GC24-5091-2
File No. S370-36

Systems

OS/VS Programmer's Reference Digest

OS/VS1 Release 2
OS/VS2 Release 1

IBM

PREFACE

This publication contains quick reference information for the experienced programmer. 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 Systems Reference Library publication.

This publication, the *OS/VS2 TSO Command Language Reference Summary*, GX28-0647, the *OS/VS Service Aids Reference Summary*, GX28-0634, may be ordered by specifying *OS/VS Reference Digest Package*, BOF-3200 rather than individual order numbers.

This publication does not contain information about program debugging or control blocks. Refer to the *OS/VS1 Debugging Guide*, GC28-6670, the *OS/VS1 System Data Areas*, SY28-0605, the *OS/VS2 Debugging Guide*, GC28-0632, or the *OS/VS2 System Data Areas*, SY28-0606 to find this information.

Third Edition (April 1973)

This edition applies to Release 2 of OS/VS1, Release 1 of OS/VS2, 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/360 and System/370 Bibliography*, GA22-6822, and the *IBM System/370 Advanced Function Bibliography*, GC20-1763, for the editions that are applicable and current.

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.

SECTIONS 1, 3, 5, 7, 8, or 9 REFLECT ADDED:

- Base Publications Supporting OS/VS1 and OS/VS2
- Code Translation Tables
- General Services Macros
- JCL Statements
- Load Module Control Macros
- Program Interruption Control Macros
- RES Operator Commands
- RES Workstation Commands
- Synchronization Macros
- Task Control Macros
- TCAM Macros
- Termination Macros
- Utility Programs
- Virtual Storage Macros
- VS1 Operator Commands
- VS2 Operator Commands

SECTIONS 2, 3, 4, or 7 REFLECT UPDATED:

- Data Management Macros
- OS/VS1 Completion Code Summary
- OS/VS2 Completion Code Summary
- Programming Conventions for SVC Routines
- Summary of Supervisor Operands
- Supervisor Macro Outlines
- SVC Summary for OS/VS1
- SVC Summary for OS/VS2
- TCAM Devices Supported
- TCAM Macro Operands
- UCB Sense Information

CONTENTS

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 OS/VS System Assembler is in *OS/VS and DOS/VS Assenbler 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

1. EBCDIC graphics shown are standard bit pattern assignments. For specific print train/chain see printer manual.
2. Add C (check bit) for odd or even parity as needed, except as noted.
3. For even parity use CA.

Code Translation Table (con't)

Dec.	Hex	Instruction (RR)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
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
64	40	STH		SP	@	(3)	no punches	0100 0000
65	41	LA			A		12-0-1-9	0100 0001
66	42	STC			B		12-0-2-9	0100 0010
67	43	IC			C		12-0-3-9	0100 0011
68	44	EX			D		12-0-4-9	0100 0100
69	45	BAL			E		12-0-5-9	0100 0101
70	46	BCT			F		12-0-6-9	0100 0110
71	47	BC			G		12-0-7-9	0100 0111
72	48	LH			H		12-0-8-9	0100 1000
73	49	CH			I		12-1-8	0100 1001
74	4A	AH		¢	J		12-2-8	0100 1010
75	4B	SH	.	.	K	B A 8 2 1	12-3-8	0100 1011
76	4C	MH	¤)	<	L	B A 8 4	12-4-8	0100 1100
77	4D		[	(	M	B A 8 4 1	12-5-8	0100 1101
78	4E	CVD	<	+	N	B A 8 4 2	12-6-8	0100 1110
79	4F	CVB	≠	|	O	B A 8 4 2 1	12-7-8	0100 1111
80	50	ST	& +	&	P	B A	12	0101 0000
81	51				Q		12-11-1-9	0101 0001
82	52				R		12-11-2-9	0101 0010
83	53				S		12-11-3-9	0101 0011
84	54	N			T		12-11-4-9	0101 0100
85	55	CL			U		12-11-5-9	0101 0101
86	56	O			V		12-11-6-9	0101 0110
87	57	X			W		12-11-7-9	0101 0111
88	58	L			X		12-11-8-9	0101 1000
89	59	C			Y		11-1-8	0101 1001
90	5A	A		!	Z		11-2-8	0101 1010
91	5B	S	$	$	[	B 8 2 1	11-3-8	0101 1011
92	5C	M	*	*	\	B 8 4	11-4-8	0101 1100
93	5D	D	]	)	]	B 8 4 1	11-5-8	0101 1101
94	5E	AL	;	;	¬ ^	B 8 4 2	11-6-8	0101 1110
95	5F	SL	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

Code Translation Table (con't)

Dec.	Hex	Instruction (RX)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
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
128	80	SSM					12-0-1-8	1000 0000
129	81			a			12-0-1	1000 0001
130	82	LPSW		b			12-0-2	1000 0010
131	83	Diagnose		c			12-0-3	1000 0011
132	84	WRD		d			12-0-4	1000 0100
133	85	RDD		e			12-0-5	1000 0101
134	86	BXH		f			12-0-6	1000 0110
135	87	BXLE		g			12-0-7	1000 0111
136	88	SRL		h			12-0-8	1000 1000
137	89	SLL		i			12-0-9	1000 1001
138	8A	SRA					12-0-2-8	1000 1010
139	8B	SLA					12-0-3-8	1000 1011
140	8C	SRDL					12-0-4-8	1000 1100
141	8D	SLDL					12-0-5-8	1000 1101
142	8E	SRDA					12-0-6-8	1000 1110
143	8F	SLDA					12-0-7-8	1000 1111
144	90	STM					12-11-1-8	1001 0000
145	91	TM		j			12-11-1	1001 0001
146	92	MVI		k			12-11-2	1001 0010
147	93	TS		l			12-11-3	1001 0011
148	94	NI		m			12-11-4	1001 0100
149	95	CLI		n			12-11-5	1001 0101
150	96	OI		o			12-11-6	1001 0110
151	97	XI		p			12-11-7	1001 0111
152	98	LM		q			12-11-8	1001 1000
153	99			r			12-11-9	1001 1001
154	9A						12-11-2-8	1001 1010
155	9B						12-11-3-8	1001 1011
156	9C	SIO, SIOF					12-11-4-8	1001 1100
157	9D	TIO					12-11-5-8	1001 1101
158	9E	HIO, HDV					12-11-6-8	1001 1110
159	9F	TCH					12-11-7-8	1001 1111
160	A0						11-0-1-8	1010 0000
161	A1			~			11-0-1	1010 0001
162	A2			s			11-0-2	1010 0010
163	A3			t			11-0-3	1010 0011
164	A4			u			11-0-4	1010 0100
165	A5			v			11-0-5	1010 0101
166	A6			w			11-0-6	1010 0110
167	A7			x			11-0-7	1010 0111
168	A8			y			11-0-8	1010 1000
169	A9			z			11-0-9	1010 1001
170	AA						11-0-2-8	1010 1010
171	AB						11-0-3-8	1010 1011
172	AC						11-0-4-8	1010 1100
173	AD						11-0-5-8	1010 1101
174	AE						11-0-6-8	1010 1110
175	AF	MC					11-0-7-8	1010 1111

Code Translation Table (con't)

