} NET, ID= { { *application program name* / *cluster name* / *line name* / *ncp name* } [,EVERY\|E / ,ACT\|A / ,INACT\|I / ,NONE\|N] / { *terminal name* / *physical unit name*, {PATH\|P} } }

{HALT / Z} NET [,QUICK]

{MODIFY / F} *[procname.] identifier* {
- DUMP, ID=*ncp name* [,RMPO]
- NEGPOLL=*number*, ID=*line name*
- NETSOL= {YES\|NO}
- POLL=*number*, ID=*line name*
- SESSION=*number*, ID=*line name*
- SUPP= {NOSUP\|INFO\|WARN\|NORM\|SER}
- TEST
- {TRACE / NOTRACE} ,ID= { { *cluster name* / *component name* / *ncp name* / *terminal name* } ,TYPE= {IO / BUF} / *line name*, TYPE=LINE / VTAMBUF, TYPE=SMS }
- TRANLIM=*number*, ID=*terminal name*
}

{START / S} *procname[.identifier] [, , , (parameters , . . .)]*

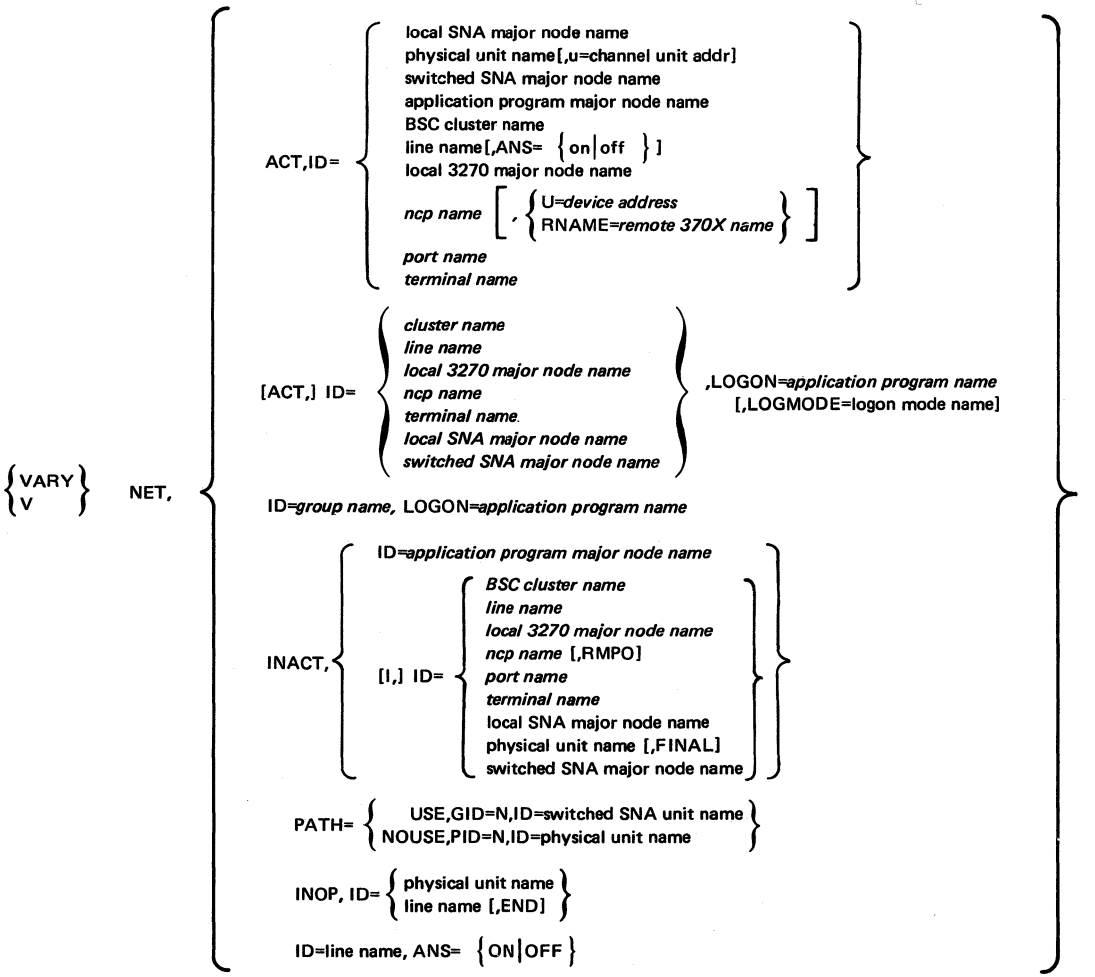
The *parameters* are:

CONFIG=*id*
LIST=*id*
MAXSUBA=*number*
NETSOL= {YES\|NO}
SUPP= {NOSUP\|INFO\|WARN\|NORM\|SER}
SSCPID=number
{TRACE / NOTRACE} ,TYPE= { {IO / BUF} ,ID= { *cluster name* / *component name* / *ncp name* / *terminal name* } / LINE, ID=*line name* / SMS,ID=VTAMBUF }

APBUF=*(bno, bsz, bth)* *
CRPLBUF=*(bno, bsz, bth)*
IOBUF=*(bno, bsz, bth)*
LFBUF=*(bno, bsz, bth)*
LPBUF=*(bno, bsz, bth)*
NPBUF=*(bno, bsz, bth)*
PPBUF=*(bno, bsz, bth)*
SFBUF=*(bno, bsz, bth)*
SPBUF=*(bno, bsz, bth)*
UECBUF=*(bno, bsz, bth)*
WPBUF=*(bno, bsz, bth)*

* bno= number of buffers in a pool
bsz = size in bytes of each buffer
bth = threshold value for a buffer type

Command Operand Field



Section 8: Utilities and Service Aids

The information in this section is concerned with a variety of utilities and service aids used for data-base maintenance and system modification.

Source Publications

Additional information is in:

- *OS/VS Utilities,* GC35-0005
- *OS/VS1 Service Aids,* GC28-0665
- *OS/VS1 SYS1.LOGREC Error Recording,* GC28-0668
- *OS/VS System Modification Program (SMP),* GC28-0673

For information on Access Method Services, a group of VSAM utilities, see *OS/VS1 Access Method Services,* GC26-3840.

For information on Mass Storage System service aids, see *OS/VS Mass Storage System (MSS) Services for Space Management,* GC35-0012.

For information on Analysis Program-1 (AP-1), see *OS/VS and DOS/VS Analysis Program-1 (AP-1) User's Guide,* GC26-3855.

Utility Programs - listed by class

SYSTEM Utility Programs	DATA SET Utility Programs	INDEPENDENT Utility Programs
IEHATLAS	IEBCOMPR	IBCDASDI
IEHDASDR	IEBCOPY	IBCDMPRS
IEHINITT	IEBDG	ICAPRTBL
IEHIOSUP	IEBEDIT	
IEHLIST	IEBGENER	
IEHMOVE	IEBISAM	
IEHPROGM	IEBPTPCH	
IFHSTATR	IEBTCRIN	
	IEBUPDTE	

The utilities section is arranged in alphabetical order for easy reference.

The control statement for the utility programs have the following standard format:

label	operation	operand

The label symbolically identifies the control statement. When included, a label must begin in the first position of the statement and must be followed by one or more blanks. It can contain from one to eight alphameric characters, the first of which must be alphabetic.

The operation identifies the type of control statement. It must be preceded and followed by one or more blanks.

The operand is made up of one or more keyword parameters separated by commas. The operand field must be preceded and followed by one or more blanks. Commas, parentheses, and blanks can be used only as delimiting characters.

A definition of operands table is located at the back of this section. It should be used, when needed, as a recall mechanism; it is not intended for use as tutorial information. If you require additional information, refer to the source publication listed for this section.

Guide to Utility Program Functions

Task		Utility Program
Add	a password	IEHPROGM
Analyze	tracks on direct access	IEHATLAS, IEHDASDR, IBCDASDI
Assign alternate tracks	to a direct access volume	IEHATLAS, IEHDASDR, IBCDASDI
Build	a generation index	IEHPROGM
	a generation	IEHPROGM
	an index	IEHPROGM
Catalog	a data set	IEHPROGM
	a generation data set	IEHPROGM
Change	data set organization	IEBUPDTE
	logical record length	IEBGENER
	volume serial number of direct access volume	IEHDASDR
Compare	a partitioned data set	IEBCOMPR
	sequential data sets	IEBCOMPR
Compress-in-place	a partitioned data set	IEBCOPY
Connect	volumes	IEHPROGM
Construct	records from MTST and MTDI input	IEBTCRIN
Convert to partitioned	a sequential data set created as a result of an unload	IEBCOPY
	sequential data sets	IEBUPDTE, IEBGENER
Convert to sequential	a partitioned data set	IEBUPDTE, IEBCOPY
	an indexed-sequential data set	IEBISAM, IEBDG
Copy	a catalog	IEHMOVE
	a direct access volume	IEHDASDR, IBCDMPRS, IEHMOVE
	a partitioned data set	IEBCOPY, IEHMOVE
	a volume of data sets	IEHMOVE
	an indexed-sequential data set	IEBISAM
	cataloged data sets	IEHMOVE
	dumped data from tape to direct access	IEHDASDR, IBCDMPRS
	job steps	IEBEDIT
	members	IEBGENER, IEBUPDTE, IEBDG
	selected members	IEBCOPY, IEHMOVE
	sequential data sets	IEBGENER, IEHMOVE, IEBUPDTE
	to tape	IBCDMPRS
Create	a library of partitioned members	IEBUPDTE
	a member	IEBDG
	a sequential output data set	IEBDG
	an index	IEHPROGM
	an output job stream	IEBEDIT
Delete	a password	IEHPROGM
	an index structure	IEHPROGM
	records in a partitioned data set	IEBUPDTE
Dump	a direct access volume	IEHDASDR, IBCDMPRS
Edit	MTDI input	IEBTCRIN
Edit and convert to partitioned	a sequential data set	IEBGENER, IEBUPDTE
Edit and copy	a job stream	IEBEDIT
	a sequential data set	IEBGENER, IEBUPDTE
Edit and list	error statistics by volume (ESV) records	IFHSTATR
Edit and print	a sequential data set	IEBPTPCH
Edit and punch	a sequential data set	IEBPTPCH
Enter	a procedure into a procedure library	IEBUPDTE
Exclude	a partitioned data set member from a copy operation	IEBCOPY, IEHMOVE
Expand	a partitioned data set	IEBCOPY
	a sequential data set	IEBGENER
Generate	test data	IEBDG
Get alternate tracks	on a direct access volume	IEHDASDR, IBCDASDI, IEHATLAS

Guide to Utility Program Functions (cont'd)

Task		Utility Program
Include	changes to members or sequential data sets	IEBUPDTE
Initialize	a direct access volume	IEHDASDR, IBCDASDI
Insert records	into a partitioned data set	IEBUPDTE
Label	magnetic tape volumes	IEHINITT
List	a password entry	IEHPROGM
	a volume table of contents	IEHLIST
	contents of direct access volume on system output device	IEHDASDR
	number of unused directory blocks and tracks	IEBCOPY
	partitioned directories	IEHLIST
	the contents of the catalog (SYSCTLG data set)	IEHLIST
Load	a previously unloaded partitioned data set	IEBCOPY
	an indexed sequential data set	IEBISAM
	an unloaded data set	IEHMOVE
	UCS and FCB buffers of a 3211 or 3203-4	ICAPRTBL
Merge	partitioned data sets	IEHMOVE, IEBCOPY
Modify	a partitioned or sequential data set	IEBUPDTE
Move	a catalog	IEHMOVE
	a volume of data sets	IEHMOVE
	cataloged data sets	IEHMOVE
	partitioned data sets	IEHMOVE
	sequential data sets	IEHMOVE
Number records	in a new member	IEBUPDTE
	in a partitioned data set	IEBUPDTE
Password protect	add a password	IEHPROGM
	delete a password	IEHPROGM
	list passwords	IEHPROGM
	replace a password	IEHPROGM
Print	a sequential data set	IEBGENER, IEBUPDTE, IEBPTPCH
	partitioned data sets	IEBPTPCH
	selected records	IEBPTPCH
Punch	a partitioned data set member	IEBPTPCH
	a sequential data set	IEBPTPCH
	selected records	IEBPTPCH
Read	Tape Cartridge Reader input	IEBTCRIN
Reblock	a paritioned data set	IEBCOPY
	a sequential data set	IEBGENER, IEBUPDTE
Recover	data from defective tracks on direct access volumes	IEHATLAS
Release	a connected volume	IEHPROGM
Rename	a partitioned data set member	IEBCOPY, IEHPROGM
	a sequential or partitioned data set	IEHPROGM
	moved or copied members	IEHMOVE
Renumber	logical records	IEBUPDTE
Replace	a password	IEHPROGM
	data on an alternate track	IEHATLAS
	identically named members	IEBCOPY
	logical records	IEBUPDTE
	members	IEBUPDTE
	records in a member	IEBUPDTE
	records in a partitioned data set	IEBUPDTE, IEBCOPY
	selected members	IEBCOPY
	selected members in a move or copy operation	IEHMOVE, IEBCOPY
Restore	a dumped direct access volume from tape	IBCDMPRS, IEHDASDR
Scratch	a volume table of contents	IEHPROGM
	data sets	IEHPROGM
Uncatalog	data sets	IEHPROGM
Unload	a partitioned data set	IEHMOVE, IEBCOPY
	a sequential data set	IEHMOVE
	an indexed sequential data set	IEBISAM
Update	in place a partitioned data set	IEBUPDTE
	TTR Entries in the supervisor call library	IEHIOSUP
Write	IPL records and a program on a direct access volume	IEHDASDR

IBCDASDI

IBCDASDI, an independent utility:

- Assigns alternate tracks to a direct access volume.
- Initializes a direct access volume.

Job Control Statements

Because IBCDASDI is an independent utility, operating-system job control statements are not used.

Control Statements

JOB	indicates the beginning of an IBCDASDI job.
MSG	defines an output device for operator messages.
DADEF	defines the volume to be initialized.
VLD	contains information for constructing an initial volume label and for allocating space for additional labels.
VTOCD	contains information for controlling the location of the volume table of contents.
IPLTEXT (optional)	separates utility control statements from any IPL program text statements.
GETALT	assigns an alternate track on a volume.
END	indicates the end of an IBCDASDI job.
LASTCARD (optional)	used to end a series of stacked IBCDASDI jobs.

VTOC Entries per Track

Device	VTOC Entries per Track
2313	25
2319	25
2305-1	18
2305-2	34
3330	39
3340	24
3350	47

IBCDASDI (cont'd)

Format

Name	Operation	Operands
[label]	JOB	[user-information]
[label]	MSG	TODEV=xxxx ,TOADDR=cuu
[label]	DADEF	TODEV=xxxx ,TOADDR=cuu [,IPL=YES] ,VOLID={serial / SCRATCH} [,FLAGTEST=NO] [,PASSES=n] [,BYPASS=YES] [,MODEL=n]
[label]	VLD	NEWVOLID=serial ,VOLPASS={0 / 1} [,OWNERID=xxxxxxxxxx] [,ADDLABEL=n]
[label]	VTOCD	STRTADR=nnnnn ,EXTENT=nnnn
	IPLTXT	
[label]	GETALT	TODEV=xxxx ,TOADDR=cuu ,TRACK=ccchhhh ,VOLID=serial [,FLAGTEST=NO] [,PASSES=n] [,BYPASS=YES] [,MODEL=n]
[label]	END	[user-information]
	LASTCARD	

IBCDMPRS

IBCDMPRS, an independent utility:

- Copies a direct access volume.
- Copies dumped data from tape to a direct access volume.
- Copies to tape.
- Dumps a direct access volume.
- Restores a dumped direct access volume from tape.

Job Control Statements

Because IBCDMPRS is an independent utility, operating-system job control statements are not used.

Control Statements

JOB	begins an IBCDMPRS job.
MSG	defines an output device for operator messages.
DUMP	identifies the volume to be dumped and the receiving volume.
VDRL	specifies the upper and lower track limits of a partial dump.
RESTORE	identifies the source volume whose data is to be restored and the receiving volume.
END	indicates the end of an IBCDMPRS job.

Format

Name	Operation	Operands
[label]	JOB	[user-information]
[label]	MSG	TODEV=xxxx ,TOADDR=cuu
[label]	DUMP	FROMDEV=xxxx ,FROMADDR=cuu ,TODEV=xxxx ,TOADDR=cuu [,VOLID=serial[,serial]] [,MODE=mm] [,MODEL=n]
[label]	VDRL	BEGIN=nnnnn [,END=nnnnn]
[label]	RESTORE	FROMDEV=xxxx ,FROMADDR=cuu ,TODEV=xxxx ,TOADDR=cuu ,VOLID=serial [,MODE=mm] [,MODEL=n]
[label]	END	[user-information]

ICAPRTBL

ICAPRTBL, an independent utility:

- Loads UCS and FCB buffers of a 3211 or 3203-4.

Job Control Statements

Because ICAPRTBL is an independent utility, operating-system job control statements are not used.

Control Statements

JOB	indicates the beginning of an ICAPRTBL job.
DFN	defines the address of the 3211 or 3203-4.
UCS	contains an image of the characters to be loaded into the UCS buffer.
FCB	defines the image to be loaded into the FCB.
END	indicates the end of an ICAPRTBL job.

Format

Name	Operation	Operands
[label]	JOB	[user-information]
	DFN	ADDR=Cuu [,FOLD={Y / N}] [,DEVT={3203-4 / 3211}] [,UCS={image-id / AN / ALL}]* [,FCB={STD1 / STD2}]
[ucsname]	UCS	ucs-image
[fcbname]	FCB	LPI={6 / 8} ,LNCH=((l,c)[,(l,c)...]) ,FORMEND=x
[label]	END	[user-information]

*AN is the default for a 3203-4, ALL is the default for a 3211.

ICAPRTBL Wait-State Codes

Code	Meaning	Code	Meaning
B01	Visually check the train image printed on the 3211.	B12	Reader not ready.
B02	Missing control card or control card out of order.	B13	Reader unit check (display low main storage location 2 through 7 for sense information).
B03	Incorrect JOB statement.	B14	Reader channel error.
B04	Incorrect DFN statement.	B15	No device end on reader.
B05	Incorrect UCS statement.	B19	Printer not online.
B06	Incorrect FCB statement.	B1B	Printer unit check (display low main storage location 2 through 7 for sense information).
B07	Incorrect END statement.		
B0A	External interrupt.		
B0B	Program check interrupt.	B1C	Printer channel error.
B0C	Machine check interrupt.	B1D	No device end on printer.
B11	Reader not online.		

IEBCOMPR

IEBCOMPR, a data set utility:

- Compares partitioned data sets.
- Compares sequential data sets.