Dec.	Hex	Instruction (RS & SI)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
176	B0						12-11-0-1-8	1011 0000
177	B1						12-11-0-1	1011 0001
178	B2	See below					12-11-0-2	1011 0010
179	B3						12-11-0-3	1011 0011
180	B4						12-11-0-4	1011 0100
181	B5						12-11-0-5	1011 0101
182	B6	STCTL					12-11-0-6	1011 0110
183	B7	LCTL					12-11-0-7	1011 0111
184	B8						12-11-0-8	1011 1000
185	B9						12-11-0-9	1011 1001
186	BA						12-11-0-2-8	1011 1010
187	BB						12-11-0-3-8	1011 1011
188	BC						12-11-0-4-8	1011 1100
189	BD	CLM					12-11-0-5-8	1011 1101
190	BE	STCM					12-11-0-6-8	1011 1110
191	BF	ICM					12-11-0-7-8	1011 1111
192	C0		?	{		B A 8 2	12-0	1100 0000
193	C1		A	A		B A 1	12-1	1100 0001
194	C2		B	B		B A 2	12-2	1100 0010
195	C3		C	C		B A 2 1	12-3	1100 0011
196	C4		D	D		B A 4	12-4	1100 0100
197	C5		E	E		B A 4 1	12-5	1100 0101
198	C6		F	F		B A 4 2	12-6	1100 0110
199	C7		G	G		B A 4 2 1	12-7	1100 0111
200	C8		H	H		B A 8	12-8	1100 1000
201	C9		I	I		B A 8 1	12-9	1100 1001
202	CA						12-0-2-8-9	1100 1010
203	CB						12-0-3-8-9	1100 1011
204	CC			⌠			12-0-4-8-9	1100 1100
205	CD						12-0-5-8-9	1100 1101
206	CE			⊦			12-0-6-8-9	1100 1110
207	CF						12-0-7-8-9	1100 1111
208	D0		!	}		B 8 2	11-0	1101 0000
209	D1	MVN	J	J		B 1	11-1	1101 0001
210	D2	MVC	K	K		B 2	11-2	1101 0010
211	D3	MVZ	L	L		B 2 1	11-3	1101 0011
212	D4	NC	M	M		B 4	11-4	1101 0100
213	D5	CLC	N	N		B 4 1	11-5	1101 0101
214	D6	OC	O	O		B 4 2	11-6	1101 0110
215	D7	XC	P	P		B 4 2 1	11-7	1101 0111
216	D8		Q	Q		B 8	11-8	1101 1000
217	D9		R	R		B 8 1	11-9	1101 1001
218	DA						12-11-2-8-9	1101 1010
219	DB						12-11-3-8-9	1101 1011
220	DC	TR					12-11-4-8-9	1101 1100
221	DD	TRT					12-11-5-8-9	1101 1101
222	DE	ED					12-11-6-8-9	1101 1110
223	DF	EDMK					12-11-7-8-9	1101 1111
224	E0		≠	\		A 8 2	0-2-8	1110 0000
225	E1						11-0-1-9	1110 0001
226	E2		S	S		A 2	0-2	1110 0010
227	E3		T	T		A 2 1	0-3	1110 0011
228	E4		U	U		A 4	0-4	1110 0100
229	E5		V	V		A 4 1	0-5	1110 0101
230	E6		W	W		A 4 2	0-6	1110 0110
231	E7		X	X		A 4 2 1	0-7	1110 0111

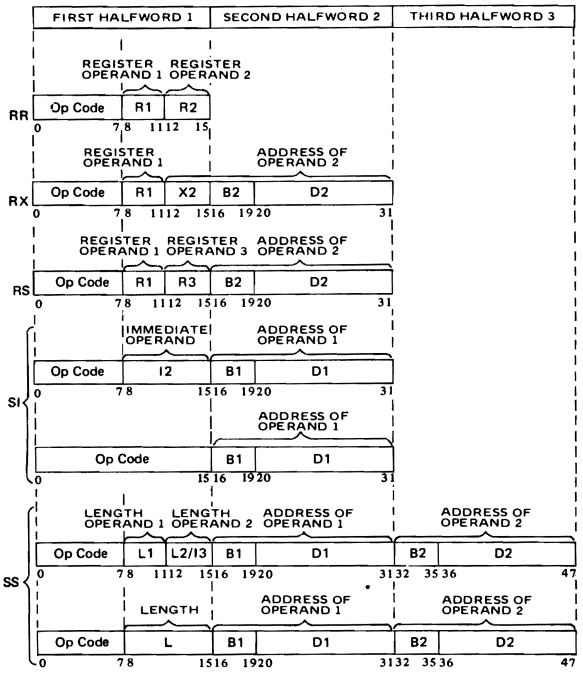
Op code

B202 – STIDP
B203 – STIDC
B204 – SCK
B205 – STCK

Code Translation Table (con't)

Dec.	Hex	Instruction (SS)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	Card Code	Binary
232	E8		Y	Y		A 8	0-8	1110 1000
233	E9		Z	Z		A 8 1	0-9	1110 1001
234	EA						11-0-2-8-9	1110 1010
235	EB						11-0-3-8-9	1110 1011
236	EC			┌			11-0-4-8-9	1110 1100
237	ED						11-0-5-8-9	1110 1101
238	EE						11-0-6-8-9	1110 1110
239	EF						11-0-7-8-9	1110 1111
240	F0	SRP	0	0		8 2	0	1111 0000
241	F1	MVO	1	1		1	1	1111 0001
242	F2	PACK	2	2		2	2	1111 0010
243	F3	UNPK	3	3		2 1	3	1111 0011
244	F4		4	4		4	4	1111 0100
245	F5		5	5		4 1	5	1111 0101
246	F6		6	6		4 2	6	1111 0110
247	F7		7	7		4 2 1	7	1111 0111
248	F8	ZAP	8	8		8	8	1111 1000
249	F9	CP	9	9		8 1	9	1111 1001
250	FA	AP		¦			12-11-0-2-8-9	1111 1010
251	FB	SP					12-11-0-3-8-9	1111 1011
252	FC	MP					12-11-0-4-8-9	1111 1100
253	FD	DP					12-11-0-5-8-9	1111 1101
254	FE						12-11-0-6-8-9	1111 1110
255	FF						12-11-0-7-8-9	1111 1111

MACHINE INSTRUCTION FORMATS



CONTROL REGISTERS

CR	Bits	Name of field	Function or feature	Reset
0	0	Block-multiplex mode	Block-multiplex'g control	0
	24	Timer mask	Ext'd external masking	1
	25	Interrupt key mask		1
	26	External signal mask		1
2	0-31	Channel masks	Extended I/O masking	1
8	16-31	Monitor masks	Monitoring	0
14	0	Hard stop mode	Machine-check handling	1
	1	Synchronous MCEL mask		1
	2	I/O extended logout mask		0
	4	Recovery report mask		0
	5	Configuration report mask		0
	6	External damage report mask		1
	7	Warning mask		0
	8	Asynchronous MCEL mask		0
	9	Asynchronous fixed log mask		0
15	8-31	MCEL pointer	Machine-check handling	512

Condition Codes

CONDITION CODES

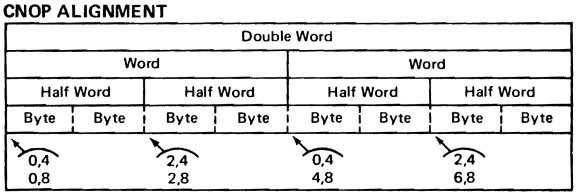
Condition Code Setting	0	1	2	3
Mask Bit Position	8	4	2	1
Floating-Point Arithmetic				
Add Normalized S/L/E	zero	<zero	>zero	–
Add Unnormalized S/L	zero	<zero	>zero	–
Compare S/L (A:B)	equal	A low	A high	–
Load and Test S/L	zero	<zero	>zero	–
Load Complement S/L	zero	<zero	>zero	–
Load Negative S/L	zero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

Codes for Program Interruption - CNOP Alignment - Edit and EDMK Pattern Characters

CODES FOR PROGRAM INTERRUPTION

Interruption Code		Program Interruption Cause	Interruption Code		Program Interruption Cause
Dec	Hex		Dec	Hex	
1	0001	Operation	10	000A	Decimal overflow
2	0002	Privileged operation	11	000B	Decimal divide
3	0003	Execute	12	000C	Exponent overflow
4	0004	Protection	13	000D	Exponent underflow
5	0005	Addressing	14	000E	Significance
6	0006	Specification	15	000F	Floating-point divide
7	0007	Data	64	0040	Monitoring
8	0008	Fixed-point overflow			
9	0009	Fixed-point divide			

CNOP ALIGNMENT



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

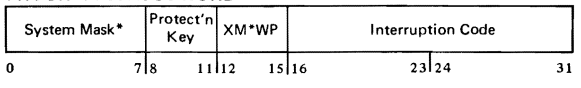
FIXED STORAGE LOCATIONS*

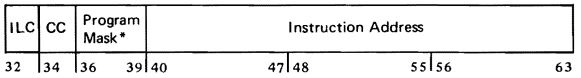
Area, dec.	Hex addr	Purpose
0-7	0	Initial program loading PSW, restart new PSW
8-15	8	Initial program loading CCW1, restart old PSW
16-23	10	Initial program loading CCW2
24-31	18	External old PSW
32-39	20	Supervisor Call old PSW
40-47	28	Program old PSW
48-55	30	Machine-check old PSW
56-63	38	Input/output old PSW
64-71	40	Channel status word
72-75	48	Channel address word
76-79	4C	Unused
80-83	50	Timer (uses bytes 50, 51, and 52)
84-87	54	Unused
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
128-147	80	Reserved
148-149	94	Monitor class [0-7 zeros, 8-15 class number]
150-155	96	Reserved
156-159	9C	Monitor code [0-7 zeros, 8-31 monitor code]
160-167	A0	Reserved
168-171	A8	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)
180-231	B4	Reserved
232-239	E8	Machine-check interruption code (see diagram)
240-247	F0	Unused
248-251	F8	Failing processor storage address [0-7 zeros, 8-31 addr]
252-255	FC	Model-dependent
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 extended logout area (size varies)

*Information from GA22-7000-1. Functions and use of fields beyond 127 may vary between models. See functional characteristics manual for specific model.