Return Codes

00 - successful completion.
08 - unequal comparison - processing continues.
12 - unrecoverable error - job step terminated.
16 - a user routine passed a return code of 16 to IEBCOMPR - job step is terminated.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEBCOMPR
//SYSPRINT  DD      data set definition (output messages)
//SYSUT1    DD      data set definition (input data set)
//SYSUT2    DD      data set definition (input data set)
//SYSIN     DD      {*     }
                    {DUMMY }

                "IEBCOMPR control statements"

/*
```

Note - If the input is sequential and no user exits are provided, the DUMMY parameter for the SYSIN DD statement is used. In this case, no utility control statements are required.

Control Statements

COMPARE	indicates the organization of a data set.
EXITS	identifies the user exit routines to be used.
LABELS	indicates whether user labels are to be treated as data.

Format

Name	Operation	Operands
[label]	COMPARE	TYPORG={PS / PO}
[label]	EXITS [1]	[INHDR=routinename] [INTLR=routinename] [ERROR=routinename] [PRECOMP=routinename]
[label]	LABELS	DATA={YES / NO / ALL / ONLY}

[1] If you code more than one operand, separate them with commas.

IEBCOPY

IEBCOPY, a data set utility:

- Compress-in-place, a partitioned data set.
- Converts to partitioned, a sequential data set.
- Converts to sequential, a partitioned data set.
- Copies a partitioned data set.
- Copies selected members.
- Excludes a partitioned data set member from a copy operation.
- Expands a partitioned data set.
- Lists the number of unused directory blocks or tracks.
- Loads a previously unloaded partitioned data set.
- Merges partitioned data sets.
- Reblocks a partitioned data set.
- Renames a partitioned data set member.
- Replaces records or selected members in a partitioned data set.
- Unloads a partitioned data set.

Return Codes

00 - indicates successful completion.
04 - indicates a condition from which recovery may be possible.
08 - indicates an unrecoverable error. The job step is terminated.

Job Control Statements

```
//name      JOB
//name      EXEC    PGM=IEBCOPY[,PARM='SIZE=nnnnnnnn[K]']
//SYSPRINT  DD      data set definition (output message)
//INPUT     DD      data set definition (input data set)
//OUTPUT    DD      data set definition (output data set)
//SYSUT3    DD      data set definition (spill data set - optional)
//SYSUT4    DD      data set definition (spill data set - optional)
//SYSIN     DD      *

          "IEBCOPY control statements"

/*
```

The optional PARM information in the EXEC statement is used to define the number of bytes used as a buffer. The nnnnnnnn can be replaced by one to eight digits. The K causes the nnnnnnnn to be multiplied by 1024.

Control Statements

COPY	indicates the beginning of a copy operation.
SELECT	specifies which members in the input data set are to be copied.
EXCLUDE	specifies members in the input data set to be excluded from the copy step.

IEBCOPY (cont'd)

Format

Name	Operation	Operands
[label]	COPY	OUTDD=ddname [,INDD=({ddname1[,ddname2]... {ddname1[,ddname2] [,(ddname2,R)]... {((ddname1,R)[,ddname2]...)})]* [,LIST=NO]
		*The INDD parameter may appear on a separate card; if this option is selected, the INDD parameter is not preceded by a comma (,).
[label]	SELECT	MEMBER={name... {[(] (name,,R)...[)] {[(] name,newname[,R]...[)]}[,...]
[label]	EXCLUDE	MEMBER=[(]membername1[,membername2]...[)]

IEBDG

IEBDG, a data set utility:

- Converts to sequential, an indexed sequential data set.
- Copies or creates members.
- Creates a sequential output data set.
- Generates test data.

Return Codes

00 - successful completion.
04 - a user routine returned a code of 16 to the IEBDG program. The job step is terminated at the user's request.
08 - an error occurred while processing a set of utility control statements. No data is generated following the error. Processing continues normally with the next set of utility control statements, if any.
12 - indicates that an error occurred while processing an input or output data set. The job step is terminated.
16 - an error occurred from which recovery is not possible. The job step is terminated.

Job Control Statements

```
//name      JOB     parameters
//          EXEC    PGM=IEBDG[,PARM=LINECNT=nnnn]
//SYSPRINT  DD      data set definition (output message)
//SEQIN     DD      data set definition (sequential input - optional)
//PARIN     DD      data set definition (partitioned input - optional)
//SEQOUT    DD      data set definition (sequential output - optional)
//PAROUT    DD      data set definition (partitioned output - optional)
//SYSIN     DD      {*    }
                    {DATA}

                "IEBDG control statements"

/*
```

The optional PARM information in the EXEC statement is used to specify the number of lines to be printed between headings in the message data set. The nnnn is a four-digit decimal number that specifies the number of lines (0000 to 9999) to be printed per page of output listing.

The DSNAME parameter for the PARIN and PAROUT DD statements can be coded as DSNAME=setname (membername).

Control Statements

DSD	specifies the ddnames of input and output data sets.
FD	defines the contents and lengths of fields to be used in creating output records.
CREATE	defines the contents of output records.
REPEAT	specifies the number of times a CREATE statement or group of CREATE statements are to be used in generating output records.
END	marks the end of a set of IEBDG utility control statements.

IBM Supplied Patterns

Type	Expressed in Hexadecimal	Expressed in Printable Characters
Alphameric	C1 C2...E9 F0...F9	ABC...Z 0...9
Alphabetic	C1 C2...E9	ABC...Z
Zoned Decimal	F0F0...F0F1	00...01
Packed Decimal	0000...001C (Positive pattern) 0000...001D (Negative pattern)	Not applicable
Binary Number	00...01 (Positive pattern) FF...FF (Negative pattern)	Not applicable
Collating Sequence	40...F9	b¢.<(+|&!$*);¬-/,%_>?:'@'=" A...Z 0...9
Random Number	Random hexadecimal digits	Not applicable
Note: A packed-decimal or binary number is right-aligned in the defined field.		

Format

Name	Operation	Operands
[label]	DSD	OUTPUT=(ddname) [,INPUT=(ddname,...)]
[label]	FD	NAME=name ,LENGTH=length-in-bytes [,STARTLOC=starting-byte-location] [,FILL={'character' / X'2-hexadecimal-digits'}] [,FORMAT=pattern*[,CHARACTER=character]] [,PICTURE=length,{'character-string' / P'decimal-number' / B'decimal-number'}] [,SIGN=sign] [,ACTION=action]** [,INDEX=number[,CYCLE=number] [,RANGE=number]] [,INPUT=ddname] [,FROMLOC=number]
* specifies IBM supplied patterns - see table below. **specifies how the contents of a defined field are to be altered - see table below.		
[label]	CREATE [1]	[QUANTITY=number] [FILL={'character' / X'2-hexadecimal-digits'}] [INPUT={ddname / SYSIN[(cccc)]}] [PICTURE=length,startloc,{'character-string' / P'decimal-number' / B'decimal-number'}] [NAME={name / (name1,name2...) / (name,(COPY=name1,name2...)...)}] [EXIT=routinename]

[1] Use at least one of the optional parameters. If you code more than one operand, separate them with commas.

IEBDG (cont'd)

Format (cont'd)

[label]	REPEAT	QUANTITY=number[,CREATE=number]
[label]	END	

Format =

FORMAT=AN --	alphameric.
FORMAT=ZD --	zoned decimal.
FORMAT=PD --	packed decimal.
FORMAT=CO --	collating sequence.
FORMAT=BI --	binary.
FORMAT=AL --	alphobetic.
FORMAT=RA --	random binary number.

Action =

ACTION=SL --	shift left.
ACTION=SR --	shift right.
ACTION=TL --	truncate left.
ACTION=TR --	truncate right.
ACTION=RO --	roll.
ACTION=WV --	wave.
ACTION=FX --	fixed.
ACTION=RP --	ripple.

IEBEDIT

IEBEDIT, a data set utility:

- Copies job steps.
- Creates an output job stream.
- Edits and copies a job stream.

Return Codes

00 - successful completion.
04 - indicates that an error occurred. The output data set may not be usable as a job stream. Processing continues.
08 - indicates that an unrecoverable error occurred while attempting to process the input, output, or control data set. The job step is terminated.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEBEDIT
//SYSPRINT  DD      data set definition (output message)
//SYSUT1    DD      data set definition (input data set)
//SYSUT2    DD      data set definition (output data set)
//SYSIN     DD      *

          "IEBEDIT control statements"

/*
```

Control Statements

EDIT	indicates which step or steps of a specified job in the input data set are to be included in the output data set. Any number of EDIT statements can be included in an operation, thus including selected jobs in the output data set.

Format

Name	Operation	Operands
[label]	EDIT [1]	[START=jobname] [TYPE={POSITION / INCLUDE / EXCLUDE}] [STEPNAME={name / name-name}[,{name / name-name}],...)] [NOPRINT]

[1] If you code more than one operand, separate them with commas.

IEBGENER

IEBGENER, a data set utility:

- Changes logical record length.
- Converts to partitioned, sequential data sets.
- Copies members.
- Copies sequential data sets.
- Edits and converts to partitioned, a sequential data set.
- Edits and copies a sequential data set.
- Expands a sequential data set.
- Prints a sequential data set.
- Reblocks a sequential data set.

Return Codes

00 - successful completion.
04 - probable successful completion. A warning to the user is written.
08 - processing was terminated after the user requested processing of user header labels only.
12 - an unrecoverable error has occurred. The job step is terminated.
16 - a user routine has passed a return code of 16 to the IEBGENER program. The job step is terminated.

Job Control Statements

```
//name      JOB   parameters
//          EXEC  PGM=IEBGENER
//SYSPRINT  DD    data set definition (output message)
//SYSUT1    DD    data set definition (input data set)
//SYSUT2    DD    data set definition (output data set)
//SYSIN     DD    parameters

          "IEBGENER control statements (when required)"

/*
```

Control Statements

GENERATE	used to indicate the number of member names and alias names, record identifiers, literals, and editing information contained in the control data set.
EXITS	used to indicate that user routines are provided.
LABELS	used to specify user-label processing.
MEMBER	used to specify the member name and alias of member of a partitioned data set to be created.
RECORD	used to define a record group to be processed and to supply editing information.

IEBGENER (cont'd)

Conversion Table

Code	Conversion	Output length (input length = L)
PZ	Packed to unpacked decimal mode	2L-1
ZP	Unpacked to packed decimal mode	(L/2)+C*
HE	H-set BCD to EBCDIC mode	L

* If L is odd, C is 1/2; if L is even, C is 1.

Note: PZ type (packed to unpacked) conversion is impossible for packed decimal records longer than 16K bytes. For ZP type (unpacked to packed conversion, the normal 32K byte maximum applies.

If no conversion is specified, the field is moved to the output area without change.

When the ZP parameter is specified, the conversion is performed in place. The original unpacked field is replaced by the new packed field. Therefore, the ZP parameter must be omitted from subsequent references to that field. If the field is needed in its original unpacked form, it must be referenced prior to the use of the ZP parameter.

Format

Name	Operation	Operands
[label]	GENERATE[1]	[MAXNAME=n] [MAXFLDS=n] [MAXGPS=n] [MAXLITS=n]
[label]	EXITS [1]	[INHDR=routinename] [OUTHDR=routinename] [INTLR=routinename] [OUTTLR=routinename] [KEY=routinename] [DATA=routinename] [IOERROR=routinename] [TOTAL=(routinename,size)]
[label]	LABELS	DATA={YES / NO / ALL / ONLY / INPUT}
[label]	MEMBER	NAME=(name[,alias]...)
[label]	RECORD	[IDENT=(length,'name',input-location) FIELD=([length],[input-location-or-'literal'],[conversion],[output-location])... LABELS=n]

[1] If you code more than one operand, separate them with commas.

IEBIMAGE

IEBIMAGE

IEBIMAGE, a data set utility, processes 3800 Printer control modules:

- Creates and prints a forms control buffer module.
- Creates and prints a character arrangement table module.
- Creates and prints a graphic character modification module.
- Creates and prints a copy modification module.
- Modifies and prints a character arrangement table module.
- Modifies and prints a graphic character modification module.
- Modifies and prints a copy modification module.
- Prints a forms control buffer module.
- Prints a character arrangement table module.
- Prints a graphic character modification module.
- Prints a copy modification module.

Return Codes

00 - successful completion.
04 - unusual exception found; processing continues.
08 - control statement error found; processing continues.
12 - I/O error occurred. Processing continues with no further updates to output library, or job step terminates.
16 - input data set can't be opened; job step terminates.
20 - output data set can't be opened; job step terminates.
24 - invalid parameters passed; processing terminates.

Job Control Statements

```
//name        JOB
//            EXEC PGM=IEBIMAGE
//SYSPRINT    DD dataset definition (output listing)
//SYSIN       DD dataset definition (Library I/O)
//SYSIN       DD *

                IEBIMAGE control statements

/*
```

Control Statements

INCLUDE	specifies a module that is to be modified.
FCB	specifies that a forms control buffer module is to be created, printed.
TABLE	specifies that a character arrangement table module is to be created, modified, printed.
GRAPHIC	specifies that a graphic character modification module is to be created, modified, printed.
COPYMOD	specifies that a copy modification module is to be created, modified, printed.
NAME	specifies the end of an operation.

Name	Operation	Operands
[label]	FCB (Notes 1,2)	[SIZE=110 | SIZE=length in tenths] [LPI=6 | LPI=((N,m),...(N [,n | ,remaining lines]))] [CHx = (line-number,...)]
[label]	TABLE (Notes 1,2)	[LOC= ((table-location[,WCGM location [,WCGM reference | ,00]]),...)] [CGMID=(CGM identification,...)] [GCMLIST= (name,...)]
[label]	GRAPHIC (Notes 1,2,3)	[REF= ((segment-number[current assignment | ,new assignment val]),...) [IBM-supplied GCM | ,GCM=name]] [ASSIGN=(assignment val [,10 | ,pitch])]
[label]	COPYMOD (Notes 1,2)	COPIES=(initial copy number [,number of copies | ,1]) LINES=(initial line number [,number of lines | ,1]) POS=print position TEXT= (([dup factor | 1] C,'character data' | X,'hexadecimal data'),...)
[label]	INCLUDE (Note 1)	name [,DELSEG= (segment number,...)]
[label]	NAME	name

Notes:
1. If more than one operand is used, they must be separated by commas.
2. For print-only option, there must be no operands and no comments.
3. Graphic data statements must follow a statement with the ASSIGN keyword.

Graphic Data Statement

10 Pitch

1 - 18 Each nonblank = one bit	25-29 SEQ=	30,31 number

12 Pitch

1 - 15 Each nonblank = one bit	16-18 blank		25-29 SEQ=	30,31 number

15 Pitch

1 - 3 blank	4 - 15 Each nonblank = one bit	16-18 blank		25-29 SEQ=	30,31 number

IEBISAM

IEBISAM, a data set utility:

- Converts to sequential, a partitioned data set.
- Copies an indexed sequential data set.
- Loads an indexed sequential data set.
- Unloads an indexed sequential data set.

Return Codes

00 - successful completion.
04 - a return code of 04 or 12 was passed to the IEBISAM program by a user routine.
08 - the program terminated operation because an error condition was encountered during processing.
12 - a return code other than 00, 04, 08, or 12 was passed from a user routine to the IEBISAM program. The job step is terminated.
16 - the program terminated operation because an error condition was encountered during processing.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEBISAM,PARM={COPY
                                     {UNLOAD
                                     {LOAD
                                     {PRINTL
                                     {'PRINTL[,N] [,EXIT=routinename]'}
//SYSPRINT  DD      data set definition (output messages)
//SYSUT1    DD      data set definition (input data set)
//SYSUT2    DD      data set definition (output data set)
/*
```

The PARM parameter on the EXEC statement is used to control the execution of IEBISAM.

Control Statements

The IEBISAM program is controlled by job control statements.
No utility control statements are required.

IEBPTPCH

IEBPTPCH, a data set utility:

- Edits and prints a sequential data set.
- Edits and punches a sequential data set.
- Prints a sequential data set.
- Prints partitioned data sets.
- Prints selected records.
- Punches a partitioned data set member.
- Punches a sequential data set.
- Punches selected records.

Return Codes

00 - indicates successful completion.
04 - indicates that either a physical sequential data set is empty or a partitioned data set has no members.
08 - indicates that a member specified for printing does not exist in the input data set. Processing continues with the next member.
12 - indicates that an unrecoverable error occurred or that a user routine passed a return code of 12 to IEBPTPCH. The job step is terminated.
16 - indicates that a user routine passed a return code of 16 to IEBPTPCH. The job step is terminated.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEBPTPCH
//SYSPRINT  DD      data set definition (output message)
//SYSUT1    DD      data set definition (input data set)
//SYSUT2    DD      data set definition (output data set)
//SYSIN     DD      *

          "IEBPTPCH control statements"

/*
```

Control Statements

PRINT or PUNCH	specifies that the data is to be either printed or punched.
TITLE	specifies that a title is to precede the printed or punched data.
EXITS	specifies that user routines are provided.
MEMBER	specifies that the input is a partitioned data set and that a selected member is to be printed or punched.
RECORD	specifies whether editing is to be performed, that is, records are to be printed or punched to non-standard specifications.
LABELS	specifies whether user labels are to be treated as data.

!EBPTPCH (cont'd)

Format

Name	Operation	Operands
[label]	PRINT [1] PUNCH	[PREFORM=A] [PREFORM=M] [TYPORG=PS] [TYPORG=PO] [TOTCONV=XE] Applicable to a PRINT or [TOTCONV=PZ] PUNCH operation. [CNTRL=n] [STRTAFT=n] [STOPAFT=n] [SKIP=n] [MAXNAME=n] [MAXFLDS=n] [MAXGPS=n] [MAXLITS=n]
		[INITPG=n] Applicable only to a PRINT [MAXLINE=n] operation.
		[CDSEQ=n] Applicable only to a PUNCH [CDINCR=n] operation.
[label]	TITLE	ITEM=('title'[,output-location]) [,ITEM...]
[label]	EXITS [1]	[INHDR=routinename] [INTLR=routinename] [INREC=routinename] [OUTREC=routinename]
[label]	MEMBER	{NAME=membername } {NAME=aliasname }
[label]	RECORD [1]	[IDENT=(length, 'name', input-location)] [FIELD=(length, [input-location], [conversion], [output-location]) [,FIELDs...]]
[label]	LABELS	DATA= {YES } {ALL } {NO } {ONLY}

Conversion Table

Code	Conversion	Output Length (Where L is the Input Length)
PZ	Packed to unpacked decimal mode	2L-1
XE	Alphameric to hexadecimal representation	2L

[1] If you code more than one operand, separate them with commas.

IEBTCRIN

IEBTCRIN, a data set utility:

- Constructs records from MTST and MTDI input.
- Edits MTDI input.
- Reads Tape Cartridge Reader input.

Return Codes

00 - normal termination.

04 - warning message issued; execution permitted. Conditions leading to issuance of this code are: (1) SYSPRINT, SYSIN, SYSUT2, or SYSUT3 DD statements missing and (2) DCB parameters missing in SYSUT2 or SYSUT3 DD statements.

12 - Diagnostic error message issued; execution terminated. Conditions leading to issuance of this code are: (1) SYSUT1 DD statement missing, (2) conflicting DCB parameters in DD statements, and (3) invalid or conflicting utility control statements.

16 - Terminal error message issued; execution terminated. Conditions leading to issuance of this code are: (1) permanent input/output errors (not including data checks on the TCR), (2) unsuccessful opening of data sets, (3) requests for termination by user exit routine, (4) insufficient storage available for execution, and (5) user exit routine not found.

Job Control Statements

```
//name          JOB
//              EXEC     PGM=IEBTCRIN
//SYSPRINT      DD       data set reference (output messages)
//SYSUT1        DD       data set definition (input data set)
//SYSUT2        DD       data set definition (output data set-valid records)
//SYSUT3        DD       data set definition (output data set-error records)
//SYSIN         DD       *

              "IEBTCRIN control statements"

/*
```

Special Purpose Codes

MTDI Codes

X'00'	(LZ)	X'1E'	(VOK)	X'74'	(P4)
X'11'	(DUP)	X'3C'	(RM)	X'75'	(P5)
X'12'	(LZS)	X'71'	(P1)	X'76'	(P6)
X'18'	(CAN)	X'72'	(P2)	X'77'	(P7)
X'1D'	(GS)	X'73'	(P3)	X'78'	(P8)

MTST Codes

X'10'	(cr)	X'14'	(CR)	X'51'	(as)
X'11'	(sw)	X'15'	(SW)	X'55'	(AS)
X'13'	(fd)	X'17'	(FD)	X'80'	(src)
				X'81'through X'FF'	

The special purpose codes listed are used by IEBTCRIN when constructing records. Use of these codes causes a message to be issued and the utility to be terminated.

MTDI Codes from TCR

Bit Positions 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11				Bit Positions 0, 1
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2, 3
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit
0000	0	LZ				SP	&	-							0	082	0	
0001	1		DUP					/	P1					A	J		1	
0010	2		LZS						P2					B	K	S	2	
0011	3								P3					C	L	T	3	
0100	4								P4					D	M	U	4	
0101	5								P5					E	N	V	5	
0110	6								P6					F	O	W	6	
0111	7								P7					G	P	X	7	
1000	8		CAN						P8					H	Q	Y	8	
1001	9		ED											I	R	Z	9	
1010	A					ç	!		:									
1011	B					.	$	,	#									
1100	C				RM	<	*	%	@									
1101	D		GS			(	)	-	/									
1110	E		VOK			+	;	>	=									
1111	F					!	¬	?	"									

Special Control:
LZ = Left zero fill
DUP = Duplicate
LZS = Left zero start
ED = End Data
GS = Group Separator

Start of Record (SOR):
P1 = Program level 1
P2 = Program level 2
P3 = Program level 3
P4 = Program level 4
P5 = Program level 5
P6 = Program level 6
P7 = Program level 7
P8 = Program level 8
CAN = Cancel

End of Record (EOR):
RM = Record mark
VOK = Verify OK

This figure represents the character set and control codes as read from an MTDI created cartridge.

IEBTCRIN (cont'd)

MTST Codes from TCR

Bit Positions 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11				Bit Positions 0, 1
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2, 3
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digi
0000	0	z	cr	5	0	l	tab	'	s	src								
0001	1	2	sw	6	9	.	as	i	w									
0010	2	t		e	h	j	sp	p	y									
0011	3	n	fd	k	b	=		q	-									
0100	4	Z	CR	%	)	°	TAB	"	S	SRC								
0101	5	@	SW	ç	(	•	AS	I	W									
0110	6	T		E	H	J	SP	P	Y									
0111	7	N	FD	K	B	+		Q	-									
1000	8	1		7	4	m	bsp	r	o									
1001	9	3	st	8		v		a										
1010	A	x		d	l	g		;	/									
1011	B	u		c		f	stx	,										
1100	C	±		&	$	M	BSP	R	O									
1101	D	#	ST	*		V		A										
1110	E	X		D	L	G		:	?									
1111	F	U		C		F	STX	,										

cr and CR = Carrier return co
sw and SW = Switch co
fd and FD = Feed code
st and ST = stop code
tab and TAB = Tab code
as and AS = Automatic search
sp and SP = Space
bsp and BSP = Backspac
stx and STX = Stop tran
src and SRC = Search

This figure represents the character set and control codes as read from an MTST created cartridge.

IEBTCRIN (cont't)

MTST Codes after Translation by IEBTCRIN with TRANS = STDCL

Bit Position 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11				Bit Positions 0, 1
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11	Bit Positions 2, 3
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	First Hexadecimal Digit
0000	0					SP	&	-									0	
0001	1							/		a	j	°		A	J		1	
0010	2			STX						b	k	s		B	K	S	2	
0011	3									c	l	t		C	L	T	3	
0100	4									d	m	u		D	M	U	4	
0101	5	TAB								e	n	v		E	N	V	5	
0110	6		BSP							f	o	w		F	0	W	6	
0111	7									g	p	×		G	P	X	7	
1000	8									h	q	y		H	Q	Y	8	
1001	9									i	r	z		I	R	Z	9	
1010	A					ç	!		:									
1011	B					.	$	,	#									
1100	C						*	%	@									TAB = Tab code
1101	D	CR				(	)	_	'									CR = Carrier return
1110	E		SRC			+	;	=			±							BSP = Backspace SRC = Search STX = Stop transfer
1111	F							?	"									SP = Space

Note: The STDUC option permits translating both lowercase and uppercase alphabetic characters to uppercase.

IEBTCRIN (cont'd)

Control Statements

TCRGEN	specifies whether MTDI or MTST input is to be processed and the type of processing to be performed.
EXITS	specifies any exit routines provided by the user.

Format

Name	Operation	Operands	Comments
[label]	TCRGEN [1]	[TYPE={MTDI / MTST}]	
		[TRANS={STDUC / STDLC / name / NOTRAN}]	valid only with TYPE=MTST specification.
		[EDIT={EDITD / EDITR / NOEDIT}]	Valid only with TYPE=MTDI specification.
		[VERCHK={NOCHK / VOKCHK}]	Valid only with TYPE=MTDI and either an EDIT=EDITD or EDIT=EDITR specification.
		[MINLN=n]	Valid only with TYPE=MTDI and either an EDIT=EDITD or EDIT=EDITR specification.
		[MAXLN=n]	Default=120
		[REPLACE={X'19' / X'xx'}]	
		[ERROPT={NORMAL / NOERR}]	This operand is ignored if a user routine is specified for the ERROR operand in the EXITS utility control statement.
[label]	EXITS [1]	[ERROR=routine name]	This exit is taken just prior to passing an error record to the error output data set (SYSUT3)
		[OUTREC=routine name]	This exit is taken just prior to passing a record to the normal output data set (SYSUT2).
		[OUTHDR2=routine name]	This exit is taken during the opening of the SYSUT2 data set.
		[OUTHDR3=routine name]	This exit is taken during the opening of the SYSUT3 data set.
		[OUTTLR2=routine name]	This exit is taken during the closing of the SYSUT2 data set.
		[OUTTLR3=routine name]	This exit is taken during the closing of the SYSUT3 data set.

[1] If you code more than one operand, separate them with commas.

IEBUPDTE

IEBUPDTE, a data set utility:

- Changes data set organization.
- Converts to partitioned, sequential data sets.
- Converts to sequential, a partitioned data set.
- Copies members.
- Copies sequential data sets.
- Creates a library of partitioned members.
- Deletes records in a partitioned data set.
- Edits and converts to partitioned, a sequential data set.
- Edits and copies a sequential data set.
- Enters a procedure into a procedure library.
- Includes changes to members or sequential data sets.
- Inserts records into a partitioned data set.
- Modifies a partitioned or sequential data set.
- Numbers records in a new member or in a partitioned data set.
- Prints a sequential data set.
- Reblocks a sequential data set.
- Renumbers logical records.
- Replaces logical records, members, records in a member, or records in a partitioned data set.
- Updates in place, a partitioned data set.

Return Codes

00 - indicates successful completion.
04 - indicates that a control statement is coded incorrectly or used erroneously. If either the input or output is sequential, the job step is terminated. If both are partitioned, the program continues processing with the next function to be performed.
12 - indicates an unrecoverable error. The job step is terminated.
16 - indicates that a label-processing code of 16 was received from a user's label-processing routine. The job step is terminated.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEBUPDTE,PARM=({NEW|MOD}[,inhdr][,intlr])
//SYSPRINT  DD      data set definition (output messages)
//SYSUT1    DD      data set definition (input data set)
//SY9UT2    DD      data set definition (output data set)
//SYSIN     DD      {*   }
                    {DATA}

./          "IEBUPDTE control statements"
(optional data or label statements)
/*
```

IEBUPDTE (cont'd)

Function Statements

A function statement is used to initiate the utility operation. At least one function statement must be provided for each member or data set to be processed. A function statement contains:

1-2	Name	Operation	Operands
./	[label]	ADD[1] REPL CHANGE REPRO	[LIST=ALL] [SEQFLD=dd1] [NEW=PO] Applicable to partitioned [NEW=PS] or sequential organization. [MEMBER=cccccccc] [COLUMN=dd] [UPDATE=INPLACE]
			[INHDR=cccccccc] [INTLR=cccccccc] Applicable to sequential [OUTHDR=cccccccc] organization only. [OUTTLR=cccccccc] [TOTAL=(routinename,size)]
			[NAME=cccccccc] [LEVEL=hh] Applicable to partitioned [SOURCE=x] organization only. [SSI=hhhhhhhh]

[1] If you code more than one operand, separate them with commas.

Detail Statements

A detail statement is used with a function statement for certain applications, such as deleting or renumbering selected logical records. A detail statement contains:

1-2	Name	Operation	Operands
./	[label]	NUMBER[1] DELETE	[SEQ1=cccccccc] Used with the NUMBER or [SEQ2=cccccccc] DELETE statement.
			[SEQ1=ALL] [NEW1=cccccccc] Used only with the NUMBER [INCR=cccccccc] statement. [INSERT=YES]

[1] If you code more than one operand, separate them with commas.

Data Statement

A Data Statement is used with a Function statement, or with a Function statement and a Detail statement. It contains a logical record used as replacement data for an existing logical record, or new data to be incorporated in the output master data set.

Label Statement

The LABEL statement indicates that the following data statements are to be treated as user labels. These new user labels are placed on the output data set. The next Function statement indicates to IEBUPDTE that the last label data statement of the group has been read. The label statement contains:

1-2	Name	Operation
./	[label]	LABEL

IEBUPDTE (cont'd)

ALIAS Statement

An ALIAS statement creates or retains an alias in an output (partitioned) master directory. The ALIAS statement can be used with any of the function statements. Multiple alias names can be assigned to each member. The ALIAS statement contains:

1-2	Name	Operation	Operand
./	[label]	ALIAS	NAME=cccccccc

ENDUP Statement

An ENDUP statement can be used to indicate the end of SYSIN input to this job step. It serves as an end-of-data indication if there is no other indication. The ENDUP statement follows the last group of SYSIN control statements and contains:

1-2	Name	Operation
./	[label]	ENDUP

IEHATLAS

IEHATLAS, a system utility:

- Analyzes tracks on direct access.
- Assigns alternate tracks to a direct access volume.
- Gets alternate tracks on a direct access volume.
- Recovers data from defective tracks on direct access volumes.
- Replaces data on an alternate track.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEHATLAS
//SYSPRINT  DD      data set reference (output messages)
//SYSUT1    DD      data set definition (data set that contains the
                                        bad record)
//SYSIN     DD      *

          "IEHATLAS control statements"

/*
```

Control Statements

The utility control statement consists of either:

```
TRACK=bbbbccccchhhhrrrkkdddd[S]
```

or

```
VTOC=bbbbccccchhhhrrrkkdddd
```

TRACK=
specifies that an alternate track is to be assigned for a track that does not contain VTOC records.

VTOC=
specifies that an alternate track is to be assigned for a track that contains VTOC records.

IEHDASDR

IEHDASDR, a system utility:

- Analyzes tracks on direct access.
- Assigns alternate tracks to a direct access volume.
- Changes the volume serial number of a direct access volume.
- Copies a direct access volume.
- Copies dumped data from tape to direct access.
- Dumps a direct access volume.
- Gets alternate tracks on a direct access volume.
- Initializes a direct access volume.
- Lists the contents of a direct access volume on a system output device.
- Restores a dumped direct access volume from tape.
- Writes IPL records and a program on a direct access volume.

Return Codes

00 - indicates successful completion.
04 - indicates that an unusual condition was encountered; however, the overall result is successful. A warning message is issued.
08 - indicates that a specified operation did not complete successfully. An attempt is made to perform any additional operations.
16 - indicates that either an error occurred upon invoking IEHDASDR, or IEHDASDR was unable to open the input or message data set. The job step is terminated.

Job Control Statements

```
//name      JOB
//           PGM=IEHDASDR [,PARM='N=n'                    ]
                          [,PARM='LINECNT=xx'             ]
                          [,PARM='LINECNT=xx,N=n'         ]
//SYSPRINT  DD      data set definition (output messages)
//anyname   DD      data set definition (direct access device)
//tapename  DD      data set definition (magnetic tape unit)
//SYSIN     DD      *
          "IEHDASDR control statements"
/*
```

The optional PARM information is used by the program to control line density on output listings, and to indicate the maximum number of operations of the same type that can be performed concurrently in the job step.

LINECNT=xx
: specifies the number of lines per page in the listing of the SYSPRINT data set. The number xx is a 2-digit decimal number ranging from 01 to 99.

N=n
: specifies a decimal number from 1 to 6. The number represents the maximum number of like functions that can be performed concurrently by the IEHDASDR program.

Control Statements

ANALYZE	used to analyze the recording surface to test for defective tracks, assign alternates for any defective tracks found, and format the volume to make it ready for use.
FORMAT	used to make a volume ready for use without performing an analysis of the recording surface.
LABEL	used to change the volume serial number of a direct access volume and, optionally, to update the owner field.

Control Statements (cont'd)

GETALT	used to assign an alternate track for a specified track.
DUMP	used to dump a single track, a group of tracks, or an entire direct access volume.
RESTORE	used to restore a previously dumped direct access volume to a direct access device.
IPLTXT	signals the beginning of IPL program text statements.
PUTIPL	specifies that IPL records and a program are to be written on a direct access device.

Format

Name	Operation	Operands
[label]	ANALYZE	{TODD=(cuu,...) / TODD=(ddname,...)} ,VTOC=xxxxx ,EXTENT=xxxxx [,NEWVOLID=serial] [,IPLDD=ddname] [,FLAGTEST={YES / NO}] [,PASSES={n / 0'}] [,OWNERID=name] [,PURGE={YES / NO}] [,MSS]
[label]	FORMAT	TODD=(ddname,...) ,VTOC=xxxxx ,EXTENT=xxxxx [,NEWVOLID=serial] [,IPLDD=ddname] [,OWNERID=name] [,PURGE={YES / NO}]
[label]	LABEL	TODD={cuu / ddname} ,NEWVOLID=serial [,OWNERID=name]
[label]	GETALT	TODD=ddname ,TRACK=ccccchhhh
[label]	DUMP	FROMDD=ddname ,TODD=(ddname,...) [,CPYVOLID={YES / NO}] [,BEGIN=ccccchhhh] [,END=ccccchhhh] [,PURGE={YES / NO}]
[label]	RESTORE	TODD=(ddname,...) ,FROMDD=ddname [,CPYVOLID={YES / NO}] [,PURGE={YES / NO}]
[label]	IPLTXT	

IEHDASDR (cont'd)

Format (cont'd)

[label]	PUTIPL	FROMDD=ddname ,TODD=ddname [,PURGE={YES} {NO}]

PURGE=YES - Operator Replies

Reply	Meaning
U	All unexpired data sets on the volume can be overwritten. (The operation continues.)
T	The volume contains unexpired data sets that must not be overwritten. (The operation is terminated.)

IEHINITT

IEHINITT, a system utility:

- Labels magnetic tape volumes.

Return Codes

00 - successful completion. A message data set was created.
04 - successful completion. No message data set was defined by the user.
08 - the program completed its operation but error conditions were encountered during processing. A message data set was created.
12 - the program completed its operation but error conditions were encountered during processing. No message data set was defined by the user.
16 - the program terminated operation because of error conditions encountered while attempting to read the control data set. A message data set was created if defined by the user.

Job Control Statements

```
//name      JOB
//          EXEC     PGM=IEHINITT[,PARM=LINECNT=nn]
//SYSPRINT  DD       data set definition (sequential output)
//anyname   DD       data set definition (tape unit-labeling)
//SYSIN     DD       *
          "INITT control statement(s)"
/*
```

The optional PARM information on the EXEC statement specifies the number of lines to be printed between headings in the message data set.

Control Statements

The IEHINITT program uses an INITT utility control statement to provide control information for a labeling operation. Any number of INITT utility control statements can be included for a given execution of the program. An identically named DD statement must exist for a utility control statement in the job step.

Format

Name	Operation	Operands
label	INITT	SER=xxxxxx [,OWNER='ccccccccc[cccc]'] [,NUMBTAPE=n] {,DISP=REWIND} {,DISP=UNLOAD} [,LABTYPE=AL]

IEHIOSUP

IEHIOSUP, a system utility:

- Updates TTR entries in the supervisor call library.

Return Codes

00 - successful completion.
12 - an unrecoverable error has occurred. The job step is terminated.

Job Control Statements

```
//name      JOB
//          EXEC    PGM=IEHIOSUP
//SYSUT1    DD      data set definition (object data set - SYS1.SVCLIB)
//SYSPRINT  DD      data set definition (output messages)
/*
```

Control Statements

IEHIOSUP is executed or invoked with job control statements. No utility control statements are required.

IEHLIST

IEHLIST, a system utility:

- Lists a volume table of contents.
- Lists partitioned directories.
- Lists the contents of the catalog (SYSCTLG) data set.

Return Codes

00 - successful completion.
08 - due to an error condition, a specified request was ignored. Processing continues.
12 - indicates that a permanent input/output error occurred. The job is terminated.
16 - indicates that an unrecoverable error occurred while reading the data set. The job is terminated.