PSW - CAW - CCW - CSW

PROGRAM STATUS WORD

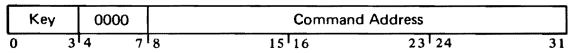




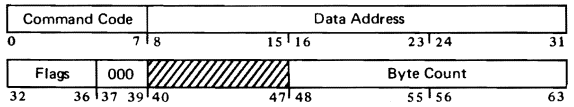
0 Channel 0 mask
1 Channel 1 mask
2 Channel 2 mask
3 Channel 3 mask
4 Channel 4 mask
5 Channel 5 mask
6 Mask for channel 6 and up
7 External mask
12 Not used — must be zero
13 (M) Machine check mask
14 (W=1) Wait state
15 (P=1) Problem state
32-33 (ILC) Instruction length code
34-35 (CC) Condition code
36 Fixed-point overflow mask
37 Decimal overflow mask
38 Exponent underflow mask
39 Significance mask

*A one-bit permits an interrupt.

CHANNEL ADDRESS WORD (hex 48)

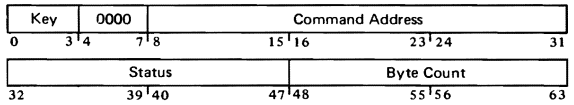


CHANNEL COMMAND WORD



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.

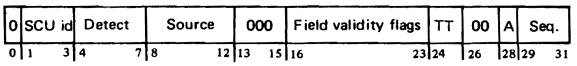
CHANNEL STATUS WORD (hex 40)



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-controlled 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

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

LIMITED CHANNEL LOGOUT (hex B0)



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

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)	Acc 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,(L2,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)	Acc Data Dec Oflo	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	Acc Fxpt Oflo	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)	Acc	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)		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.	Spec Exp Oflo Exp Uflo Sign	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
Add Normalized (Long)	AD	6A	RX	R1, D2(X2, B2)	FP Add opr 2 to opr 1 (Sto) (FPR) S | Char | Fraction 0 1 7 8 63	Acc Spec Sign Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
Add Normalized (Long)	ADR	2A	RR	R1, R2	FP Add opr 2 to opr 1 (FPR) (FPR)	Spec Sign Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
Add Normalized (Short)	AE	7A	RX	R1, D2(X2, B2)	FP Add opr 2 to opr 1 (Sto) (FPR) S | Char | Fraction 0 1 7 8 31 (Low-order halves of FPR ignored and unchanged)	Acc Spec Sign Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
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)	Spec Sign Exp Oflo Exp Uflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --

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)	Acc Spec Sign Exp Oflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
Add Unnormalized (Long)	AWR	2E	RR	R1,R2	FP Add opr 2 to opr 1 (FPR) (FPR)	Spec Sign Exp Oflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
Add Unnormalized 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)	Acc Spec Sign Exp Oflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
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)	Spec Sign Exp Oflo	0 Fract = 0 1 Result < 0 2 Result > 0 3 --
AND	N	54	RX	R1, D2(X2,B2)	Place the product of both opr s into opr 1	Acc	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)	Acc	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	Acc	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 High	BH	47(BC 2)	RX,Ext.	D2(X2,B2)	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 + if opr 3 addr is even. Branch to opr 2 addr if sum > 3/opr 3 + 1	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 if Mixed	BM	47(BC 4)	RX, Ext.	D2 (X2, B2)	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 Not Equal	BNE	47(BC 7)	RX, Ext.	D2 (X2, B2)	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 Low	BNL	47(BC 11)	RX, Ext.	D2 (X2, B2)	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 Ones	BNO	47(BC 14)	RX, Ext.	D2 (X2, B2)	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 Zeros	BNZ	47(BC 7)	RX, Ext.	D2 (X2, B2)	Branch if mask = cond code	None	Unchanged
Branch if Ones	BO	47(BC 1)	RX, Ext.	D2 (X2, B2)	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 Plus	BP	47(BC 2)	RX, Ext.	D2 (X2, B2)	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 on Zero	BZ	47(BC 8)	RX, Ext.	D2 (X2, B2)	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	47(BC 15)	RR, Ext.	R2	Branch if mask = cond code	None	Unchanged
Compare	C	59	RX	R1, D2(X2, B2)	Compare opr 1 algebraically to opr 2 (Reg)	Acc	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 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)	Acc Data	0 opr's = 1 1st < 2 1st >
Compare Halfword	CH	49	RX	R1, D2(X2,B2)	Compare opr 1 algebraically to opr 2 (Hi-order 16 bits expanded) opr 2	Acc	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)	Acc	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)	Acc	0 opr's = 1 1st < 2 1st >
Compare Logical	CLI	95	SI	D1(B1), I2	Compare opr 1 to opr 2 (Imm) (Sto) (binary left to right) (Terminates if/when ≠ found)	Acc	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)	None	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)	Acc	0 opr's = 1 1st < 2 1st > 3 --

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)	Acc Spec	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)	Acc Spec	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)	Spec	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)	Acc Spec	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)	Spec	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	Acc Spec 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	Acc	Unchanged
Diagnose	----	83		See IBM System/370 Principles of Operation, GA22-7000	See IBM System/370 Principles of Operation, GA22-7000	Priv Oper Acc Spec	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 quotient and remainder	Acc Spec Fxpt Div	Unchanged
Divide	DR	1D	RR	R1, R2	Divide opr 1 by opr 2 Dividend: even and odd pair regs Opr 1 becomes quotient and remainder (full word only)	Spec 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.)	Acc Spec Data Dec Div	Unchanged
Divide (Long)	DD	6D	RX	R1, D2(X2,B2)	FP Divide opr 1 by opr 2 (FPR) (Sto) Opr 1 becomes quotient (prenormalized)	Acc Spec Exp Oflo FP Div Exp Uflo	Unchanged
Divide (Long)	DDR	2D	RR	R1, R2	FP Divideopr 1 by opr 2 Prenormalize (FPR) (FPR) (Dividend) (Divisor) Oper 1 becomes quotient	Spec 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)	Acc Spec Exp Oflo Exp Uflo FP Div	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)	Spec Exp Oflo FP Div	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	Acc Data	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)	Acc Data	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	Acc	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.	Acc	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.	Acc	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.	Acc Exec	Set by executed instruction
Halve, Long	HDR	24	RR	R1, R2	Opr 2 is divided by 2 and placed in opr 1.	Spec	Unchanged
Halve, Short	HER	34	RR	R1, R2	Opr 2 is divided by 2 and placed in opr 1.	Spec	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 3 Not oper
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 oper
Insert Character	IC	43	RX	R1, D2(X2,B2)	Byte at opr 2 is inserted in high order byte of reg at opr 1.	Acc	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.	Acc	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 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 Adr Spec	Unchanged
Load	L	58	RX	R1, D2(X2,B2)	Load opr 2 into opr 1	Acc	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.)	Spec	0 Result Fraction = 0 1 Result < 0 2 Result > 0
Load and Test (Short)	LTER	32	RR	R1, R2	Opr 2 into 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.)	Spec	0 Result Fraction = 0 1 Result < 0 2 Result > 0
Load Complement	LCR	13	RR	R1, R2	2's complement of opr 2 into oper 1 (overflow when max negative number is complemented)	Fxpt Oflo	0 Result = 0 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 1 (FPR) (FPR) (Opr 1 sign inverted, low-order half unchanged) (Opr 2 unchanged)	Spec	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)	Same as LCER	Same as LCER
Load Control	LCTL	B7	RS	R1, R3, D2(B2)	Cntl regs from opr 1 to opr 3 loaded with info starting at opr 2.	Acc Spec Priv	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)	Acc	Unchanged
Load (Long)	LD	68	RX	R1, D2(X2,B2)	Opr 2 into opr 1 (Sto) (FPR)	Acc Spec	Unchanged
Load (Long)	LDR	28	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR)	Spec	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 opr 1, ending reg specified by opr 3 (Reg wrap-around possible)	Acc	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	Spec	Result 0 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)	Spec	Result 0 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	Spec	Result 0 Fract = 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)	Spec	Result 0 Fract = 0 2 Result > 0