Job Control Statements

```
//name      JOB     parameters
//          EXEC    PGM=IEHLIST[,PARM='LINECNT=xx']
//SYSPRINT  DD      data set definition (output message)
//anyname1  DD      data set definition (permanently mounted volume)
//anyname2  DD      data set definition (mountable device type)
//SYSIN     DD      *
          "IEHLIST control statements"
/*
```

The optional PARM information on the EXEC statement specifies the number of lines to be printed per page. The value of xx is a decimal number from 01 through 99.

Control Statements

LISTCTLG	used to request a listing of all or part of a catalog.
LISTPDS	used to request a directory listing of one or more partitioned data sets.
LISTVTOC	used to request a listing of all or part of a volume table of contents.

Format

Name	Operation	Operands
[label]	LISTCTLG [1]	[VOL=device=serial] [NODE=name]
[label]	LISTPDS [1]	DSNAME=(name[,name]...) [VOL=device=serial] {DUMP} {FORMAT}
[label]	LISTVTOC [1]	{DUMP} {FORMAT} [DATE=dddyy] [VOL=device=serial] [DSNAME=(name[,name]...)

[1] If you code more than one operand, separate them with commas.

IEHMOVE

IEHMOVE, a system utility:

- Copies a catalog.
- Copies a direct access volume.
- Copies a partitioned data set.
- Copies a volume of data sets.
- Copies cataloged data sets.
- Copies selected members.
- Copies sequential data sets.
- Excludes a partitioned data set member from a copy operation.
- Loads an unloaded data set.
- Merges partitioned data sets.
- Moves a catalog.
- Moves a volume of data sets.
- Moves cataloged data sets.
- Moves partitioned data sets.
- Moves sequential data sets.
- Renames moved or copied members.
- Replaces selected members in a move or copy operation.
- Unloads a partitioned data set.
- Unloads a sequential data set.

Return Codes

00 - successful completion.
04 - a specified function was not completely successful. Processing continues.
08 - a condition has occurred from which recovery is possible. Processing continues.
12 - an unrecoverable error has occurred. The job step is terminated.
16 - impossible to open the SYSIN or SYSPRINT data set.

Job Control Statements

```
//name      JOB     parameters
//          EXEC    PGM=IEHMOVE [,PARM= {'POWER=n'
                                        {'POWER=n,LINECNT=xx'
                                        {'LINECNT=xx'        ]
//SYSPRINT  DD      data set definition (output message)
//SYSUT1    DD      data set definition (work data set)
//anyname1  DD      data set definition (permanently mounted volume)
//anyname2  DD      data set definition (mountable device type)
//tape      DD      data set definition (tape volume)
//SYSIN     DD      *
          "IEHMOVE control statements"
/*
```

The optional PARM information in the EXEC statement is used to allocate additional work space and/or control line density on output listings.

The POWER=n parameter is used to request that the normal amount of space for work area is to be increased n times.

The LINECNT=xx parameter specifies the number of lines per page in the listing of the SYSPRINT data set.

Control Statements

MOVE DSNAME	used to move a data set.
COPY DSNAME	used to copy a data set.
MOVE DSGROUP	used to move a group of cataloged data sets.
COPY DSGROUP	used to copy a group of cataloged data sets.
MOVE PDS	used to move a partitioned data set.

IEHMOVE (cont'd)

Control Statements (cont'd)

COPY PDS	used to copy a partitioned data set.
MOVE CATALOG	used to move cataloged entries.
COPY CATALOG	used to copy cataloged entries.
MOVE VOLUME	used to move a volume of data sets.
COPY VOLUME	used to copy a volume of data sets.

In addition, there are four subordinate control statements that can be used to modify the effect of a MOVE or COPY DSGROUP, MOVE or COPY PDS, or MOVE or COPY CATALOG operation. The subordinate statements and the control statements with which they can be combined are shown in the following table:

Valid Combinations of Control Statements

Utility Statements	Subordinate Statements
MOVE DSGROUP or COPY DSGROUP	INCLUDE EXCLUDE
MOVE PDS or COPY PDS	INCLUDE EXCLUDE REPLACE SELECT
MOVE CATALOG or COPY CATALOG	EXCLUDE

Format

Name	Operation	Operands
[label]	MOVE	DSNAME=name ,TO=device=list [,FROM=device=list] [,CVOL=device=serial] [,UNCATLG] [,RENAME=name] [,FROMDD=ddname] [,TODD=ddname]
[label]	COPY	DSNAME=name ,TO=device=list [,FROM=device=list] [,CVOL=device=serial] [,UNCATLG] [,CATLG] [,RENAME=name] [,FROMDD=ddname] [,TODD=ddname]
[label]	MOVE	DSGROUP [=name] ,TO=device=list [,CVOL=device=serial] [,PASSWORD] [,UNCATLG] [,TODD=ddname]

IEHMOVE (cont'd)

Format (cont'd)

[label] COPY	DSGROUP [=name] ,TO=device=list [,CVOL=device=serial] [,PASSWORD] [,UNCATLG] [,CATLG] [,TODD=ddname]
[label] MOVE	PDS=name ,TO=device=serial [,FROM=device=serial] [,CVOL=device=serial] [,EXPAND=nn] [,UNCATLG] [,RENAME=name] [,FROMDD=ddname] [,TODD=ddname]
[label] COPY	PDS=name ,TO=device=serial [,FROM=device=serial] [,CVOL=device=serial] [,EXPAND=nn] [,UNCATLG] [,CATLG] [,RENAME=name] [,FROMDD=ddname] [,TODD=ddname]
[label] MOVE	CATALOG[=name] ,TO=device=serial [,CVOL=device=serial] [,FROM=device=serial] [,FROMDD=ddname] [,TODD=ddname]
[label] COPY	CATALOG[=name] ,TO=device=serial [,CVOL=device=serial] [,FROM=device=serial] [,FROMDD=ddname] [,TODD=ddname]
[label] MOVE	VOLUME=device=serial ,TO=device=list [,PASSWORD] [,TODD=ddname]
[label] COPY	VOLUME=device=serial ,TO=device=list [,PASSWORD] [,CATLG] [,TODD=ddname]
[label] INCLUDE	DSNAME=name [,MEMBER=membername] [,FROM=device=list] [,CVOL=device=serial]
[label] EXCLUDE	{DSGROUP=name} {MEMBER=membername}
[label] SELECT	{MEMBER=(name[,name]...)} {MEMBER=((name,newname)[,(name,newname)]...)}
[label] REPLACE	DSNAME=name ,MEMBER=membername [,FROM=device=serial] [,CVOL=device=serial]

IEHPROGM

IEHPROGM, a system utility:

- Adds a password.
- Builds a generation - data - group index.
- Builds a generation.
- Builds an index.
- Catalogs a data set.
- Catalogs a generation data set.
- Connects volumes.
- Creates an index.
- Deletes a password.
- Deletes an index structure.
- Lists a password entry.
- Password - protects add, delete, list, or replace password operations.
- Releases a connected volume.
- Renames a partitioned data set member.
- Renames a sequential or partitioned data set.
- Replaces a password.
- Scratches a volume table of contents.
- Scratches data sets.
- Uncatalogs data sets.

Return Codes

00 - successful completion.
04 - a syntax error has been found in the name field of the control statement or in the PARM field in the EXEC statement. Processing is continued.
08 - a request for a specific operation has been ignored because of an invalid control statement or an otherwise invalid request. The operation is not performed.
12 - an I/O error has been detected when trying to read or write from or onto SYSPRINT, SYSIN, or the VTOC.
16 - an unrecoverable error has occurred. The job step is terminated.

Job Control Statements

```
//name      JOB   parameters
//          EXEC  PGM=IEHPROGM [,PARM='LINECNT=xx {,PRINT  }{,1}]
                                                 {,NOPRINT}
//SYSPRINT  DD    data set definition (output message)
//anyname1  DD    data set definition (permanently mounted volume)
//anyname2  DD    data set definition (mountable device type)
//SYSIN     DD    *

          "IEHPROGM control statements"

/*
```

The optional PARM information on the EXEC statement is used to control the number of lines per page on the output listing and to suppress printing of utility control statements. The value xx is a 2-digit decimal number from 01 through 99.

Control Statements

SCRATCH	used to delete a data set or member from a direct access volume.
RENAME	used to change the name or alias of a data set or member residing on a direct access volume.
CATLG	used to generate an entry in the index of a catalog.
UNCATLG	used to remove an entry from the lowest level index of the catalog.
BLDX	used to create a new index in the catalog.
DLTX	used to remove a low level index from the catalog.

IEHPROGM (cont'd)

Control Statements (cont'd)

BLDA	used to assign an alias previously assigned to an index at the highest level of the catalog.
DLTA	used to delete an alias previously assigned to an index at the highest level of the catalog.
CONNECT	used to place an entry into an index at the highest level of the catalog.
RELEASE	used to remove an entry from the highest level index of a volume.
BLDG	used to build an index for a generation data group and establish the action to be taken should the index overflow.
ADD	used to add a password entry into the PASSWORD data set.
REPLACE	used to replace information in a password entry.
DELETEP	used to delete an entry in the PASSWORD data set.
LIST	used to format and list information from a password entry.

Format

Name	Operation	Operands
[label]	SCRATCH	{DSNAME=name {VTOC } ,VOL=device=list [,PURGE] [,MEMBER=name] [,SYS]
[label]	RENAME	DSNAME=name ,VOL=device=list ,NEWNAME=name [,MEMBER=name]
[label]	CATLG	DSNAME=name ,VOL=device=list [,CVOL=device=serial]
[label]	UNCATLG	DSNAME=name [,CVOL=device=serial]
[label]	BLDX	INDEX=name [,CVOL=device=serial]
[label]	DLTX	INDEX=name [,CVOL=device=serial]
[label]	BLDA	INDEX=name ,ALIAS=name [,CVOL=device=serial]
[label]	DLTA	ALIAS=name [,CVOL=device=serial]
[label]	CONNECT	INDEX=name ,VOL=device=serial [,CVOL=device=serial]

IEHPROGM (cont'd)

Format (cont'd)

Name	Operation	Operands
[label]	RELEASE	INDEX=name [,CVOL=device=serial]
[label]	BLDG	INDEX=name ,ENTRIES=n [,CVOL=device=serial] [,EMPTY] [,DELETE]
[label]	ADD	DSNAME=name [,PASWORD2=new-password] [,CPASWORD=control-password] [,TYPE=code] [,VOL=device=list] [,DATA='user-data']
[label]	REPLACE	DSNAME=name [,PASWORD1=current-password] [,PASWORD2=new-password] [,CPASWORD=control-password] [,TYPE=code] [,VOL=device=list] [,DATA='user-data']
[label]	DELETEP	DSNAME=name [,PASWORD1=current-password] [,CPASWORD=control-password] [,VOL=device=list]
[label]	LIST	DSNAME=name ,PASWORD1=current-password

IFHSTATR

IFHSTATR, a system utility:

- Edits and lists error statistics by volume (ESV) records.

Job Control Statements

```
//              JOB
//              EXEC    PGM=IFHSTATR
//SYSUT1        DD      data set definition (input data set)
//SYSUT2        DD      data set definition (output data set)
/*
```

Control Statements

IFHSTATR is controlled by job control statements. Utility control statements are not used.

Definition of Operands

ACTION=	action	specifies that the contents of a defined field are to be altered after the field's inclusion in an output record.
ADDR=	cuu	specifies the channel number, c, and unit number, uu, of the 3211 or 3203-4.
ADDLABEL=	n	specifies the total number of additional labels for which space is to be allocated. The value can be 1 through 7.
ALIAS=	name	specifies an unqualified name to be assigned as the alias, or specifies the unqualified name of the index alias to be deleted.
BEGIN=	ccccchhhh	specifies in hexadecimal a cylinder number, cccc, and head number, hhhh, that identifies the first track to be dumped.
	nnnnn	specifies a one- to five-byte reltative track address that identifies the first track to be dumped.
BYPASS=	YES	specifies that no check is to be made for defective tracks.
CATALOG	[=name]	specifies the catalog entries to be moved or copied.
CATLG		specifies that the copied data set(s) is to be cataloged on its receiving volume(s) if it is a direct access volume. If a catalog does not exist on the receiving volume, it is created.
CDINCR=	n	specifies the increment to be used in generating sequence numbers. If CDINCR is omitted and CDSEQ is coded, 10 is assumed as an increment value for sequence numbering.
CDSEQ	n	specifies the initial sequence number of a deck of punched cards.
CHARACTER=	character	specifies the starting character of a field.
CNTRL=	n	specifies a control character for the output device that either indicates line spacing, or is used to select the stacker as follows: 1 indicates single spacing or first stacker; 2 indicates double spacing or second stacker; and 3 indicates triple spacing.

Definition of Operands (cont'd)

COLUMN=	dd	specifies, in decimal, the starting column of a data field within a logical record image. The field extends to the end of the image. Column is valid only when CHANGE is coded.
CPASWORD=	control-password	specifies the control password for the data set.
CPYVOLID=	YES	specifies that all receiving or restored direct access volumes are to be assigned the serial number of the dumped volume.
	NO	specifies that receiving or restored volumes are to keep their own serial numbers.
CVOL=	device=serial	specifies the device type and volume serial number of the volume, catalog entry, or index to be operated upon.
CYCLE=	number	specifies a number of output records that are treated as a group by the INDEX keyword.
DATA=	ALL	specifies that user labels are to be treated as data regardless of any return code.
	INPUT	specifies that user labels for the output data set are supplied as 80 byte input records in the data portion of SYSIN.
	NO	specifies that user labels are not to be treated as data.
	ONLY	specifies that only user header labels are to be treated as data.
	routinename	specifies the symbolic name of a routine that modifies the physical record before it is processed by IEBGENER.
	'user-data'	specifies that user data is to be included in the password entry. The user data must be in single quotes and must not exceed 77 characters.
	YES	specifies that any user labels that are not rejected by a user's label processing routine are to be treated as data.
DATE=	dddyy	specifies that each entry that expires before this date is to be flagged with an asterisk(*) in the listing.
DELETE		specifies that generation data sets are to be scratched after their entries are removed from the index.
DEVT=	3203-4 3211	specifies the 3203-4 printer. specifies the 3211 printer

Definition of Operands (cont'd)

DISP=	REWIND	specifies that a tape is to be rewound (but not unloaded) after the label has been written.
	UNLOAD	specifies that a tape is to be unloaded after the label has been written.
DSGROUP	=name	specifies a qualified name.
	[=name]	specifies the cataloged data sets to be processed.
DSNAME=	name	specifies the fully qualified name of the data set to be processed.
	(name[,name]...)	specifies the fully qualified names of the data sets whose directories or entries are to be listed.
DUMP		specifies that the listing is to be in unedited, hexadecimal form.
EDIT=	EDITD	specifies that the input is to be edited and that SOR and EOR codes are to be deleted and not included as part of the output record.
	EDITR	specifies that the input is to be edited and SOR and EOR codes are to be kept as part of the output record.
	NOEDIT	specifies that no editing is to be performed.
EMPTY		specifies that all entries be removed from the generation-data-group index when it overflows.
END=	ccchhhh	specifies, in hexadecimal, a cylinder number, cccc, and head number, hhhh, that identify the last track to be dumped.
	nnnnn	specifies the relative track address of the last track to be dumped.
ENTRIES=	n	specifies the number of entries to be contained in the generation-data-group index; n must not exceed 255.
ERROPT=	NORMAL	specifies that all error records are to be placed in the error data set (SYSUT3).
	NOERR	specifies that all records (including error records) are placed in the normal output data set (SYSUT2). No records are placed in the error data set (SYSUT3).
ERROR=	routinename	specifies the symbolic name of a routine that is to receive control for error handling.
EXIT=	routinename	specifies the name of a user routine that is to receive control from IEBDG before writing each output record.
EXPAND=	nn	specifies the number of 256-byte records (up to 99 decimal) to be added to the directory of the specified partitioned data set.
EXTENT=	nnnn	specifies the length (number of tracks) of the VTOC.
	xxxxx	specifies the decimal length of the VTOC in tracks.
FCB=	STD1	specifies the standard FCB image-id STD1 (6 lines per inch on an 8.5 inch form).
	STD2	specifies the standard FCB image-id STD2 (6 lines per inch on an 11 inch form).
FIELD=	conversion	specifies a two-byte code that indicates the type of conversion to be performed on this field.

Definition of Operands (cont'd)

FIELD= (cont'd)	input-location	specifies the starting byte of the field to be processed.
	length	specifies the length (in bytes) of the input field or literal to be processed.
	'literal'	specifies a literal (maximum length of 40 bytes) to be replaced in the specified output location.
	output-location	specifies the starting location of this field in the output records.
FILL=	'character'	specifies an EBCDIC character to be placed in each byte of the defined field or output record.
	X'2 hex-digits'	specifies two hexadecimal digits to be placed in each byte of the defined field or output record.
FLAGTEST=	NO	specifies that the program is not to check for previously flagged tracks on this volume.
	YES	specifies that each track is to be checked to see if it was previously flagged as defective.
FOLD=	N	specifies that lower case letters are not to be printed as upper case letters.
	Y	specifies that lower case letters are to be printed as upper case letters when the lower case print train is not available.
FORMAT		specifies that the listing is to be edited for each directory entry, or that a comprehensive edited listing is to be generated.
FORMAT=	pattern	specifies an IBM-supplied pattern that is to be placed in the defined field. FORMAT= must not be used when PICTURE is used.
FORMEND=	x	specifies the number of lines (max. 180) on the printer form. For an 11 inch form, spacing six lines per inch, x must be 66.
FROM=	device= list	specifies the volume or volumes on which the data set currently resides, if it is not cataloged.
	device= serial	specifies the device type and volume serial number of the volume to be processed.
FROMADDR=	cuu	specifies channel number, c, and unit number, uu, of the source device.
FROMDD=	ddname	specifies the ddname of the DD statement defining the device that contains the appropriate input data.
FROMDEV=	xxxx	specifies the type of the source device, for example, 3330 or 2400.
FROMLOC=	number	specifies the location of the selected field within the input logical record.
IDENT=	input-location	specifies the starting location of the field that contains the identifying name in the input records.