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Load PSW	LPSW	82	S	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 Acc Spec	Set by new PSW 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)	Acc Spec	Unchanged
Load (Short)	LER	38	RR	R1, R2	Opr 2 into opr 1 (FPR) (FPR) (Low-order half of opr 1 unchanged)	Spec	Unchanged
Load Real Address	LRA	B1	RX	R1, D2(X2,B2)	Real adr corresponding to opr 2 logical adr placed in opr 1.	Priv Adr Trans	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.	Spec Exp Oflo	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.	Spec Exp Oflo	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 Spec	Unchanged
Move	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)	Acc	Unchanged
Move	MVI	92	SI	D1(B1), I2	Move the 1 byte from the instruction stream (8-15) to opr 1.	Acc	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.	Acc Spec	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.)	Acc	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.)	Acc	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)	Acc	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)	Acc Spec	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)	Spec	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)	Spec Exp Oflo Exp Uflo	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.	Acc Spec Data	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)	Acc	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.)	Acc Spec Exp Oflo Exp Uflo	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.)	Spec Exp Oflo Exp Uflo	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 PFR 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)	Acc Spec Exp Oflo Exp Uflo	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)	Spec Exp Oflo Exp Uflo	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.)	Acc Spec Exp Oflo Exp Uflo	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.)	Spec Exp Oflo Exp Uflo	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	O	56	RX	R1, D2(X2,B2)	The ORed sum of both opr's into opr 1	Acc	0 Result = 0 1 Result ≠ 0
OR	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)	Acc	0 Result = 0 1 Result ≠ 0
OR	OR	16	RR	R1, R2	The ORed sum of both opr's into opr 1	None	0 Result = 0 1 Result ≠ 0
OR	OI	96	SI	D1(B1), I2	OR the 1 byte from the instruction stream (8-15) to opr 1	Acc	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)	Acc	Unchanged
Purge TLB	PTLB	B20D	S	---	Invalidate current info in TLB.	Priv	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 Acc	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 Adr	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.	Acc Spec Priv	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	Acc Priv Spec	Unchanged
Set CPU Timer	SPT	B208	S	D1(B1)	Dblwd at opr 1 replaces curr value of CPU timer.	Acc Priv Spec	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 (18-23) (Bits 0, 1 and 8-31 of opr 1 are ignored and unchanged.)	None	Set by bits 2 and 3
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)	Priv Adr Spec	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 Acc	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.	Acc Data Dec Oflo	0 Result = 0 1 Result < 0 2 Result > 0 3 Result Oflo
Shift Left Double	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).	Spec 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)	Spec	Unchanged
Shift Left Single	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	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).	Spec	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). (Vacated bits are replaced with zeros) (Hi-order bit participates in the shift)	Spec	Unchanged
Shift Right Single	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
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	Acc	Unchanged
Store Channel ID	STIDC	B203	S	D1(B1)	Store opr 1 at loc 168 in main storage.	Priv	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.	Acc	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.	Acc	Unchanged
Store Clock	STCK	B205	S	D1(B1)	Current val of TOD clock stored in 8 bytes at opr 1.	Acc	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.	Acc Priv Spec	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 Acc Spec	Unchanged
Store CPU ID	STIDP	B202	S	D1(B1)	CPU info stored in 8 bytes at opr 1.	Priv Acc Spec	Unchanged
Store CPU Timer	STPT	B209	S	D1(B1)	Curr contents of CPU timer stored in dblwd at opr 1.	Priv Acc Spec	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)	Acc	Unchanged
Store (Long)	STD	60	RX	R1, D2(X2,B2)	FP opr 1 to opr 2's location.	Adr Prot Spec Oper	Unchanged
Store Multiple	STM	90	RS	R1, R3, D2(B2)	Opr 1 thru opr 3 are stored at opr 2's location in ascending order. Starting reg specified by opr 1, ending reg specified by opr 3. (Reg wrap-around possible)	Acc	Unchanged

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
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)	Acc Spec	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.	Acc Priv	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.	Acc Priv	Unchanged
Subtract	S	5B	RX	R1, D2(X2,B2)	Subtract opr 2 from opr 1 and place the difference into opr 1.	Acc Fxpt Oflo	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)	Acc Data Dec Oflo	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.	Acc Fxpt Oflo	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.	Acc	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)	Spec Exp Oflo Exp Uflo Sign	0 Fract = 0 1 Fract < 0 2 Fract > 0 3 --

Operation	Mnemonic	Op Code	Format	Operands	Description	Exceptions	Cond Code
Subtract Normalized (Extended) (Con't)	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.)	Acc Spec Sign Exp Oflo Exp Uflo	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 --
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.)	Spec Sign 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).	Acc Spec Sign 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)	Spec Sign 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.)	Acc Spec Sign Exp Oflo	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.)	Spec Sign Exp Oflo	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.)	Acc Spec Sign Exp Oflo	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.)	Spec Sign Exp Oflo	Result 0 Fract = 0 1 Result < 0 2 Result > 0 3 --

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	Acc	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.	Acc	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(L1,B1), D2(B2)	Opr 1 (argument byte) added to the initial adr of opr 2 (24-31). This adr now is the loc of the function byte which replaces the original argument byte (left to right byte by byte) (All data is valid) (Oper is terminated when opr 1 field is exhausted)	Acc	Unchanged
Translate and Test	TRT	DD	SS	D1(L,B1), D2(B2)	(Same as TR) When the function byte is a zero the next argument byte is translated. Both opr's remain unchanged. When the function byte is a non-zero the operation is completed. The generated argument adr is placed into GPR 1, 8-31. Bits 0-7 remain unchanged. The function byte is placed into GPR 2, 24-31. (Left to right byte by byte). Bits 0-23 remain unchanged. If opr 1 is exhausted before a non-zero cond, the opr is completed and GPRs 1 and 2 remain unchanged.	Acc	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.)	Acc	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 Acc	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.)	Acc Data Dec Oflo	0 Result = 0 1 Result < 0 2 Result > 0 3 Overflow

Operation	Name Entry	Operand Entry
ACTR	Blank	A SETA expression
AGO	A sequence symbol or blank	A sequence symbol
AIF	A sequence symbol or blank	A logical expression enclosed in parentheses, immediately followed by a sequence symbol
ANOP	A sequence symbol or blank	Must not be present
CCW	Any symbol or blank	Four operands, separated by commas
CNOP	Any symbol or blank	Two absolute expressions, separated by a comma
COM	Any symbol or blank	Must not be present
COPY	Must not be present	One ordinary symbol
CSECT	Any symbol or blank	Must not be present
CXD	Any symbol or blank	Must not be present
DC	Any symbol or blank	One or more operands, separated by commas
DROP	A sequence symbol or blank	One to sixteen absolute expressions, separated by commas; or blank
DS	Any symbol or blank	One or more operands, separated by commas
DSECT	Any symbol or blank	Must not be present
DXD	Any symbol	One or more operands, separated by commas
EJECT	A sequence symbol or blank	Must not be present
END	A sequence symbol or blank	A relocatable expression or blank
ENTRY	A sequence symbol or blank	One or more relocatable symbols, separated by commas
EQU	An ordinary symbol or a variable symbol	One to three operands, separated by commas
EXTRN	A sequence symbol or blank	One or more relocatable symbols, separated by commas
GBLA	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas [2]
GBLB	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas [2]

OS/VS System Assembler Instructions (con't)

Operation	Name Entry	Operand Entry
GBLC	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas[2]
ICTL	Must not be present	One to three decimal values, separated by commas
ISEQ	Must not be present	Not present or a blank
LCLA	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas[2]
LCLB	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas[2]
LCLC	Must not be present	One or more variable symbols that are to be used as SET symbols, separated by commas[2]
LTORG	Any symbol or blank	Must not be present
MACRO[1]	Must not be present	Must not be present
MEND[1]	A sequence symbol or blank	Must not be present
MEXIT[1]	A sequence symbol or blank	Must not be present
MNOTE	A sequence symbol or blank	A severity code followed by a comma (this much is optional) followed by any combination of characters enclosed in apostrophes
OPSYN	An ordinary symbol	A machine instruction mnemonic code, an extended mnemonic code, a macro operation, an assembler operation, an operation code defined by a previous OPSYN instruction, or blank
ORG	Any symbol or blank	A relocatable expression or blank
POP	A sequence symbol or blank	One or more operands, separated by a comma
PRINT	A sequence symbol or blank	One to three operands
PUNCH	A sequence symbol or blank	One to eighty characters, enclosed in apostrophes
PUSH	A sequence symbol or blank	One or more operands, separated by a comma
REPRO	A sequence symbol or blank	Must not be present
SETA	A SETA symbol	An arithmetic expression
SETB	A SETB symbol	A 0 or a 1, a SETB symbol, variable symbols except &SYSLIST, or a logical expression enclosed in parentheses

OS/VS System Assembler Instructions (con't)

Operation	Name Entry	Operand Entry
SETC	A SETC symbol	A duplication factor (a SETA expression enclosed in parentheses) if desired, followed by a type attribute, a character expression, a substring notation, or a concatenation of character expressions and substring notations
SPACE	A sequence symbol or blank	A decimal self-defining term or blank
START	Any symbol or blank	A self-defining term or blank
TITLE	A variable symbol, alphameric character string, or a combination of variable symbol and character string, or a sequence symbol, or a blank.	One to 100 characters, enclosed in apostrophes
USING	A sequence symbol or blank	An absolute or relocatable expression followed by 1 to 16 absolute expressions, separated by commas
WXTRN	A sequence symbol or blank	One or more relocatable symbols, separated by commas

[1]Can be used only as part of a macro definition.
[2]SET symbols can be defined as subscripted SET symbols.

OS/VS 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.

OS/VS 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.

PLEASE COMMENT

Use Reader's Comment Form

Statement	Variable Symbols: Symbolic Parameter	Global SET Symbols: SETA	Global SET Symbols: SETB	Global SET Symbols: SETC	Local SET Symbols: SETA	Local SET Symbols: SETB	Local SET Symbols: SETC	System Variable Symbols: &SYSNDX	&SYSECT	&SYSLIST	&SYSPARM	&SYSDATE	&SYSTIME	Attributes: 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[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
Iner 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).

OS/VS System Assembler Conditional Assembly Expressions

Expression	Arithmetic Expressions	Character Expressions	Logical Expressions
Can contain	• Self-defining terms • Length, scaling, integer, count, and number attributes • SETA and SETB symbols • SETC symbols whose values are a decimal self-defining term • &SYSPARM if its value is a decimal self-defining term • Symbolic parameters if the corresponding operand is a decimal self-defining term • &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	• Any combination of characters enclosed in apostrophes • Any variable symbol enclosed in apostrophes • A concatenation of variable symbols and other characters enclosed in apostrophes • A type attribute reference	• A 0 or a 1 • SETB symbols • Arithmetic relations[1] • Character relations[2] • Variable symbols except &SYSLIST
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 • 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.

OS/VS 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

OS/VS 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 instrucfion	• 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

OS/VS System Assembler Variable Symbols (con't)

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.

Hexadecimal and Decimal Conversion

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

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

BINARY CONVERSION

Dec	Hex	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001 0000

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

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

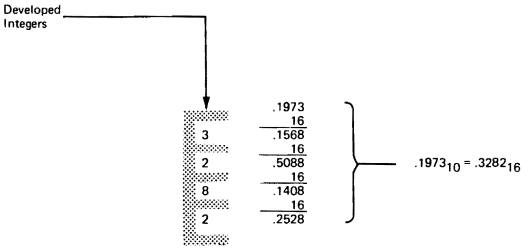
Example: 2 x 4 = 08, F x 2 = 1E

	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 to Hexadecimal Conversion Information

Decimal to Hexadecimal Conversion: Locate the decimal fraction (.1973) in the table. If the exact figure is not shown, locate the next higher and lower fractions (.19726563, .19750977). The first digits of the hexadecimal fraction are at the top of the column (.32). To locate the third digit, determine by observation or subtraction the smaller difference between the known fraction and each of the found fractions. The smaller difference identifies the correct line (.008). The hexadecimal equivalent is .328.

If more places to the right of the decimal point are required in the hexadecimal fraction, multiply the decimal fraction by 16 and develop integers as successive terms of the hexadecimal fraction. Using the previous sample decimal fraction:



Hexadecimal to Decimal Conversion: Locate the first two digits (.1E) of the hexadecimal fraction (.1E9) in the horizontal row of column headings. Locate the third digit (.009) in the left most column of the table. Follow the .009 line horizontally to the right to the .1E column. The decimal equivalent is .11938477. The decimal fractions in the table were carried to eight places and rounded. If 2 places are required, or if the hexadecimal fraction exceeds the capacity of the table, express the hexadecimal fraction as powers of 16 (expansion). For example:

$$.1E94_{16} = 1(16^{-1}) + 14(16^{-2}) + 9(16^{-3}) + 4(16^{-4})$$

$$= 1(.0625) + 14(.00390625) + 9(.000244440625) + 4(.0000152587890625)$$

$$= .1194458007812500_{10}$$

	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09	.0A	.0B	.0C	.0D	.0E	.0F
.000	.00000000	.00390625	.00781250	.01171875	.01562500	.01953125	.02343750	.02734375	.03125000	.03515625	.03906250	.04296875	.04687500	.05078125	.05468750	.05859375
.001	.00024414	.00415039	.00805664	.01196289	.01586914	.01977539	.02368164	.02758789	.03149414	.03540039	.03930664	.04321289	.04711914	.05102539	.05493164	.05883789
.002	.00048828	.00439453	.00830078	.01220703	.01611328	.02001953	.02392578	.02783203	.03173828	.03564453	.03955078	.04345703	.04736328	.05126953	.05517578	.05908203
.003	.00073242	.00463867	.00854492	.01245117	.01635742	.02026367	.02416992	.02807617	.03198242	.03588867	.03979492	.04370117	.04760742	.05151367	.05541992	.05932617
.004	.00097656	.00488281	.00878906	.01269531	.01660156	.02050781	.02441406	.02832031	.03222656	.03613281	.04003906	.04394531	.04785156	.05175781	.05566406	.05957031
.005	.00122070	.00512695	.00903320	.01293945	.01684570	.02075195	.02465820	.02856445	.03247070	.03637695	.04028320	.04418945	.04809570	.05200195	.05590820	.05981445
.006	.00146484	.00537109	.00927734	.01318359	.01708984	.02099609	.02490234	.02880859	.03271484	.03662109	.04052734	.04443359	.04833984	.05224609	.05615234	.06005859
.007	.00170898	.00561523	.00952148	.01342773	.01733398	.02124023	.02514648	.02905273	.03295898	.03686523	.04077148	.04467773	.04858398	.05249023	.05639648	.06030273
.008	.00195313	.00585938	.00976563	.01367188	.01757813	.02148438	.02539063	.02929688	.03320313	.03710938	.04101563	.04492188	.04882813	.05273438	.05664063	.06054688
.009	.00219727	.00610352	.01000977	.01391602	.01782227	.02172852	.02563477	.02954102	.03344727	.03735352	.04125977	.04516602	.04907227	.05297852	.05688477	.06079102
.00A	.00244141	.00634766	.01025391	.01416016	.01806641	.02197266	.02587891	.02978516	.03369141	.03759766	.04150391	.04541016	.04931641	.05322266	.05712891	.06103516
.00B	.00268555	.00659180	.01049805	.01440430	.01831055	.02221680	.02612305	.03002930	.03393555	.03784180	.04174805	.04565430	.04956055	.05346680	.05737305	.06127930
.00C	.00292969	.00683594	.01074219	.01464844	.01855469	.02246094	.02636719	.03027344	.03417969	.03808594	.04199219	.04589844	.04980469	.05371094	.05761719	.06152344
.00D	.00317383	.00708008	.01098633	.01489258	.01879883	.02270508	.02661133	.03051758	.03442383	.03833008	.04223633	.04614258	.05004883	.05395508	.05786133	.06176758
.00E	.00341797	.00732422	.01123047	.01513672	.01904297	.02294922	.02685547	.03076172	.03466797	.03857422	.04248047	.04638672	.05029297	.05419922	.05810547	.06201172
.00F	.00366211	.00756836	.01147461	.01538086	.01928711	.02319336	.02709961	.03100586	.03491211	.03881836	.04272461	.04663086	.05053711	.05444336	.05834961	.06225586
	.10	.11	.12	.13	.14	.15	.16	.17	.18	.19	.1A	.1B	.1C	.1D	.1E	.1F
.000	.06250000	.06640625	.07031250	.07421875	.07812500	.08203125	.08593750	.08984375	.09375000	.09765625	.10156250	.10546875	.10937500	.11328125	.11718750	.12109375
.001	.06274414	.06665039	.07055664	.07446289	.07836914	.08227539	.08618164	.09008789	.09399414	.09790039	.10180664	.10571289	.10961914	.11352539	.11743164	.12133789
.002	.06298828	.06689453	.07080078	.07470703	.07861328	.08251953	.08642578	.09033203	.09423828	.09814453	.10205078	.10595703	.10986328	.11376953	.11767578	.12158203
.003	.06323242	.06713867	.07104492	.07495117	.07885742	.08276367	.08666992	.09057617	.09448242	.09838867	.10229492	.10620117	.11010742	.11401367	.11791992	.12182617
.004	.06347656	.06738281	.07128906	.07519531	.07910156	.08300781	.08691406	.09082031	.09472656	.09863281	.10253906	.10644531	.11035156	.11425781	.11816406	.12207031
.005	.06372070	.06762695	.07153320	.07543945	.07934570	.08325195	.08715820	.09106445	.09497070	.09887695	.10278320	.10668945	.11059570	.11450195	.11840820	.12231445
.006	.06396484	.06787109	.07177734	.07568359	.07958984	.08349609	.08740234	.09130859	.09521484	.09912109	.10302734	.10693359	.11083984	.11474609	.11865234	.12255859
.007	.06420898	.06811523	.07202148	.07592773	.07983398	.08374023	.08764648	.09155273	.09545898	.09936523	.10327148	.10717773	.11108398	.11499023	.11889648	.12280273
.008	.06445313	.06835938	.07226563	.07617188	.08007813	.08398438	.08789063	.09179688	.09570313	.09960938	.10351563	.10742188	.11132813	.11523438	.11914063	.12304688
.009	.06469727	.06860352	.07250977	.07641602	.08032227	.08422852	.08813477	.09204102	.09594727	.09985352	.10375977	.10766602	.11157227	.11547852	.11938477	.12329102
.00A	.06494141	.06884766	.07275391	.07666016	.08056641	.08447266	.08837891	.09228516	.09619141	.10009766	.10400391	.10791016	.11181641	.11572266	.11962891	.12353516
.00B	.06518555	.06909180	.07299805	.07690430	.08081055	.08471680	.08862305	.09252930	.09643555	.10034180	.10424805	.10815430	.11206055	.11596680	.11987305	.12377930
.00C	.06542969	.06933594	.07324219	.07714844	.08105469	.08496094	.08886719	.09277344	.09667969	.10058594	.10449219	.10839844	.11230469	.11621094	.12011719	.12402344
.00D	.06567383	.06958008	.07348633	.07739258	.08129883	.08520508	.08911133	.09301758	.09692383	.10083008	.10473633	.10864258	.11254883	.11645508	.12036133	.12426758
.00E	.06591797	.06982422	.07373047	.07763672	.08154297	.08544922	.08935547	.09326172	.09716797	.10107422	.10498047	.10888672	.11279297	.11669922	.12060547	.12451172
.00F	.06616211	.07006836	.07397461	.07788086	.08178711	.08569336	.08959961	.09350586	.09741211	.10131836	.10522461	.10913086	.11303711	.11694336	.12084961	.12475586
	.20	.21	.22	.23	.24	.25	.26	.27	.28	.29	.2A	.2B	.2C	.2D	.2E	.2F
.000	.12500000	.12890625	.13281250	.13671875	.14062500	.14453125	.14843750	.15234375	.15625000	.16015625	.16406250	.16796875	.17187500	.17578125	.17968750	.18359375
.001	.12524414	.12915039	.13305664	.13696289	.14086914	.14477539	.14868164	.15258789	.15649414	.16040039	.16430664	.16821289	.17211914	.17602539	.17993164	.18383789
.002	.12548828	.12939453	.13330078	.13720703	.14111328	.14501953	.14892578	.15283203	.15673828	.16064453	.16455078	.16845703	.17236328	.17626953	.18017578	.18408203
.003	.12573242	.12963867	.13354492	.13745117	.14135742	.14526367	.14916992	.15307617	.15698242	.16088867	.16479492	.16870117	.17260742	.17651367	.18041992	.18432617
.004	.12597656	.12988281	.13378906	.13769531	.14160156	.14550781	.14941406	.15332031	.15722656	.16113281	.16503906	.16894531	.17285156	.17675781	.18066406	.18457031
.005	.12622070	.13012695	.13403320	.13793945	.14184570	.14575195	.14965820	.15356445	.15747070	.16137695	.16528320	.16918945	.17309570	.17700195	.18090820	.18481445
.006	.12646484	.13037109	.13427734	.13818359	.14208984	.14599609	.14990234	.15380859	.15771484	.16162109	.16552734	.16943359	.17333984	.17724609	.18115234	.18505859
.007	.12670898	.13061523	.13452148	.13842773	.14233398	.14624023	.15014648	.15405273	.15795898	.16186523	.16577148	.16967773	.17358398	.17749023	.18139648	.18530273
.008	.12695313	.13085938	.13476563	.13867188	.14257813	.14648438	.15039063	.15429688	.15820313	.16210938	.16601563	.16992188	.17382813	.17773438	.18164063	.18554688
.009	.12719727	.13110352	.13500977	.13891602	.14282227	.14672852	.15063477	.15454102	.15844727	.16236352	.16625977	.17016602	.17407227	.17797852	.18188477	.18579102
.00A	.12744141	.13134766	.13525391	.13916016	.14306641	.14697266	.15087891	.15478516	.15869141	.16259766	.16650391	.17041016	.17431641	.17822266	.18212891	.18603516
.00B	.12768555	.13159180	.13549805	.13940430	.14331055	.14721680	.15112305	.15502930	.15893555	.16284180	.16674805	.17065430	.17456055	.17846680	.18237305	.18627930
.00C	.12792969	.13183594	.13574219	.13964844	.14355469	.14746094	.15136719	.15527344	.15917969	.16308594	.16699219	.17089844	.17480469	.17871094	.18261719	.18652344
.00D	.12817383	.13208008	.13598633	.13989258	.14379883	.14770508	.15161133	.15551758	.15942383	.16333008	.16723633	.17114258	.17504883	.17895508	.18286133	.18676758
.00E	.12841797	.13232422	.13623047	.14013672	.14404297	.14794922	.15185547	.15576172	.15966797	.16357422	.16748047	.17138672	.17529297	.17919922	.18310547	.18701172
.00F	.12866211	.13256836	.13647461	.14038086	.14428711	.14819336	.15209961	.15600586	.15991211	.16381836	.16772461	.17163086	.17553711	.17944336	.18334961	.18725586
	.30	.31	.32	.33	.34	.35	.36	.37	.38	.39	.3A	.3B	.3C	.3D	.3E	.3F
.000	.18750000	.19140625	.19531250	.19921875	.20312500	.20703125	.21093750	.21484375	.21875000	.22265625	.22656250	.23046875	.23437500	.23828125	.24218750	.24609375
.001	.18774414	.19165039	.19555664	.19946289	.20336914	.20727539	.21118164	.21508789	.21899414	.22290039	.22680664	.23071289	.23461914	.23852539	.24243164	.24633789
.002	.18798828	.19189453	.19580078	.19970703	.20361328	.20751953	.21142578	.21533203	.21923828	.22314453	.22705078	.23095703	.23486328	.23876953	.24267578	.24658203
.003	.18823242	.19213867	.19604492	.19995117	.20385742	.20776367	.21166992	.21557617	.21948242	.22338867	.22729492	.23120117	.23510742	.23901367	.24291992	.24682617
.004	.18847656	.19238281	.19628906	.20019531	.20410156	.20800781	.21191406	.21582031	.21972656	.22363281	.22753906	.23144531	.23535156	.23925781	.24316406	.24707031
.005	.18872070	.19262695	.19653320	.20043945	.20434570	.20825195	.21215820	.21606445	.21997070	.22387695	.22778320	.23168945	.23559570	.23950195	.24340820	.24731445
.006	.18896484	.19287109	.19677734	.20068359	.20458984	.20849609	.21240234	.21630859	.22021484	.22412109	.22802734	.23193359	.23583984	.23974609	.24365234	.24755859
.007	.18920898	.19311523	.19702148	.20092773	.20483398	.20874023	.21264648	.21655273	.22045898	.22436523	.22827148	.23217773	.23608398	.23999023	.24389648	.24780273
.008	.18945313	.19335938	.19726563	.20117188	.20507813	.20898438	.21289063	.21679688	.22070313	.22460938	.22851563	.23242188	.23632813	.24023438	.24414063	.24804688
.009	.18969727	.19360352	.19750977	.20141602	.20532227	.20922852	.21313477	.21704102	.22094727	.22485352	.22875977	.23266602	.23657227	.24047852	.24438477	.24829102
.00A	.18994141	.19384766	.19775391	.20166016	.20556641	.20947266	.21337891	.21728516	.22119141	.22509766	.22900391	.23291016	.23681641	.24072266	.24462891	.24853516
.00B	.19018555	.19409180	.19799805	.20190430	.20581055	.20971680	.21362305	.21752930	.22143555	.22534180	.22924805	.23315430	.23706055	.24096680	.24487305	.24877930
.00C	.19042969	.19433594	.19824219	.20214844	.20605469	.20996094	.21386719	.21777344	.22167969	.22558594	.22949219	.23339844	.23730469	.24121094	.24511719	.24902344
.00D	.19067383	.19458008	.19848633	.20239258	.20629883	.21020508	.21411133	.21801758	.22192383	.22583008	.22973633	.23364258	.23754883	.24145508	.24536133	.24926758
.00E	.19091797	.19482422	.19873047	.20263672	.20654297	.21044922	.21435547	.21826172	.22216797	.22607422	.22998047	.23388672	.23779297	.24169922	.24560547	.24951172
.00F	.19116211	.19506836	.19897461	.20288086	.20678711	.21069336	.21459961	.21850586	.22241211	.22631836	.23022461	.23413086	.23803711	.24194336	.24584961	.24975586