Definition of Operands (cont'd)

IDENT= (cont'd)	length	specifies the length (in bytes) of the identifying name of the last record of the input group to which the FIELD parameters or member statement applies. The length cannot exceed eight characters.
	'name'	specifies the exact literal that identifies the last record of a record group.
INCR=	cccccccc	specifies the increment value used for assigning successive sequence numbers to new or replacement logical records, or specifies an increment value used for renumbering existing logical records.
INDD=	ddname	specifies the ddname which is indicated on a DD statement of an input data set.
	R	specifies that all members to be copied or loaded from this input data set are to replace any identically named members on the output partitioned data set.
INDEX=	name	specifies the qualified name of the index to be processed, or specifies the unqualified index name to be acted upon.
	number	specifies a number to be added to this field whenever a specified number of records have been written.
INHDR=	cccccccc	specifies the symbolic name of a user routine that handles any user input (SYSUT1) header labels.
	routinename	specifies the symbolic name of a routine that processes user input header labels.
INITPG=	n	specifies the initial page number; the pages are numbered sequentially thereafter.
INPUT=	ddname (ddname,...)	specifies the ddname of a DD statement defining a data set used as input to the program.
	SYSIN[(cccc)]	specifies that the SYSIN data set contains records (other than utility control statements) to be used in the construction of output records.
INREC=	routinename	specifies the symbolic name of a routine that manipulates each logical record before it is processed.
INSERT=	YES	specifies the insertion of a block of logical records.
INTLR=	cccccccc	specifies the symbolic name of the user routine that handles input (SYSUT1) trailer labels.
	routinename	specifies the symbolic name of a routine that processes user input trailer labels.
IOERROR=	routinename	specifies the symbolic name of a routine that handles permanent I/O error conditions.
IPL=	YES	specifies that an IPL program is to be written on the volume.
IPLDD=	ddname	specifies the ddname of a DD statement defining the data set containing the IPL program.

Definition of Operands (cont'd)

ITEM=	output-location	specifies the starting position at which a literal for this item is to be placed in the output record.
	'title'	specifies the title or subtitle literal (maximum length of 40 bytes), enclosed in apostrophes.
KEY=	routinename	specifies the symbolic name of a routine that creates the output record key.
LABELS=	n	indicates the number of records in the SYSIN data set to be treated as user labels. The number n is a number from 1 to 8. If this parameter is included, DATA=INPUT must be coded on a LABELS statement before it is in the input stream.
LABTYPE=	AL	specifies that an ANS volume label is to be created.
LENGTH=	length in bytes	specifies the length in bytes of the defined field.
LEVEL=	hh	specifies the change (update) level in hexadecimal (00-FF). This parameter is valid only when a member of a partitioned data set is being processed.
LIST=	ALL	specifies that the SYSPRINT data set is to contain the entire updated member or data set and the control statements used in its creation.
	NO	specifies that the names of copied members are not to be listed on SYSPRINT at the end of each input data set.
LNCH=	((l,c)[,(l,c,)...])	specifies the channels of the FCB image. Each set of parentheses must contain the line number (1-180), a comma, and the channel number (1-12) to be assigned to that line. One or all of the 12 channels may be assigned in any order. Each set must be separated by commas and the entire group surrounded by parentheses.
LPI=	6	specifies that six lines per inch will be printed.
	8	specifies that eight lines per inch will be printed.
MAXFLDS=	n	specifies a number that is no less than the total number of FIELD parameters appearing in subsequent RECORD statements. MAXFLDS is required if there are any FIELD parameters in subsequent RECORD statements.
MAXGPS=	n	specifies a number that is no less than the total number of IDENT parameters appearing in subsequent RECORD statements. MAXGPS is required if there are any IDENT parameters in subsequent RECORD statements.
MAXLINE=	n	specifies the maximum number of lines to a printed page. Spaces, titles, and subtitles are included in this number.
MAXLITS=	n	specifies a number that is no less than the total number of characters contained in the FIELD or IDENT literals of subsequent RECORD statements.
MAXLN=	n	specifies the number of bytes, plus four for the record descriptor word when variable records are specified, to be contained in all but the last record passed to the output routine when editing is not performed.

Definition of Operands (cont'd)

MAXNAME=	n	specifies a number that is no less than the total number of member names and aliases appearing in subsequent MEMBER statements. MAXNAME is required if there are one or more MEMBER statements.
MEMBER=	cccccccc	specifies a name to be assigned to the member placed in the partitioned data set defined by the SYSUT2 DD statement.
	membername	specifies the name of a member of the partitioned data set named in the DSNAME parameter, or identifies a member to be excluded from the partitioned data set being moved or copied when the EXCLUDE statement modifies a MOVE partitioned data set or COPY partitioned data set statement.
	[(] membername1[,membername2]...[)]	specifies members on the input data set that are not to be copied, unloaded, or loaded to the output data set. The members are not deleted from the input data set unless the entire data set is deleted.
	name	specifies the name or alias for a member (in the named data set) that is to be processed.
	(name[,name]...)	identifies the members to be moved or copied.
	((name,newname)[,(name,newname)]...)	identifies the members to be moved or copied and gives the new name for each member.
	newname	specifies a newname for a selected member.
	R	specifies that the input member is to replace any identically named member that exists on the output partitioned data set. The replace option is not valid for an unload operation.
MINLN=	n	specifies the byte length of the shortest valid edited record.
MODE=	mm	specifies the bit density for data written onto the receiving magnetic tape volume.
MODEL=	n	specifies a decimal model number (1 or 2) for the 2305.
NAME=	aliasname cccccccc	indicates the name of the member placed in the partitioned data set, or specifies a one- to eight-character alias depending on the operation.
	membername	specifies a member by its member name.
	name (name1,...) (name1,namen...) (name,(COPY=name1,namen...)...)	specifies the name of the field defined by this FD statement, or specifies the name(s) of a field(s) to be included in the applicable output records. (cont'd)

Definition of Operands (cont'd)

Continued from preceding page		COPY indicates that all fields named in the inner parentheses (maximum of twenty) are to be treated as a group and included the specified number of times in each output record.
	(name[,alias]...)	specifies a member name followed by a list of its aliases.
NEW=	PO	specifies that the old master data set is a sequential data set, and that the updated output is to become a member of a partitioned data set.
	PS	specifies that the old master data set is a partitioned data set, and that a member of that data set is to be converted into a sequential data set.
NEWNAME=	name	specifies the new fully-qualified name for the data set, or the new member of alias.
NEW1=	cccccccc	specifies the first sequence number assigned to new or replacement data, or specifies the first sequence number assigned in a renumbering operation.
NEWVOLID=	serial	specifies a one- to six-character volume serial number.
NODE=	name	specifies a qualified name.
NOPRINT		specifies that the message data set is not to include a listing of the output data set.
NUMBTAPE=	n	specifies the number of tapes to be labeled according to specifications made in this control statement. The value n represents a number from 1 to 255.
OUTDD=	ddname	specifies the name of the output partitioned data set.
OUTHDR=	cccccccc	specifies the symbolic name of the user routine that handles any user output (SYSUT2) header labels.
	routinename	specifies the symbolic name of a routine that creates user output header labels.
OUTHDR2=	routinename	specifies the symbolic name of a routine that receives control during the opening of the SYSUT2 data set.
OUTHDR3=	routinename	specifies the symbolic name of a routine that receives control during the opening of the SYSUT3 data set.
OUTPUT=	(ddname)	specifies the ddname of the DD statement defining the output data set.
OUTREC=	routinename	specifies the symbolic name of a routine that manipulates each logical record before it is printed or punched, or specifies the symbolic name of a routine that receives control before the record is passed to the normal output data set (SYSUT2).
OUTTLR=	cccccccc	specifies the symbolic name of the user routine that handles any user output (SYSUT2) trailer labels.
	routinename	specifies the symbolic name of a routine that processes user output trailer labels.
OUTTLR2=	routinename	specifies the symbolic name of a routine that receives control during the closing of the SYSUT2 data set.

Definition of Operands (cont'd)

OUTTLR3=	routinename	specifies the symbolic name of a routine that receives control during the closing of the SYSUT3 data set.
OWNER=	'ccccccccc[cccc]'	specifies the owner's name or similar identification. The information is specified as character constants, and can be up to 10 bytes in length for EBCDIC and BCD volume labels, or up to 14 bytes in length for ANS volume labels.
OWNERID=	name	specifies a one- to ten-character name or other identifying information. OWNERID is specified as an EBCDIC character string with the exclusion of the blank and the comma characters.
	xxxxxxxxxx	specifies a one- to ten-character field that identifies the owner of the volume.
PASSES=	n	specifies the number of passes to be made in analyzing a recording surface.
	0	specifies that the ANALYZE function is to bypass all surface analysis and track formatting, writing only a a VTOC, track zero records (IPL bootstrap and volume label records), and IPL text if requested.
PASSWORD		specifies that password protected data sets contained in the group are to be included in the operation.
PASSWORD1=	current-password	specifies the current password in the entry to be included in the operation.
PASSWORD2=	new-password	specifies the new password to be assigned to the entry. The password can consist of one- to eight-alphameric characters.
PDS=	name	specifies the fully qualified name of the partitioned data set to be moved or copied.
PICTURE=	B'decimal number'	specifies a decimal number that is to be converted to binary and right-aligned in the defined field.
	'character string'	specifies an EBCDIC character string that is to be placed in the defined field or applicable records.
	length	specifies the number of bytes the picture will occupy.
	p'decimal number'	specifies a decimal number that is to be converted to packed decimal and right-aligned in the defined field.
	startloc	specifies a starting byte (within any applicable output record) in which the picture is to begin.
PRECOMP=	routinename	specifies the symbolic name of a routine that processes logical records (physical blocks in the case of VS or VBS records longer than 32K bytes) from either or both of the input data sets before they are compared.
PREFORM=	A	specifies that an ASA control character is provided as the first character of each record to be printed or punched.

Definition of Operands (cont'd)

PREFORM= (cont'd)	M	specifies that a machine-code control character is provided as the first character of each record to be printed or punched.
PURGE		specifies that each data set specified by DSNAME or VTOC be scratched, even if its expiration data has not elapsed.
PURGE=	YES	indicates that all unexpired data sets on the volume can be overwritten provided that the operator signals his concurrence when the first unexpired data set is encountered, or that the program may be written over any user labels, or over any data that follows the volume label record.
	NO	specifies that the operation is to be terminated if an unexpired data set is encountered, or specifies that the program may not be written over standard user labels.
QUANTITY=	number number[,CREATE=number]	specifies the number of records that this statement is to generate (each record is specified by the other parameters), or specifies the number of times the defined group of CREATE statements is to be used repetitively. CREATE specifies the number of following CREATE statements to be included in the group.
RANGE=	number	specifies an absolute value which the contents of this field can never exceed.
RENAME=	name	specifies that the data set is to be renamed, and indicates the new name.
REPLACE=	X'xx'	specifies the hexadecimal representation of the character to be used by IEBTCRIN to replace error bytes.
SEQFLD=	ddl	specifies, in decimal, the starting column (up to column 80) and length (8 or less) of sequence numbers within existing logical records and subsequent Data statements.
SEQ1=	cccccccc	specifies the sequence number of the first logical record to be renumbered or deleted.
	ALL	specifies a renumbering operation for the entire member or data set.
SEQ2=	cccccccc	specifies the sequence number of the last logical record to be renumbered or deleted.
SER=	xxxxxx	specifies the volume serial number of the first or only tape to be labeled.
SIGN=	sign	specifies a mathematical sign (+ or -), which is used when defining a packed-decimal or binary field.
SKIP=	n	specifies that every nth record is to be printed or punched.

Definition of Operands (cont'd)

SOURCE=	x	specifies user modifications when the x value is 0, or IBM modifications when the x value is 1. This parameter is valid only when a member of a partitioned data set is being processed.
SSI=	hhhhhhhh	specifies eight hexadecimal characters of system status information to be placed in the directory of the new master data set as four packed hexadecimal bytes of user data.
START=	jobname	specifies the name of the input job to which the EDIT statement applies.
STARTLOC=	starting-byte location	specifies a starting location (within all output records using this field) in which a field is to begin.
STEPNAME=	name	specifies the first job step to be placed in the output data set when coded with TYPE=POSITION. Job steps preceding this step are not copied to the output data set. When coded with TYPE=INCLUDE or TYPE=EXCLUDE, STEPNAME specifies the names of job steps that are to be included in, or excluded from, the operation.
STOPAFT=	n	specifies, for sequential data sets, the number of logical records to be punched or printed. For partitioned data sets, this specifies the number of logical records to be punched or printed in each member to be processed.
STRTADR=	nnnnn	specifies the one- to five-byte track address, relative to the beginning of the volume, at which the VTOC is to begin.
STRTAFT=	n	specifies, for sequential data sets, the number of logical records to be skipped before printing or punching begins. For partitioned data sets, STRTAFT=n specifies the number of logical records to be skipped in each member before printing or punching begins.
SYS		specifies that data sets that are to be scratched have names that begin with "AAAAAAAA.AAAAAAAA.AAAAAAAA.AAAAAAAA." or "SYSnnnnn.T" and "F" or "V" in position 19. These names are assigned to data sets by the operating system.
TO=	device= list	specifies the volume(s) to which the data set(s) is to be moved or copied.
TO=	device= serial	specifies the device type and volume serial number of the volume to which the partitioned data set or cataloged entries are to be moved or copied.
TOADDR=	cuu	specifies the channel number, c, and unit number, uu, of the message output or receiving device.
TODD=	cuu (cuu,...)	specifies the channel and unit address of the direct access device containing the volume to be processed.
	ddname (ddname,...)	specifies the ddname of a DD statement defining the device that contains the volume to be processed.
TODEV=	xxxx	specifies the type of output or receiving device, for example, 2400.

Definition of Operands (cont'd)

TOTAL=	routinename	specifies the name of the user's totaling routine.
	size	specifies the number of bytes required for the user's data.
TOTCONV=	PZ	specifies that data (packed decimal mode) is to be converted to unpacked decimal mode.
	XE	specifies that data is to be printed or punched in 2-character-per-byte hexadecimal representation (for example C3,40,F4,F6).
TRACK=	ccchhhh	specifies the address of the track for which an alternate is requested, where cccc is the cylinder number and hhhh is the head number.
TRANS=	name	specifies a user-translate table to be used by IEBTCRIN.
	NOTRAN	specifies that no translation and no special processing is to be performed.
	STDLC	specifies that the MTST code is to be translated to standard EBCDIC, alphabetic characters are translated as lowercase.
	STDUC	specifies that the MTST code is to be translated to standard EBCDIC; alphabetic characters are translated to uppercase.
TYPE=	code	specifies the protection code of the password and, if a control password is being assigned to a direct access, online data set, specifies the protection status of the data set.
	EXCLUDE	specifies that the output data set is to contain a JOB statement and all job steps belonging to the job except those steps specified in the STEPNAME parameter.
	INCLUDE	specifies that the output data set is to contain a JOB statement and all job steps specified in the STEPNAME parameter.
	MTDI	specifies that the input was created on a Magnetic Data Inscriber.
	MTST	specifies that the input data was created on a Magnetic Tape Selectric Typewriter.
	POSITION	specifies that the output is to consist of a JOB statement, the job step specified in the STEPNAME parameter, and all steps that follow it.
TYPORG=	PO	specifies that the input data sets are partitioned data sets.
	PS	specifies that the input data sets are sequential data sets.
UCS	image-id	specifies UCS image-id for the 3203-4 or 3211 printer.
UNCATLG		specifies that the catalog entry pertaining to the data set is to be removed.
UPDATE	INPLACE	specifies the old master data set is to be updated within the space it actually occupies. The old master data set must reside on a direct access device. UPDATE is valid only when coded with CHANGE.

Definition of Operands (cont'd)

VERCHK=	NOCHK	specifies that no record verification check is to be made.
	VOKCHK	specifies that a record verification check is to be made.
VOL=	device= list	specifies the volume or volumes that contain the data set to be processed.
	device= serial	specifies the device type and volume serial number of the volume to be processed.
VOLID=	SCRATCH	specifies that no volume serial number check is to be made.
	serial serial[,serial]...	specifies the volume serial number of the volume to be processed.
VOLPASS=	0	specifies that the volume is not security protected. If VOLPASS is omitted, 0 is assumed.
	1	specifies that the volume is security protected.
VOLUME=	device= serial	specifies the device type and volume serial number of the source volume.
VTOC		specifies that all data sets on the specified volume, except those protected by a password or those whose expiration dates have not yet expired, are to be scratched.
VTOC=	xxxxx	specifies a one- to five-byte decimal relative track address representing a primary track on which the volume table of contents is to begin. The VTOC cannot occupy track 0.

Service Aids-Reference Summary

OS/VS1 Service Aids Reference Summary

This summary is a quick guide to the major or most-used functions of the service aids. The JCL and control statement examples are valid as shown, but they represent only the simplest form of the programs they illustrate. For complete information about control statement parameters and other details pertaining to performance, refer to OS/VS1 Service Aids, GC28-0665; OS/VS1 SYS1.LOGREC Error Recording, GC28-0668; and OS/VS System Modification Program, GC28-0673.

HMDSADMP

Program Type: Stand-alone program, supplied as a macro definition in the system macro library SYS1.MACLIB.

Function: Produces high-speed or low-speed dump of real storage. The high-speed version can also dump the page data sets.

Invoked by: Performing IPL procedure on stand-alone dump program, created by assembling the macro instruction and executing the resulting job stream to initialize the dump program on a residence volume.

When SADMP begins processing, it may begin to send messages; you must reply to these messages before processing can continue. (Note that if the console is unavailable, the dump program bypasses operator communication and attempts to dump real storage to the unit address specified in the HMDSADMP macro instruction).

Controlled by: Responses to prompting messages and variations in coding the macro instruction, whose simplest form is:

```
HMDSADMP   IPL=resvolume,TYPE=speed,OUTPUT=device
```

The basic variations of the macro parameters are:

```
HMDSADMP   TYPE= { HI }, IPL= { Tunit } ,OUTPUT= { Tunit }
                 { LO }       { Dunit }          { Punit }
```

Some of the prompting messages that you will encounter are :

HMD001A TAPE= or HMD001A PTR= , which requires you to specify an output device or, by pressing EOB, to accept the device specified in the macro instruction.

HMD011A TITLE= , which permits up to 100 characters of title; EOB produces an untitled dump.

HMD012D ENTER Y OR N FOR PAGEDUMP=
Replying Y (for "yes") dumps the page data set; N terminates the program.

HMD021A PAGE DEVICE ADDRESS = , which requests the address of the page device that you want dumped.

HMD024D ENTER Y OR N - PAGE DUMP CONTINUE
Replying Y repeats the prompting messages, to obtain a dump of another page data set. Replying N terminates SADMP execution.

For information about other parameters in the macro instruction, see OS/VS1 Service Aids.

Output: Unformatted hexadecimal dump written to tape or printer. Use HMDPRDMP to print high-speed tape output, IEBPTPCH to print low-speed tape output.

Service Aids (cont'd)

GTF (Generalized Trace Facility)

Program Type: Standard feature of CS/VS.

Function: Traces all or selected system events, such as I/O interruptions, SIO operations, etc.

Invoked by: START command, such as the following:

```
START    GTF.Pn,outputdevice, outputvolume, (MODE=EXT)
                    or
START    GTF.Pn,,,(MODE=INT)
```

Controlled by:

1. START options, specified either as PARM = values on the EXEC DD statement in the started procedure, or within parentheses as the fourth parameter on the START command.
   The START options are:

   MODE= { INT | EXT }
   - INT — requests the trace data be kept in virtual storage buffers.
   - EXT — requests that the trace data be written to the data set defined by the IEFRDER DD statement in the started procedure.

   DEBUG= { YES | NO }
   - YES — requests that all error conditions encountered by GTF result in termination of GTF.
   - NO — requests that GTF attempt to recover from all errors.

   TIME= { YES | NO }
   - YES — requests that each trace record produced by GTF contain a time stamp.
   - NO — requests that no time stamp be recorded on each record.

   BUF= nn — specifies the number of buffers to be used for recording trace data. The allowable range is 1-255. The defaults are: if MODE=EXT, GTF obtains 3 buffers; if MODE=INT, GTF obtains 4 buffers.

2. TRACE options, specified in response to prompting message HHL100A. The TRACE options are:

   - DSP - record created whenever an SRB, LSR routine, or TASK is dispatched. Comprehensive unless SYSM specified also.
   - EXT - comprehensive recording of all external interrupts. Ignored if SYS, SYSM, SYSP specified also.
   - PCI - comprehensive recording of all program-controlled I/C interrupts. IO, IOP, SYS, SYSM, or SYSP must also be specified.
   - PI - comprehensive recording for all program interrupts.
   - PIP - causes prompting for specific program interrupts identified by their specific code. Ignored if SYS, SYSM, or SYSP also specified.
   - RNIO - comprehensive record of VTAM remote network activity. Ignore if SYS, SYSM, or SYSP is specified. To request minimal external recording, specify TRACE=SYSM,RNIO.
   - SIO - comprehensive recording for system SIO operations for all devices.
   - SIOP - causes prompting for specific devices for which system SIO operations should be traced. Ignored if SYS, SYSM, or SYSP specified also.
   - SVC - comprehensive recording for all SVC interrupts.
   - SVCP - causes prompting for specific SVC interrupts by number. Ignored if SYS, SYSM, or SYSP also specified.
   - SYS - comprehensive recording for I/O, SIO, SVC, program, and external interrupts and for recovery routines. If additional event keywords are specified, they also result in comprehensive recording. Do not specify IO, SIO, RNIO, SVC, PI, or EXT with SYS.

SYSM- same as SYS except that minimal trace records result. Correspondingly, additional keywords also result in minimal recording. Do not specify IO, SIO, SVC, PI, or EXT with SYSM. Specify SYSM,RNIO to record VTAM remote network activity in minimal format.

SYSP- causes prompting for specific SVC, IO, SIO, and PI events to be recorded during execution. Comprehensive records result for these events as well as for any additional event keywords that are specified at the same time. Do not specify SVC, IO, SIO, RNIO, or PI with SYSP.

TRC - record created for events being traced that are associated with GTF. If not specified, these events are not recorded.

USR - recording of all data passed to GTF via GTRACE macro.

IO - comprehensive recording for all non-PCI IO interrupts.

IOP - causes prompting for specific devices for which IO interrupts will be recorded. Ignored if SYS, SYSM, or SYSP also specified.

Here is an example of trace options specified correctly:

```
TRACE=SYSP,DSP,PCI,TRC
```

Output: For MODE=EXT, a trace data set residing on an external device. For MODE=INT, trace buffers residing in main storage. Use the EDIT function of HMDPRDMP to format and print trace data from a core image dump or a trace data set.

HMBLIST

Program Type: OS/VS problem program.

Function: Formats and prints object modules and load modules.

Lists CSECT identification records. Maps reenterable load module area.

Invoked by: JCL, such as the following:

```
//MYJOB     JOB   MSGLEVEL=(1,1)
//STEP      EXEC  PGM=HMBLIST
//INPUT     DD    DSN=libname,DISP=OLD
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
      Control Statements
/*
```

Controlled by: Control statements, as follows:

```
LISTLOAD    [OUTPUT=type][,TITLE=('title')]
            [,DDN=inputddname][,MEMBER=membername)]
            [,RELOC=hhhhhh]
```

Requests load module processing. OUTPUT=parameter requests either MODLIST (formatted listing plus ESD and RLD records), XREF (module map and cross-reference listing), or BOTH. RELOC allows you to specify a relocation or base address of up to six hexadecimal characters.

```
LISTOBJ     [TITLE=('title')][,DDN=inputddname]
             [,MEMBER=membername]
```

Requests listing of object modules.

```
LISTIDR     [OUTPUT=type][,TITLE=('title')]
             [,DDN=inputddname][,MEMBER=membername][,MODLIB]
```

Requests listing of a load module's CSECT identification records. OUTPUT=parameter requests either IDENT (only records containing SPZAP or user-supplied data) or ALL. MODLIB (valid only when DDN is present) requests condensed and formatted output; no module summary is printed.

```
LISTLPA
```

Requests mapping of the reenterable load module area.

Note that for LISTLOAD, LISTOBJ, and LISTIDR, if you omit the DDN=parameter, LIST assumes a default ddname of SYSLIB.

Output: Separate listing for each control statement.

Service Aids (cont'd)

IMCOSJQD

Program Type: OS/VS problem program, residing in SYS1.LINKLIB.

Function: Formats and prints all or selected records in the system job queue (SYS1.SYSJOBQE data set), in the scheduler work area data set (SWADS), in the incore joblist (JOBLIST), and in the system scheduler work area data set (SYS1.SYSWADS).

Invoked by: JCL such as the following:

```
//DUMP        JOB              MSGLEVEL=(1,1)
//              EXEC             PGM=IMCOSJQD
//OSJQDIN     DD               DSNAME=SYS1.SYSWADS,
//          UNIT=2314,VOL=SER=111111,DISP=SHR
//OSJQDOUT    DD               UNIT=2400,DISP=(NEW,KEEP),
//          DSNAME=QUEUEOUT,LABEL=(,NL)
//SYSPRINT    DD               SYSOUT=A
[//SYSIN      DD               *]
              .
              .
              .
/*
```

Controlled by: Control statements in the JCL or by commands entered in response to prompting messages if the program is started from the console or if the //SYSIN DD card is omitted. The control statements are the same as for IMCJOBQD.

Output: Complete or selected contents of JOBLIST, SWADS, SYS1.SYSJOBQE; complete contents of SYS1.SYSWADS - written to printer or magnetic tape.

IFCDIP00

Program Type: OS/VS problem program.

Function: Reinitialize or reallocate the SYS1.LOGREC data set.

Invoked by: JCL, such as the following:

```
//MYJOB       JOB   MSGLEVEL=(1,1)
//STEP        EXEC  PGM=IFCDIP00
//SERERDS     DD    DSNAME=SYS1.LOGREC,UNIT=2314,
//                  DISP=(OLD,KEEP),VOL=SER=111111
/*
```

Controlled by: Variations in the SERERDS DD statement. For complete information, see OS/VS1 SYS1.LOGREC Error Recording.

Output: Initialized or reallocated SYS1.LOGREC data set.

HMDPRDMP

Program Type: OS/VS problem program.

Function: Formats and prints dump data sets, including page data sets, and GTF trace data.

Invoked by: JCL, such as the following:

```
//MYJOB       JOB     MSGLEVEL=(1,1)
//STEP        EXEC    PGM=HMDPRDMP
//TAPE        DD      DSNAME=dsname,VOL=SER=volser,UNIT=ddd,
//            [LABEL=(label)],DISP=OLD
//PRINTER     DD      SYSOUT=A
//SYSPRINT    DD      SYSOUT=A
[//SYSUT1     DD      UNIT=ddd,SPACE=(2056,(n,10))]
[//SYSUT2     DD      UNIT=2400,VOL=SER=DUMP,LABEL=(,NL),]
[//            DISP=NEW]
[//SYSIN      DD      *]
        [Control Statements]
/*
```

(For a high-density dump on the 3800, use:
//PRINTER DD ...CHARS=DUMP[,FCB=STD3]
CHARS=DUMP specified high-density dump;
FCB=STD3 specifies printing at 8 lines per inch, 80 per page.)

Service Aids (cont'd)

The EXEC statement invokes HMDPRDMP. The DD statements shown are used as follows:

PRINTER - defines the PRDMP output data set.

SYSPRINT - optional, defines the PRDMP message data set.

SYSUT1 - required when input data set contains a dump and resides on a direct access device, or when input spans more than one volume. Optional, but enhances performance when used with an input dump data set that resides on a single tape.

SYSUT2 - required clearing the SYS1.DUMP data set. Never use SYSUT1 and SYSUT2 in the same step.

SYSIN - required when entering control statements in the input stream.

TAPE/anyname - input data set dump or GTF trace data set.

For details about any of the parameters used in this example, see OS/VS1 Service Aids.

Controlled by: Control statements, entered either on cards in the input stream or in reply to prompting messages. They are:

CVT=pointer - specifies the location of the CVT.

SEGTAB=pointer - specifies the location of the segment table.

NEWDUMP DDNAME=ddname - defines the input data set.

NEWTAPE - requests that a new tape volume containing another input data set be mounted on the tape device defined by TAPE DD statement.

ONGO value - establishes a preset combination of control statements to be requested in subsequent uses of the GO control statement.

GO - requests a preset combination of control statements. If not preceded by the ONGO statement, requests default combination of QCBTRACE, LPAMAP, FORMAT, EDIT, and PRINT ALL.

TITLE text - requests a specific dump title.

QCBTRACE - requests a trace of QCBs in the input dump data set.

LPAMAP - requests a map of the reenterable load module area.

FORMAT - requests formatting of major system data areas.

PRINT value - requests printing of parts of the input data set, as dictated by the subparameters, which are:

ALL - allocated storage.

CURRENT - storage associated with current task.

NUCLEUS - resident nucleus, system queue area, and pageable nucleus.

STORAGE=(addresses) - virtual storage in the address range supplied by beginning and ending hexadecimal address pairs.

JOBNAME=(jobnames) - areas in virtual storage associated with specified jobs.

F03 - areas of virtual storage associated with tasks terminated by DAR.

REAL=(addresses) - real storage in the address ranges supplied by beginning and ending hexadecimal address pairs.

PAGE=cuu=(ttrs) - areas of the page data set defined by device address and beginning and ending TTR address pairs.

EDIT value - requests formatting of all or selected GTF trace records.

END - requests normal termination, or, if no other control statements are specified, requests loading of the data set defined by the SYSUT2 DD statement.

For complete descriptions of the keywords used with the NEWDUMP, ONGO, PRINT, and EDIT control statements, see OS/VS1 Service Aids.

Output: Formatted and printed dump or trace data.

Service Aids (cont'd)

IFCEREP0

Program Type: OS/VS problem program.

Function: Formats and prints selected error records in the SYS1.LOGREC data set.

Invoked by: JCL, such as the following:

```
//MYJOB     JOB     MSGLEVEL=(1,1)
//STEP      EXEC    PGM=IFCEREP0,PARM='keyword=value'
//SERLOG    DD      DSNAME=SYS1.LOGREC,DISP=(OLD,KEEP)
//EREPPT    DD      SYSOUT=A
/*
```

Controlled by: Keyword parameters in the PARM= field of EXEC statement. Omitting a keyword causes the corresponding default value to take effect. When all defaults are in effect, IFCEREP0 summarizes, accumulates, and prints all records in the SYS1.LOGREC data set. For complete information, see OS/VS1 SYS1.LOGREC Error Recording.

Output: Listing of summarized and printed records.

HMAPTFLE

Program Type: OS/VS problem program.

Function: Application Function: Generates control statements and invokes linkage editor to apply PTF dynamically.

Generate Function: Generates job stream to apply PTFs or ICRs in a later step.

Invoked by: Application Function: JCL, such as the following:

```
//MYJOB     JOB     MSGLEVEL=(1,1)
//STEP      EXEC    PTFLE,USE='linkageeditor',LIB1=SVCLIB
//PTF.MODE  DD      *
        Control statements
        PTF object deck
        IDENTIFY control statement
/*
```

The cataloged procedure PTFLE contains symbolic parameters whose default values are USE='IEWL' and LIB1=LINKLIB.

For more information, refer to OS/VS Service Aids.

Generate Function: JCL (not invoking a cataloged procedure) such as the following:

```
//GENER     JOB     MSGLEVEL=(1,1)
//STEP      EXEC    PGM=HMAPTFLE
//PRINT     DD      SYSOUT=A
//OUTF      DD      UNIT=2400,LABEL=(,NL),DISP=(NEW,KEEP),
//          VOL=SER=OUTPUT
//PCHF      DD      UNIT=2400,LABEL=(,NL),DISP=OLD,
//          VOL=SER=SYSGEN,DCB=(LRECL=80,BLKSIZE=80)
//MODF      DD      *
        Control statements
/*
```

Controlled by: PTFLE Control statement, whose format is:

```
modulename      SSI     comments
```

The module name must begin in column 1; the SSI number must begin in column 10. Comments may begin in column 19. For the application function, a Linkage Editor IDENTIFY control statement is also required:

```
IDENTIFY    csectname('data')[,csectname(,data')]
```

Output: Application Function: Module updated with applied PTF.

Generate Function: Job stream required to apply PTF in a later step.

Service Aids (cont'd)

HMASPZAP

Program Type: OS/VS problem program.

Function: Inspect and/or modify data in a load module or in a data set on a direct access device; in other words, apply local fix.

Invoked by: JCL, such as the following:

```
//MYJOB     JOB    MSGLEVEL=(1,1)
//STEP      EXEC   PGM=HMASPZAP
//SYSPRINT  DD     SYSOUT=A
//SYSLIB    DD     DSNAME=dsname,DISP=OLD
//SYSIN     DD     *
        [CONSOLE]
    *   [control statements]
/*
```

SPZAP can also be dynamically invoked. For details, see OS/VS1 Service Aids.

Controlled by: Control statements, supplied either as cards in the input stream, or if CONSOLE is present in the input stream, as responses to prompting message HMA116A. The control statements are:

NAME member csect - specifies the CSECT to be updated.

CCHHR address - specifies the address of the field to be updated.

VERIFY offset content - requests comparison between field specified in offset parameter and data supplied in content parameter.

REP offset data - requests replacement of field specified in offset parameter by field supplied in data parameter.

IDRDATA user - provides data to be placed in the user data field of an updated CSECT identification record.

SETSSI xxyynnnn - requests updating SSI data with value supplied in xxyynnnn parameter.

DUMP[T] member csect - requests dumping of a csect (or ALL csects) in the load module specified in the csect parameter. The optional T also requests translation of the dump data.

ABSDUMP[T] value - requests dumping selected records (specify starting and ending track address pairs in the value field), a member (specify membername in the value field), or an entire data set (specify ALL in the value field). The optional T also requests translation of the dump data.

BASE offset - requests adjustment of relative displacements by supplying base address in offset field.

Output: Formatted hexadecimal dump or translated dump (if requested, or if an error occurred), and modified csects.

IMCJOBQD

Program Type: Stand-alone program, supplied as an object module in component library SYS1.ASAMPLIB. Use IEBGENER to punch the module into cards or onto magnetic tape.

Function: Formats and prints all or selected records in the system job queue (SYS1.SYSJOBQE data set), in the scheduler work area data set (SWADS), in the resident joblist (JOBLIST), or in the system scheduler work area data set (SYS1.SYSWADS).

Invoked by: Storing control register 0 and 1 at locations X'1C0' and X'1C4' (if you want full dump of the resident job list or a selective dump of SYS1.SYSJOBQE) and by IPL-ing the object module that contains IMCJOBQD.

Controlled by: Commands entered in response to prompting messages.

In response to IMC000A, press EOB to accept default devices, or enter:

[O=outputdeviceaddress] [,SYSW=syswadsdeviceaddress]

Service Aids (cont'd)

In response to IMC975A, press EOB to terminate IMCJOBQD or enter:

SYSJOBQE - to dump all SYS1.SYSJOBQE and resident joblist.

SYSJOBQE.S - for selective dump of SYS1.SYSJOBQE plus resident joblist.

SYSWADS - to dump all of SYS1.SYSWADS, only.

SYSWADS,jpn1...,jpn4 - for full dump of SYS1.SYSWADS, plus selected SWADS, selected by initiator procedure names.

SWADS=(procname.P1,procname.P2...,procname.Pn) - to dump SWADS for specified initiator procedure names in the specified patitions.

An IMC022A message will direct you to mount the volume containing the data set for which you requested a dump.

In response to IMC001A (when a selective SYS1.SYSJOBQE dump was specified), press EOB to terminate dumping SYS1.SYSJOBQE or enter:

CLASS=inputclass - for jobs of the specified input class.

SYSOUT=outputclass - for jobs of the specified output class.

JOBNAMES=(jn1...,jn4) - for the named jobs, only.

FREE - for the free chain of the job list.

HOLD - for jobs held by jobname.

QID=(qid1...,qid4) - for users with specified QID's, only.

NOTE: QID=(qid1...,qid4) [,HOLD / ,SYSOUT=outputclass] ,and

JOBNAMES=(jn1...,jn4) [,CLASS=inputclass / ,SYSOUT=outputclass] are valid.

Output: Formatted listing of all or selected records.

SERVICE AIDS USE GUIDE

Information Gathering

SADMP

1. Dumps the contents of real or selected portions of virtual storage to a tape, which can be formatted and printed using PRDMP. (Note that SADMP output may also be directed to a printer).

GTF

2. Traces all system events.
3. Traces selected events, such as I/O interruptions, SIO operations, etc.
4. Traces user programs with GTRACE macro instruction.

Patching

PTFLE

5. Generates control statements and JCL needed to apply PTFs, ICRs, or Program Products; the application function also invokes the linkage editor.

SMP (System Modification Program)

6. Provides facilities for applying PTFs or user's modifications to executable systems, and to DLIBs or permanent user libraries. For information, refer to OS/VS System Modification Program (SMP).

SPZAP

7. Modifies data in a load module.
8. Sets traps by inserting invalid instructions or user-written SVCs.
9. Dumps load modules by CSECT to allow examination of the text.
10. Dumps selected data to verify the count, key and contents of the data.

DIP00

11. Reinitializes the SYS1.LOGREC data set if destroyed.

Service Aids (cont'd)

Mapping, Formatting, and Printing

PRDMP

Specify high-density dump on the 3800 by:

```
//PRINTER  DD  ... CHARS=DUMP[,FCB=STD3]
```

12. Formats and prints the following from SADMP high-speed dumps, and SVC dumps.
    a. Link pack area.
    b. Queue control block trace.
    c. Major system data areas.
    d. Selected areas of storage by virtual or real address.
    e. Operating system nucleus.
13. Formats and prints selected records from the GTF trace data set or from trace buffers in a SYS1.DUMP or SADMP output data set. Records are selected by keywords such as JOBNAME, I/O, SVC, or SIO.