Decimal to Hexadecimal Conversion Information (con't)

	.40	.41	.42	.43	.44	.45	.46	.47	.48	.49	.4A	.4B	.4C	.4D	.4E	4F
.000	.25000000	.25390625	.25781250	.26171875	.26562500	.26953125	.27343750	.27734375	.28125000	.28515625	.28906250	.29296875	.29687500	.30078125	.30468750	.30859375
.001	.25024414	.25415039	.25805664	.26196289	.26586914	.26977539	.27368164	.27758789	.28149414	.28540039	.28930664	.29321289	.29711914	.30102529	.30493164	.30883789
.002	.25048828	.25439453	.25830078	.26220703	.26611328	.27001953	.27392578	.27783203	.28173828	.28564453	.28955078	.29345703	.29736328	.30126953	.30517578	.30908203
.003	.25073242	.25463867	.25854492	.26245117	.26635742	.27026367	.27416992	.27807617	.28198242	.28588867	.28979492	.29370117	.29760742	.30151367	.30541992	.30932617
.004	.25097656	.25488281	.25878906	.26269531	.26660156	.27050781	.27441406	.27832031	.28222656	.28613281	.29003906	.29394531	.29785156	.30175781	.30566406	.30957031
.005	.25122070	.25512695	.25903320	.26293945	.26684570	.27075195	.27465820	.27856445	.28247070	.28637695	.29028320	.29418945	.29809570	.30200195	.30590820	.30981445
.006	.25146484	.25537109	.25927734	.26318359	.26708984	.27099609	.27490234	.27880859	.28271484	.28662109	.29052734	.29443359	.29833984	.30224609	.30615234	.31005859
.007	.25170898	.25561523	.25952148	.26342773	.26733398	.27124023	.27514648	.27905273	.28295898	.28686523	.29077148	.29467773	.29858398	.30249023	.30639648	.31030273
.008	.25195313	.25585938	.25976563	.26367188	.26757813	.27148438	.27539063	.27929688	.28320313	.28710938	.29101563	.29492188	.29882813	.30273438	.30664063	.31054688
.009	.25219727	.25610352	.26000977	.26391602	.26782227	.27172852	.27563477	.27954102	.28344727	.28735352	.29125977	.29516602	.29907227	.30297852	.30688477	.31079102
.00A	.25244141	.25634766	.26025391	.26416016	.26806641	.27197266	.27587891	.27978516	.28369141	.28759766	.29150390	.29541016	.29931641	.30322266	.30712891	.31103516
.00B	.25268555	.25659180	.26049805	.26440430	.26831055	.27221680	.27612305	.28002930	.28393555	.28784180	.29174805	.29565430	.29956055	.30346680	.30737305	.31127930
.00C	.25292969	.25683594	.26074219	.26464844	.26855469	.27246094	.27636719	.28027344	.28417969	.28808594	.29199219	.29589844	.29980469	.30371094	.30761719	.31152344
.00D	.25317383	.25708008	.26098633	.26489258	.26879883	.27270508	.27661133	.28051758	.28442383	.28833008	.29223633	.29614258	.30004883	.30395508	.30786133	.31176758
.00E	.25341797	.25732422	.26123047	.26513672	.26904297	.27294922	.27685547	.28076172	.28466797	.28857422	.29248047	.29638672	.30029297	.30419922	.30810547	.31201172
.00F	.25366211	.25756836	.26147461	.26538086	.26928711	.27319336	.27709961	.28100586	.28491211	.28881836	.29272461	.29663086	.30053711	.30444336	.30834961	.31225586
	.50	.51	.52	.53	.54	.55	.56	.57	.58	.59	.5A	.5B	.5C	.5D	.5E	.5F
.000	.31250000	.31640625	.32031250	.32421875	.32812500	.33203125	.33593750	.33984375	.34375000	.34765625	.35156250	.35546875	.35937500	.36328125	.36718750	.37109375
.001	.31274414	.31665039	.32055664	.32446289	.32836914	.33227539	.33618164	.34008789	.34399414	.34790039	.35180664	.35571289	.35961914	.36352539	.36743164	.37133789
.002	.31298828	.31689453	.32080078	.32470703	.32861328	.33251953	.33642578	.34033203	.34423828	.34814453	.35205078	.35595703	.35986328	.36376953	.36767578	.37158203
.003	.31323242	.31713867	.32104492	.32495117	.32885742	.33276367	.33666992	.34057617	.34448242	.34838867	.35229492	.35620117	.36010742	.36401367	.36791992	.37182617
.004	.31347656	.31738281	.32128906	.32519531	.32910156	.33300781	.33691406	.34082031	.34472656	.34863281	.35253906	.35644531	.36035156	.36425781	.36816406	.37207031
.005	.31372070	.31762695	.32153320	.32543945	.32934570	.33325195	.33715820	.34106445	.34497070	.34887695	.35278320	.35668945	.36059570	.36450195	.36840820	.37231445
.006	.31396484	.31787109	.32177734	.32568359	.32958984	.33349609	.33740234	.34130859	.34521484	.34912109	.35302734	.35693359	.36083984	.36474609	.36865234	.37255859
.007	.31420898	.31811523	.32202148	.32592773	.32983398	.33374023	.33764648	.34155273	.34545898	.34936523	.35327148	.35717773	.36108398	.36499023	.36889648	.37280273
.008	.31445313	.31835938	.32226563	.32617188	.33007813	.33398438	.33789063	.34179688	.34570313	.34960938	.35351563	.35742188	.36132813	.36523438	.36914063	.37304688
.009	.31469727	.31860352	.32250977	.32641602	.33032227	.33422852	.33813477	.34204102	.34594727	.34985352	.35375977	.35766602	.36157227	.36547852	.36938477	.37329102
.00A	.31494141	.31884766	.32275391	.32666016	.33056641	.33447266	.33837891	.34228516	.34619141	.35009766	.35400391	.35791016	.36181641	.36572266	.36962891	.37353516
.00B	.31518555	.31909180	.32299805	.32690430	.33081055	.33471680	.33862305	.34252930	.34643555	.35034180	.35424805	.35815430	.36206055	.36596680	.36987305	.37377930
.00C	.31542969	.31933594	.32324219	.32714844	.33105469	.33496094	.33886719	.34277344	.34667969	.35058594	.35449219	.35839844	.36230469	.36621094	.37011719	.37402344
.00D	.31567383	.31958008	.32348633	.32739258	.33129883	.33520508	.33911133	.34301758	.34692383	.35083008	.35473633	.35864258	.36254883	.36645508	.37036133	.37426758
.00E	.31591797	.31982422	.32373047	.32763672	.33154297	.33544922	.33935547	.34326172	.34716797	.35107422	.35498047	.35888672	.36279297	.36669922	.37060547	.37451172
.00F	.31616211	.32006836	.32397461	.32788086	.33178711	.33569336	.33959961	.34350586	.34741211	.35131836	.35522461	.35913086	.36303711	.36694336	.37084961	.37475586
	.60	.61	.62	.63	.64	.65	.66	.67	.68	.69	.6A	.6B	.6C	.6D	.6E	.6F
.000	.37500000	.37890625	.38281250	.38671875	.39062500	.39453125	.39843750	.40234375	.40625000	.41015625	.41406250	.41796875	.42187500	.42578125	.42968750	.43359375
.001	.37524414	.37915039	.38305664	.38696289	.39086914	.39477539	.39868164	.40258789	.40649414	.41040039	.41430664	.41821289	.42211914	.42602539	.42993164	.43383789
.002	.37548828	.37939453	.38330078	.38720703	.39111328	.39501953	.39892578	.40283203	.40673828	.41064453	.41455078	.41845703	.42236328	.42626953	.43017578	.43408203
.003	.37573242	.37963867	.38354492	.38745117	.39135742	.39526367	.39916992	.40307617	.40698242	.41088867	.41479492	.41870117	.42260742	.42651367	.43041992	.43432617
.004	.37597656	.37988281	.38378906	.38769531	.39160156	.39550781	.39941406	.40332031	.40722656	.41113281	.41503906	.41894531	.42285156	.42675781	.43066406	.43457031
.005	.37622070	.38012695	.38403320	.38793945	.39184570	.39575195	.39965820	.40356445	.40747070	.41137695	.41528320	.41918945	.42309570	.42700195	.43090820	.43481445
.006	.37646484	.38037109	.38427734	.38818359	.39208984	.39599609	.39990234	.40380859	.40771484	.41162109	.41552734	.41943359	.42333984	.42724609	.43115234	.43505859
.007	.37670898	.38061523	.38452148	.38842773	.39233398	.39624023	.40014648	.40405273	.40795898	.41186523	.41577148	.41967773	.42358398	.42749023	.43139648	.43530273
.008	.37695313	.38085938	.38476563	.38867188	.39257813	.39648438	.40039063	.40429688	.40820313	.41210938	.41601563	.41992188	.42382813	.42773438	.43164063	.42554688
.009	.37719727	.38110352	.38500977	.38891602	.39282227	.39672852	.40063477	.40454102	.40844727	.41235352	.41625977	.42016602	.42407227	.42797852	.43188477	.43579102
.00A	.37744141	.38134766	.38525391	.38916016	.39306641	.39697266	.40087891	.40478516	.40869141	.41259766	.41650391	.42041016	.42431641	.42822266	.43212891	.43603516
.00B	.37768555	.38159180	.38549805	.38940430	.39331055	.39721680	.40112305	.40502930	.40893555	.41284180	.41674805	.42065430	.42456055	.42846680	.43237305	.43627930
.00C	.37792969	.38183594	.38574219	.38964844	.39355469	.39746094	.40136719	.40527344	.40917969	.41308594	.41699219	.42089844	.42480469	.42871094	.43261719	.43652344
.00D	.37817383	.38208008	.38598633	.38989258	.39379883	.39770508	.40161133	.40551758	.40942383	.41333008	.41723633	.42114258	.42504883	.42895508	.43286133	.43676758
.00E	.37841797	.38232422	.38623047	.39013672	.39404297	.39794922	.40185547	.40576172	.40966797	.41357422	.41748047	.42138672	.42529297	.42919922	.43310547	.43701172
.00F	.37866211	.38256836	.38647461	.39038086	.39428711	.39819336	.40209961	.40600586	.40991211	.41381836	.41772461	.42163086	.42553711	.42944336	.43334961	.43725586
	.70	.71	.72	.73	.74	.75	.76	.77	.78	.79	.7A	.7B	.7C	.7D	.7E	.7F
.000	.43750000	.44140625	.44531250	.44921875	.45312500	.45703125	.46093750	.46484375	.46875000	.47265625	.47656250	.48046875	.48437500	.48828125	.49218750	.49609375
.001	.43774414	.44165039	.44555664	.44946289	.45336914	.45727539	.46118164	.46508789	.46899414	.47290039	.47680664	.48071289	.48461914	.48852539	.49243164	.49633789
.002	.43798828	.44189453	.44580078	.44970703	.45361328	.45751953	.46142578	.46533203	.46923828	.47314453	.47705078	.48095703	.48486328	.48876953	.49267578	.49658203
.003	.43823242	.44213867	.44604492	.44995117	.45385742	.45776367	.46166992	.46557617	.46948242	.47338867	.47729492	.48120117	.48510742	.48901367	.49291992	.49682617
.004	.43847656	.44238281	.44628906	.45019531	.45410156	.45800781	.46191406	.46582031	.46972656	.47363281	.47753906	.48144531	.48535156	.48925781	.49316406	.49707031
.005	.43872070	.44262695	.44653320	.45043945	.45434570	.45825195	.46215820	.46606445	.46997070	.47387695	.47778320	.48168945	.48559570	.48950195	.49340820	.49731445
.006	.43896484	.44287109	.44677734	.45068359	.45458984	.45849609	.46240234	.46630859	.47021484	.47412109	.47802734	.48193359	.48583984	.48974609	.49365234	.49755859
.007	.43920898	.44311523	.44702148	.45092773	.45483398	.45874023	.46264648	.46655273	.47045898	.47436523	.47827148	.48217775	.48608398	.48999023	.49389648	.49780273
.008	.43945313	.44335938	.44726563	.45117188	.45507813	.45898438	.46289063	.46679688	.47070313	.47460938	.47851563	.48242188	.48632813	.49023438	.49414063	.49804688
.009	.43969727	.44360352	.44750977	.45141602	.45532227	.45922852	.46313477	.46704102	.47094727	.47485352	.47875977	.48266602	.48657227	.49047852	.49438477	.49829102
.00A	.43994141	.44384766	.44775391	.45166016	.45556641	.45947266	.46337891	.46728516	.47119141	.47509766	.47900391	.48291016	.48681641	.49072266	.49462891	.49853516
.00B	.44018555	.44409180	.44799805	.45190430	.45581055	.45971680	.46362305	.46752930	.47143555	.47534180	.47924805	.48315430	.48706055	.49096680	.49487305	.49877930
.00C	.44042969	.44433594	.44824219	.45214844	.45605469	.45996094	.46386719	.46777344	.47167969	.47558594	.47949219	.48339844	.48730469	.49121094	.49511719	.49902344
.00D	.44067383	.44458008	.44848633	.45239258	.45629883	.46020508	.46411133	.46801758	.47192383	.47583008	.47973633	.48364258	.48754883	.49145508	.49536133	.49926758
.00E	.44091797	.44482422	.44873047	.45263672	.45654297	.46044922	.46435547	.46826172	.47216797	.47607422	.47998047	.48388672	.48779297	.49169922	.49560547	.49951172
.00F	.44116211	.44506836	.44897461	.45288086	.45678711	.46069336	.46459961	.46850586	.47241211	.47631836	.48022461	.48413086	.48803711	.49194336	.49584961	.49975586