LIST

14. List specific object modules, load modules, or load modules in a data set.
15. Maps control sections and overlay structure, and lists cross-references within a load module.
16. Lists CSECT identification records for specific load modules.
17. Lists translation data, linkage editor modification data, or SPZAP modifications to control sections in a load module.
18. Maps reenterable load module area.

EREP0

19. Selects, formats, and prints records, record summaries, and special reports from the SYS1.LOGREC data set, by record type:
20. Selects records by device type, date, or device address.

JOBQD and OSJQD

21. Dumps entire SYS1.SYSJOBQE data set.
22. Selects, formats and prints job queue records associated with a specific job.
23. Selects, formats, and prints job queue records associated with a specific work queue.

ISDASDA0

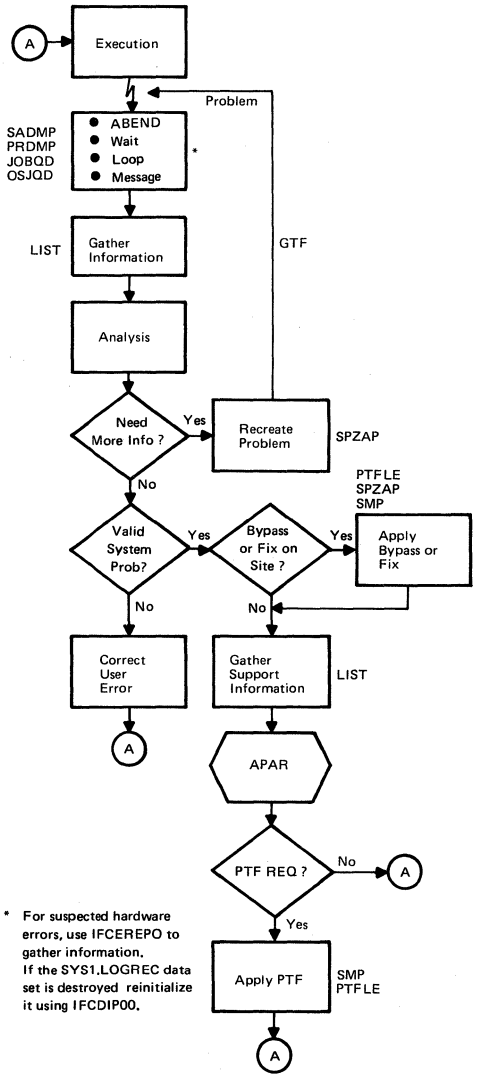
24. Selects records from the SYS1.LOGREC data set, the EREP data set, or both, and prints cartridge error statistics and usage and error information pertaining to devices in the Mass Storage System. For information on SDA0, refer to OS/VS1 SYS1.LOGREC Error Recording.

NOTES:

The numbers in this table refer to the explanatory notes on the preceding page. For each symptom, read from left to right to find out which functions of these service aids you should use to diagnose and fix the problem. For complete information about IFCDIPP00, IFCEREPO, and ISDASDA0, see *OS/VS1 SYS1.LOGREC Error Recording.* For information about the other service aids, see *OS/VS1 Service Aids,* and *OS/VS System Modification Program (SMP).*

	INFORMATION GATHERING		PATCHING			MAPPING, FORMATTING, AND PRINTING				
SYMPTOM	SADMP	GTF	SMP PTFLE	SPZAP	DIP00	PRDMP	LIST	EREP0	JOBQD OSJQD	SDA0
Warm Start Failure	1	–	–	8	–	12c-e	–	–	21	–
Scheduler ABEND	–	2	–	8	–	13	14,15	–	21	–
Writer ABEND	–	2	–	8	–	13	–	–	21	–
Problem Program ABEND	–	4	–	8	–	13	15	–	–	–
Recursive ABEND	1	2	–	8	–	12a,c-d,13	15,18	–	22	–
Disabled Loop	1	2	–	–	–	12c-e,13	–	–	–	–
Problem Program Loop	–	4	–	–	–	6	15	–	–	–
Large Loop with I/O	1	2	–	8	–	12a,c-e,13b-d	18	–	–	–
DAR Loop (VS1 Only)	1	2	–	–	11	12c&e,13	15	19	–	–
Hard Wait	1	2	–	–	11	12c-e	15	19	–	–
Enabled Wait	1	2	–	–	11	12b,13	15	19	–	–
Reader/Interpreter Failure	–	–	–	8	–	–	–	–	21	–
I/O Failure (e.g. console)	1	3	–	8,10	–	12a-e,13b-d	18	19,20	–	24
Allocation Failure	1	–	–	8	–	12b-d	15	–	–	–
Enqueued Job Lost	–	3	–	–	–	12a&c-e,13	–	–	23	–
Chain Scheduling Problem	1	3	–	–	–	–	18	–	–	–
Access Method Failure	–	3	–	8	–	13	–	20	–	–
Data Management Prgm Check	–	2,4	–	–	–	13	15	–	–	–
Module Level Unknown	–	–	–	9	–	–	16	–	–	–
User Modification Unknown	–	–	–	9	–	–	17	–	–	–
Applying Program Products	–	–	5	–	–	–	–	–	–	–
Applying PTF	–	–	5,6	–	–	–	–	–	–	–
Applying ICR	–	–	5,6	–	–	–	–	–	–	–
Applying Local Fix	–	–	6	–	–	–	–	–	–	–
APAR Documentation	1	2,4	–	9	–	12a&c-e, 13	16,18	–	21	–
Print SYS1.DUMP	–	–	–	–	–	12b-d,13	–	–	–	–
Capturing System before RE-IPL	1	–	–	–	–	12a-e,13	18	–	–	–

Service Aids (cont'd)



* For suspected hardware errors, use IFCEREPO to gather information. If the SYS1.LOGREC data set is destroyed reinitialize it using IFCDIP00.

SMP Reference Summary

Syntax Format

Write all uppercase letters as indicated.

Write all special characters as indicated.

You can use blanks freely in an SMP statement. Place at least one blank between each keyword (those in all uppercase letters), and between each keyword and associated parameters.

Words in lowercase letters represent statement parameters. An explanation of the parameters appears with each control statement.

Parameters that may be reiterated are identified by a vertical line branching up and around them.

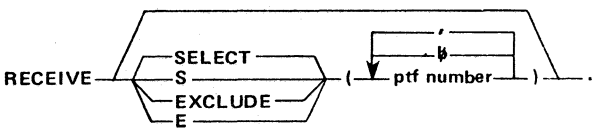
A split in the horizontal path identifies either an optional parameter or a choice in parameters at that point in the statement.

Direction of flow is left-to-right unless a reiteration is indicated; a vertical line and an arrow indicate reiteration.

SMP Control Statements

RECEIVE

The RECEIVE statement causes SMP to read and verify an input file of modifications. Only one RECEIVE statement can be included in an SMP job step. Each PTF or user modification to be handled by SMP must be processed with a RECEIVE statement before any other SMP control statement can be used.



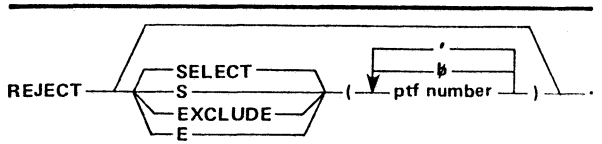
No operands specified means all modifications on the file are processed.

SELECT (ptf number)
specifies the modifications that are to be processed from the input modification file. SELECT can be coded as S.

EXCLUDE (ptf number)
specifies the PTFs or user modifications that are not to be processed from the input file. All others will be processed. EXCLUDE can be coded as E.

REJECT

The REJECT statement causes any PTFs or modifications that have been processed only by the RECEIVE statement to be deleted. More than one REJECT statement can be included in an SMP job.

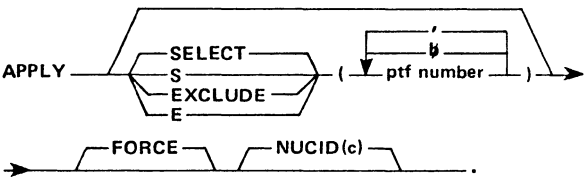


SELECT (ptf number)
specifies the PTFs or user modifications that are to be deleted. SELECT can be coded as S.

EXCLUDE (ptf number)
specifies the PTFs or user modifications that are not to be deleted. All others will be deleted. EXCLUDE can be coded as E.

APPLY

The APPLY statement causes PTFs and user modifications processed by a RECEIVE statement to be put into the operating system libraries. (This statement does not cause any changes in the distribution – DLIB – or permanent user libraries.) There can be any number of APPLY statements in an SMP job.



If no operand is specified, all PTFs and user modifications that have not been processed by an APPLY or ACCEPT statement and have not had the force indicator turned on during the RECEIVE processing will be put into the system.

SELECT (ptf number)
specifies the PTFs or user modifications that are to be put in the operating system. SELECT can be coded as S.

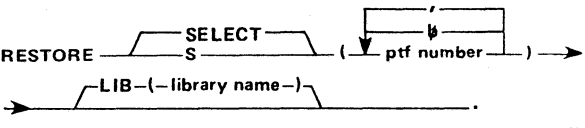
EXCLUDE (ptf number)
specified the PTFs or user modifications that are not to be put in the operating system. All others will be processed. EXCLUDE can be coded as E.

FORCE
must be specified if you want the PTF and user modifications that have the FORCE indicator set to be put in the operating system.

NUCID (n)
n is a numeral that is put on the end of IEANUC0 to form a name under which the current nucleus is saved when the nucleus is to be updated by this APPLY statement.

RESTORE

The RESTORE statement causes the PTFs or user modifications processed by the APPLY statement to be removed from the operating system. The versions of the modules that were in use before the change are put into the operating system libraries. Saved copies of macro definitions on the macro library and any required modules are reassembled and are put into the operating system libraries.

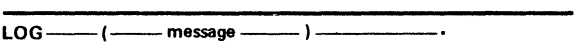


SELECT (ptf number)
specifies the PTFs or user modifications that are to be removed from the system. SELECT can be coded as S.

LIB(library name)
specifies the ddname of the permanent user library to be used. This is specified if the DLIBs do not contain the data for the correct module level to be used in restoring the operating system libraries.

LOG

This statement is used to put messages out on LOG.

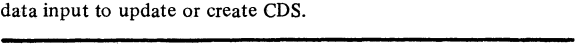


message
is written on LOG along with the time and date that the message was generated. Any character can be used between the parentheses. If parentheses are used, they must be balanced.

Messages generated by the various SMP routines will also be on LOG along with any you generate.

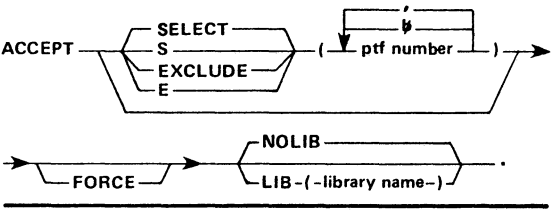
JCLIN

The JCLIN statement causes SMP to read in the Stage I output from system generation or similar data. This output is used as data input to update or create CDS.



ACCEPT

The ACCEPT statement causes the modifications processed by an APPLY statement to be put into the DLIBs or a permanent user library. There can be any number of ACCEPT statements in an SMP job step. Once this statement has processed a modification, SMP cannot remove it from your system.



If no operand is specified, all the PTFs or user modifications that have been processed by an APPLY and not an ACCEPT function will be put into the DLIBs or permanent user libraries.

SELECT (ptf number)
specifies the PTFs or user modifications that are to be put in the DLIBs or a permanent library specified in the LIB keyword. SELECT can be coded as S.

EXCLUDE (ptf number)
specifies the PTFs or user modifications that are not to be put in the DLIB or permanent library. All others will be processed. EXCLUDE can be coded as E.

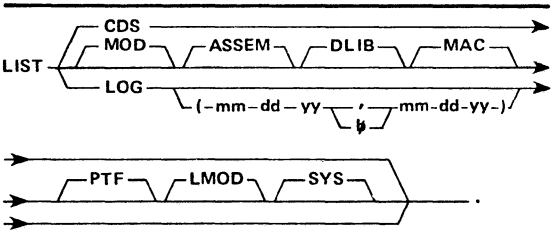
FORCE
specifies that all PTFs or user modifications that have been processed by the APPLY or RECEIVE statement are to be put into the DLIBs or the permanent library.

LIB(library name)
specifies the permanent user library to be used instead of the DLIBs for the modifications specified in the ACCEPT statement in which this keyword is coded.

NOLIB
specifies that regular ACCEPT processing will occur but that no update will be made to the permanent libraries. To zap assembled modules, use the NOLIB keyword because the module does not exist on a DLIB data set.

LIST

This statement causes a listing to be put out that contains data from the history log data set (LOG) or the control data set (CDS).



CDS
specifies that all members in CDS will be printed. If the ACDS is to be listed, the SMPCDS DD statement must point to the ACDS.

option
specifies the type of CDS members to be printed. Any combination of the following can be coded:

MOD	LMOD	PTF	DLIB
MAC	ASSEM	SYS	

LOG[(mm dd yy {b|,} mm dd yy)]
specifies that the contents of LOG are to be printed. One data range may be specified as: mm is the month, dd is the day, and yy is the year. Specify mm as 00 through 12, dd as 00 through 31, and yy as 00 through 99. Separate mm, dd, and yy with blanks.

UCLIN

The UCLIN statement is used to identify the statements immediately following it as update control language (UCL) statements.

UCLIN ——————·

ENDUCL

The ENDUCL statement is used to identify the end of the update control language (UCL) statements. It must immediately follow the last UCL statement.

ENDUCL ——————·

Data Set Requirements Chart

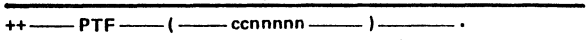
	RECEIVE	APPLY	ACCEPT	REJECT	RESTORE	LOG	JCLIN	UCLIN	LIST
SYSPRINT	X	X	X		X				
SMPOUT	X	X	X	X	X	X	X	X	X
SMPLOG	X	X	X	X	X	X	X	X	X
SMPCDS	X	X	X	X	X	X	X	X	X
SMPACDS	X		X						
SMPPTS	X	X	X	X	X				
SYSUT1	X	X	X		X		X		
SYSUT2	X	X	X		X				
SYSUT3	X	X	X		X				
SYSLIB	X				X				
SMPMACn	X		X	X	X				
SMPPTFIN	X								
SMPCNTL	X	X	X	X	X	X	X	X	X
SMPJCLIN							X		
tgtlib		X			X				
distlib			X		X				
lklib[1]		X	X						
txlib[1]	X	X	X						

[1] Required only if specified on the ++MOD or ++MAC statements.

Modification Control Statements

PTF Header Statement

This header statement indicates the beginning of a PTF or modification and must be present. This statement must be the first one for each PTF or modification.

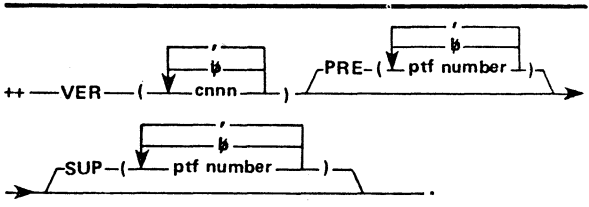


cc
modification identification. Each IBM modification will start with 'U', 'A', or 'I'; these letters are restricted for IBM use only.

nnnnn
is a five-digit number unique for each modification.

Verification Statement

This statement causes system and release level checking to be done. If this statement is not present, no checking is done. More than one verification statement can be in a PTF or modification. Only one verification statement has to pass the checking process to allow the PTF or user modification to be put in the system.



c
is the system code; S=OS, X=VS1, Y=VS2 (SVS), Z=VS2 (MV)

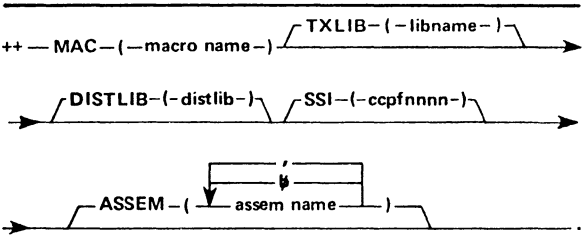
nnn
is the release number; for example, release 1 is coded as 010, release 21.6 is coded as 216 and so on.

PRE (ptf number)
specifies a prerequisite PTF or user modification number. That is the PTF or user modification specified must be in the system before the PTF or user modification being processed can be put in the system.

SUP (ptf number)
specifies a PTF or user modification that is superceded by the one being processed.

Macro Header Statement

This statement describes one total macro replacement and must immediately precede the macro definition replacement if it is in this input stream. If it is an inner macro definition that does not appear during system generation, the module(s) to be reassembled must be specified in this statement. If the replacement macro definition is in a text library that is a partitioned data set and the replacement does not immediately follow, the text library name must be specified.



macroname
is the name of the macro instruction.

TXLIB (libname)
is the name of the text library where the replacement macro is located.

DISTLIB (distlib)
is the name of the lowest-level index name.

SSI (ccpfnnnn)
is the eight-byte hexadecimal system status information.

cc
is the change level of the module. Each release starts as 00. Each change increases it by one.

P
is X'1' indicating a PTF.

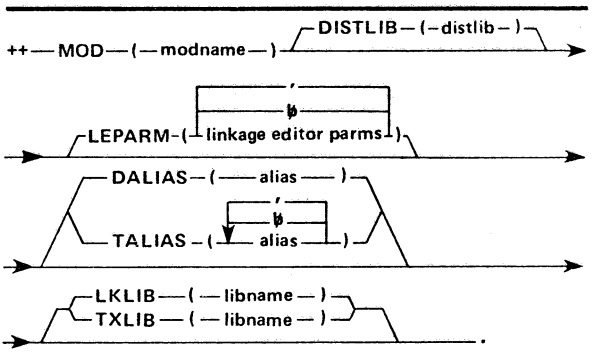
f
is flags.
X'1' IBM
X'5' Critical and IBM.
X'9' Dependency and IBM.

nnnn
are the last four digits of the PTF number.

ASSEM (assem name)
is the name of the module(s) that have to be reassembled; a maximum of 50 can be specified.

Module Header Statement

This statement describes one module that is in a PTF or user modification. It must immediately precede the object code (if there is any to be entered in this input stream). If the module is in a partitioned data set that is a text or link library, the name of the library must be specified. In this case, the object code does not follow.



modname
is the distribution library module name.

DISTLIB (distlib)
is the distribution library ddname. This is only used if the module did not exist when the system generation was done or was part of a total data set copy at system generation.

LEPARM (linkage editor parms)
any of the following linkage editor parameters can be specified for new modules:

RENT	OVLY	NE
REUS	REFR	STD
SCTR	DC	

AC=1 (VS only)
ALIGN 2 (VS only)

Specify STD to indicate that only the standard set (NCAL, SREF, LET, LIST) is to be used.

DALIAS (alias)
is an alias that exists only on a DLIB; there can be one to five aliases specified. This is used to describe modules that are included under an alias name during system generation.

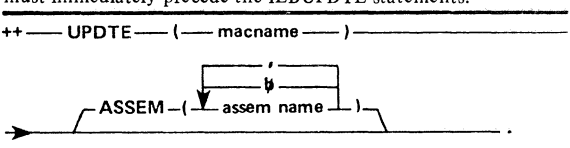
TALIAS (alias)
is an alias that exists on an operating system library and a DLIB for copied modules; from 1 to 5 aliases may be specified.

LKLIB (libname)
is the link library ddname. It contains replacement modules that are in link edited format.

TXLIB (libname)
is the text library ddname. It contains replacement modules that are in object format and have not yet been link edited.

IEBUPDTE Header Statement

This statement describes the IEBUPDTE control statements and must immediately precede the IEBUPDTE statements.

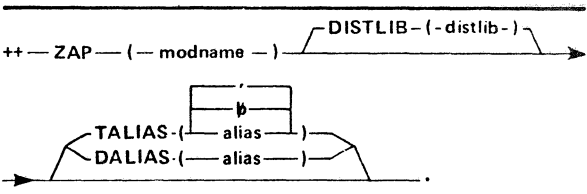


macname
is the name of the macro instruction.

ASSEM (assem name)
is the name of the module(s) that have to be reassembled; a maximum of 50 can be specified.

SPZAP Header Statement

This statement describes SPZAP input. It must immediately precede the SPZAP control cards in the PTF or user modification.



modname
is the distribution library module name.

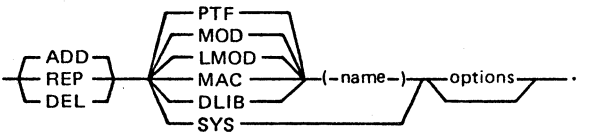
DISTLIB (distlib)
is the distribution library ddname.

TALIAS (alias)
is an alias that exists on an operating system library and a DLIB for copied modules; from 1 to 5 aliases may be specified.

DALIAS (alias)
is an alias that exists only on a DLIB; from 1 to 5 aliases may be specified.

Update Control Language (UCL)

The update control language (UCL) statements provide information that SMP uses to create, change, or delete entries in the CDS. Following the statement format below and using the data from the UCL operand table, you can construct the UCL statements you need.



ADD - Adds data to an existing CDS or ACDS entry or, if none exists, creates an entry.

REP - Replaces a CDS or ACDS entry (or fields within an entry) with the information supplied in the keywords.

DEL - Deletes either specified fields or an entire entry from the CDS or ACDS.

Update Control Language (UCL) Operands

Type	Name	Options	Descriptions
DLIB[1]	(library dsname)		1 to 8 characters — DLIB data set name
		SYSLIB(name[{b / ,} name] . . .)	ddname of DD statement for system library to which the DLIB was copied
LMOD[2]	(load module name)		Operating system load module name 1 to 8 characters
		SYSLIB(name[{b / ,} name] . . .)	System library name, 1 to 8 characters
		AC=1	Authorization code (VS only)
		ALIGN2	Align on 2K boundary (VS only)
		COPYb	Copied at system generation; linkage editor attributes not available
		DCb	Downward-compatible
		NEb	Not editable
		OVLYb	Overlay attribute
		REFRb	Refreshable
		RENTb	Reenterable
		REUSb	Reusable
		SCTRb	Scatter loadable
		STDb	Standard linkage editor attributes only
MAC[1]	(macro name)		1 to 8 characters
		ASSEM(name[{b / ,} name] . . .)	Name of assembled module 1 to 8 characters
		ID(modification identifier)	2 characters
MOD[1]	(module name)[5]		1 to 8 characters — DLIB member name
		DLIB(library name)	DLIB name 1 to 8 characters
		ID(module identifier)	2 alphabetic characters
		LMOD(name[{b / ,} name] . . .)	Operating system load module name 1 to 8 characters

Type	Name	Options	Descriptions
PTF	(ccnnnnn)		Modification number; cc modification identifier, nnnnn unique modification number
		ACPTb	Modification accepted
		APPLb	Modification applied
		REGEN	Modification in DLIB before system generation
		DATE(yyddd)[3]	Date modification received
		MOD(name[{b / ,} name] . . .)[3,5]	DLIB member affected by modification
		MAC(name[{b / ,} name] . . .)[3]	Macro affected by modification
		ZAP(name[{b / ,} name] . . .)[3,5]	DLIB member affected by modification, which is a superzap.
		FORCEb	FORCE indicator
		SUP (ccnnnnn)[3]	Superceding modification number; cc modification identifier, nnnnn unique modification number.
		XZAP(name[{b / ,} name] . ...)[3,5]	Module affected by modification which is an EXPAND with superzap.
SYS			
		SREL(srrr)	s is system type — x=VS1, y=VS2 (SVS), z=VS2 (MVS), s=OS; rrr is release level i.e., 010 = release 1, 216 = release 21.6, etc.
		NUCID(default identifier)[4]	1 numeric character default nucleus identification used to save current nucleus; otherwise current nucleus not saved unless specified in APPLY statement.
		PEMAX (nnn)	Maximum number of PTF elements; default is 50.
		TSOb	Indicates TSO in the system.

[1]To create a new entry specify all the keywords.

[2]ADD or REP cannot use this keyword to add or replace an entire load module type entry. Instead use the JCLIN control statement to obtain LMOD information from the Stage 1 output. ADD or REP can be used to change data in an existing load module entry.

[3]To create a new entry, specify DATE either with SUP or with MOD, MAC, XZAP, or ZAP. Do not code the SUP option with MOD, MAC, XZAP, or ZAP.

[4]This keyword is required when adding the system entry for the first time.

[5]Module name must be the name in the distribution library and not the load module name.

This section lists the publications that support VS1. Applicable TNLs and suffix numbers are not included.

Note: *OS/VS System Generation Introduction,* GC26-3790, which supports VS1 Release 1, is no longer included in this section; content relevant to VS1 is included in *OS/VS1 System Generation Reference,* GC26-3791. *Operator's Library: OS/VS1 RES,* GC38-0330, is removed from the VS1 library. Its content is included in *Operator's Library: OS/VS1 Reference,* GC38-0110.

If additional publication information is required, refer to the *OS/VS1 Release 6 Guide,* GC24-5116 and the *IBM System/370 Bibliography,* GC20-0001.

00 General Information

IBM System/370 Bibliography	GC20-0001-1
IBM System/370 System Summary	GA22-7001-5

03 Printers, Control Units

Introducing the IBM 3800 Printing Subsystem and Its Programming	GC26-3829-2
IBM 3800 Printing Subsystem Programmer's Guide	GC26-3846-1

04 OCR, MCR (Optical, Magnetic Character Readers), Control Units

IBM 3890 Document Processor, Machine and Programming Description	GA24-3612-2

09 Communications Systems or Equipment

Systems Network Architecture General Information	GA27-3102-0

20 Programming Systems - General Information

OS/VS1 Features	GC20-1752-2
OS/VS1 Master Index	GC24-5104-1
OS/VS1 Master Index of Logic	GY24-5164-1

21 Assembler

OS/VS, DOS/VS, and VM/370 Assembler Language	GC33-4010-4
OS/VS and VM/370 Assembler Logic	SY33-8041-1
OS/VS and VM/370 Assembler Programmer's Guide	GC33-4021-3

30 Access Methods, Data Management, Storage/Communications Control Programs

OS/VS1 Access Method Services	GC26-3840-2
OS/VS1 Access Method Services Logic	SY35-0008-1
OS/VS1 BDAM Logic	SY26-3836-0
OS/VS BTAM	GC27-6980-3
OS/VS BTAM Logic	SY27-7246-2
OS/VS1 Catalog Management Logic	SY35-0003-3
OS/VS1 DADSM Logic	SY26-3837-1
OS/VS1 Data Management for System Programmers	GC26-3837-2
OS/VS1 Data Management Macro Instructions	GC26-3872-0** (GC26-3793)
OS/VS1 Data Management Services Guide	GC26-3874-0** (GC26-3783)
OS/VS Graphics Access Method Logic	SY27-7240-0
OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP	GC27-6974-1
OS/VS Graphics Problem-Oriented Routines Logic	SY27-7241-0
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I	GC27-6973-0
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I Logic	SY27-7242-0
OS/VS1 I/O Supervisor Logic	SY24-5156-4
OS/VS1 ISAM Logic	SY26-3838-0

OS/VS Mass Storage Control Table Create	GC35-0013-3
OS/VS Mass Storage Control Table Create Logic	SY35-0016-2
OS/VS Mass Storage Control (MSC) Trace Reports Logic	SY35-0014-1
OS/VS1 Mass Storage System Communicator (MSSC) Logic	SY35-0012-1
OS/VS Message Library: Mass Storage System (MSS) Messages	GC38-1000-2
Operator's Library: IBM 3850 Mass Storage System (MSS) Under OS/VS	GC35-0014-0
OS/VS Mass Storage System (MSS) Planning Guide	GC35-0011-1
OS/VS Mass Storage System (MSS) Services: General Information	GC35-0016*
OS/VS Mass Storage System (MSS) Services Logic	SY35-0015-2
OS/VS Mass Storage System (MSS) Services: Reference Information	GC35-0017-0*
OS/VS1 Open/Close/EOV Logic	SY26-3839-3
OS/VS1 RES RTAM and Workstation Support Logic	SY28-6849-4
OS/VS Message Library: VS1 RES RTAM and Account Messages	GC38-1010-4
OS/VS1 SAM Logic	SY26-3840-2
IBM System/370 Subsystem Support Services Logic	SY30-3017-4
OS/VS Message Library: Subsystem Support Services Messages	GC38-1011-3
IBM System/370 Subsystem Support Services User's Guide	GC30-3022-4
OS/VS Tape Labels	GC26-3795-2
OS/VS TCAM Concepts and Facilities	GC30-2042-0
OS/VS1 TCAM Logic	SY30-2069-1
Operator's Library: OS/VS1 TCAM	GC30-2047-0
OS/VS1 TCAM Programmer's Guide	GC30-2054-2
OS/VS TCAM User's Guide	GC30-2045-1
OS/VS1 Virtual Storage Access Method (VSAM) Logic	SY26-3841-1
OS/VS Virtual Storage Access Method (VSAM) Options for Advanced Applications	GC26-3819-3
Planning for Enhanced VSAM under OS/VS	GC26-3842-1
OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide	GC26-3838-2
VTAM Concepts and Planning	GC27-6998-3
OS/VS1 VTAM Control Block Overview	GX27-0030-1
OS/VS1 VTAM Data Areas	SY27-7266-1
OS/VS1 VTAM Debugging Guide	GC27-0022-1
OS/VS1 VTAM Execution Sequences	SY27-7271-0*
Introduction to VTAM	GC27-6987-5
Introduction to VTAM Logic	SY27-7256-3
OS/VS1 VTAM Logic	SY27-7257-1
VTAM Macro Language Guide	GC27-6994-2
VTAM Macro Language Reference	GC27-6995-4
OS/VS VTAM Network Operating Procedures	GC27-0027-0
OS/VS VTAM Reference Summary	GX27-0034-1*
Supplement to the VTAM Macro Language Guide for the Program Operator	GC27-0036-1
OS/VS1 VTAM System Programmer's Guide	GC27-6996-1
OS Data Management Macro Logic for IBM 1285/1287/1288	GY21-0013-2
OS Data Management Services and Macro Instructions for IBM 1285/1287/1288	GC21-5004-3
OS Data Management Services and Macro Instructions for IBM 1419/1275	GC21-5006-4
OS BSAM Logic for IBM 1419/1275	GY21-0012-2
OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit	GC27-6971-0
OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment)	GC27-6972-0
Introduction to Programming the IBM 3270	GC27-6999-1

OS Programming Support for the IBM 3505 and 3525	GC21-5097-1
OS/VS1 Logic for IBM 3540 Diskette Input/Output Unit	SY24-5166-1
OS/VS1 IBM 3540 Programmer's Reference	GC24-5110-0
IBM 3740 BTAM/TCAM Programmer's Guide	GC21-5071-3
OS/VS 3886 Optical Character Reader, Model 1 Reference	GC24-5101-0

31 Support Programs

OS/VS Linkage Editor and Loader	GC26-3813-4
OS/VS Linkage Editor Logic	SY26-3815-0
OS/VS Loader Logic	SY26-3814-0
OS/VS Message Library: Linkage Editor and Loader Messages	GC38-1007-4

32 Utilities

OS/VS Analysis Program-1 (AP-1) Logic	SY26-3851-0*
OS/VS and DOS/VS Analysis Program-1 (AP-1) User's Guide	GC26-3855-0*
OS/VS Utilities	GC35-0005-4
OS/VS Utilities Logic	SY35-0005-5
OS/VS Message Library: Utilities Messages	GC38-1005-7

34 System Planning, Generation, Installation, SMF; Storage or Performance Estimates; Release Guides

DOS and DOS/VS to OS/MFT, OS/MVT, OS/VS1 Management Planning Guide	GC24-5082-1
DOS and DOS/VS to OS/VS1 Implementation Guide	GC24-5095-2
OS/VS1 Planning and Use Guide	GC24-5090-6
OS/VS1 Release 4 Guide	GC24-5113-0
OS/VS1 Release 5 Guide	GC24-5114-1
OS/VS1 Release 6 Guide	GC24-5116-0*
OS/VS1 Storage Estimates	GC24-5094-6
OS/VS1 System Management Facilities (SMF)	GC24-5115-1
OS/VS1 System Generation Reference	GC26-3791-7

36 Control Program

OS/VS1 Checkpoint/Restart	GC26-3876-0** (GC26-3784)
OS/VS1 Checkpoint/Restart Logic	SY24-5159-3
OS/VS1 System Data Areas	SY28-0605-4
OS/VS1 IPL and NIP Logic	SY24-5160-4
OS/VS1 JCL Reference	GC24-5099-3
OS/VS1 JCL Services	GC24-5100-3
OS/VS1 Job Management Logic, Volume 1	SY24-5168-0
OS/VS1 Job Management Logic, Volume 2	SY24-5169-0
OS/VS1 Programmer's Reference Digest	GC24-5091-5
OS/VS1 Supervisor Logic	SY24-5155-5
OS/VS1 Supervisor Services and Macro Instructions	GC24-5103-2
OS/VS 3886 Optical Character Reader, Model 1 Logic	SY24-5162-0
OS/VS Logic for IBM 3890 Document Processor	SY24-5163-0

37 RAS (Reliability, Availability, Serviceability): Testing, Service Aids, Problem Determination

OS/VS1 Debugging Guide	GC24-5093-4
OS/VS DSS Command Language Reference Summary	GX28-0690-1
OS/VS Dynamic Support System	GC28-0640-1
OS/VS1 Dynamic Support System Logic	SY28-0672-0
OS/VS1 OLTEP	GC28-0666-2
OS/VS1 OLTEP Logic	SY28-0662-3
OS/VS1 Recovery Management Support Logic	SY24-5170-0** (SY27-7239)
OS/VS1 Service Aids	GC28-0665-1
OS/VS1 Service Aids Logic	SY28-0635-2
OS/VS Message Library: VS1 Service Aids and OLTEP Messages	GC23-0005-0** (GC38-1006)
OS/VS1 Service Aids Reference Summary	GX23-0001-1
OS/VS System Modification Program (SMP) System Programmer's Guide	GC28-0673-3
OS/VS System Modification Program (SMP) Logic	SY28-0685-3
OS/VS1 SYS1.LOGREC Error Recording	GC28-0668-2
OS/VS1 SYS1.LOGREC Error Recording Logic	SY28-0669-2
DOS/VS and OS/VS TOLTEP for VTAM	GC28-0663-1
DOS/VS and OS/VS TOLTEP Logic	SY28-0664-1

38 Remote Job Entry (RJE, CRJE, RES, etc.)

OS/MFT, OS/MVT, and OS/VS1 CRJE Concepts and Facilities	GC30-2012-2
OS/MFT, OS/MVT, and OS/VS1 CRJE Logic	GY30-2011-1
Operator's Library: OS/VS1 CRJE	GC38-0335-0
OS/MFT, OS/MVT, and OS/VS1 CRJE System Programmer's Guide	GC30-2016-2
OS/MFT, OS/MVT, and OS/VS1 CRJE Terminal User's Guide	GC30-2014-2
OS/VS1 RES Account Facility Logic	SY28-0660-0
OS/VS1 RES System Programmer's Guide	GC28-6878-4
OS/VS1 RES Workstation User's Guide	GC28-6879-3

40 System Operation (messages, codes)

OS/VS Message Library: VS1 Routing and Descriptor Codes	GC38-1101-0** (GC38-1004)
OS/VS Message Library: VS1 System Codes	GC38-1003-7
OS/VS Message Library: VS1 System Messages	GC38-1001-6
System/370 Operator's Reference Guide	SR20-4460-0
Operator's Library: OS/VS Console Configurations	GC38-0120-5
Operator's Library: OS/VS1 Display Consoles	GC38-0255-3
Operator's Library: OS/VS1 Reference	GC38-0110-7

85 Data Processing - Introductory Manuals

IBM Data Processing Glossary	GC20-1699-4
IBM Marketing Publications KWIC Index	G320-1621-33
Introduction to Virtual Storage in System/370	GR20-4260-1

* These books are new to the publications list, but do not replace other books.

** These books have changed titles and/or order numbers, and replace the order numbers shown in parentheses directly below this order number.

Indexes to OS/VS1 publications are consolidated in the OS/VS1 Master Index. This master index references other publications that contain additional information about the subjects listed here.

READER'S COMMENT FORM

OS/VS1 Programmer's
Reference Digest
GC24-5091-5

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay a response. *For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.* Your comments will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Organized for convenient use?	☐	☐
Complete?	☐	☐

• What is your occupation?- _____

• How do you use this publication:
- As an instructor in a class? ☐
- As a student in a class? ☐
- As a reference manual? ☐

Your comments:

If you would like a reply, please supply your name and address, and check this box ⟶ ☐.
IBM Branch Office serving you _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

From _____

_____

_____

First Class
Permit 170
Endicott
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

GC24-5091-5

OS/VS1 Programmer's Reference Digest (File No. S370-36) Printed in U.S.A. GC24-5091-5

IBM®

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)