Source Publications

Additional information about linkage registers is in *Supervisor Services and Macros*, GC27-6979.

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/360 and System/370 Bibliography*. GA22-6822 for a list of these publications.

You can obtain additional information about completion codes from *OS/VS Message Library: VS1 System Codes*, GC38-1003 or *OS/VS Message Library: VS2 System Codes*, GC38-1008.

Save Area Format

Word	Contents
1	Used by PL/1 language-compiled program
2	Address of previous save area (stored by called program)
3	Address of next save area (stored by calling program)
4	Register 14 (return address)
5	Register 15 (entry-point address)
6	Register 0
7	Register 1
8	Register 2
9	Register 3
10	Register 4
11	Register 5
12	Register 6
13	Register 7
14	Register 8
15	Register 9
16	Register 10
17	Register 11
18	Register 12

Linkage Register Conventions

Linkage Register Conventions	
Reg	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

BYTE 0

DEVICE \ BIT	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 PTY 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	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WRD CNT 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			
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	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN		
3410, 3411	CMD REJ	INT REQ	BUS OUT	EQ CHK	DATA CHK	OVER-RUN	WRD CNT 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		ABN FORMAT RESET	PERM ERR (BYPASS KEY)

UCB Sense Information (con't)

BYTE 1

DEVICE \ BIT	0	1	2	3	4	5	6	7
1287	TAPE MODE	LATE STKR SELECT	NO DOC FOUND		INVAL OP			
1288		END OF PAGE	NO DOC FOUND		INVAL 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	INVAL SEQ	NO REC FOUND	FILE PROT	SERVICE OVER-RUN	OVER-FLOW INL
2400	NOISE	00-NON-XST TU 01-NOT READY 10-RDY & NO RWD 11-RDY & RWDING		7 TRK	AT LOAD POINT	WRT STATUS	FILE PROT	TAPE IND
3211	CMD RETRY	PRINT CHK	PRINT QUALITY	LINE POS	FORMS CHK	CMD SUP	MECHAN-ICAL MOTION	
3330, 3333	PERM ERR	INVLD TRK FORMAT	END OF CYL	STATE VAR PRES	NO REC FOUND	FILE PROT	WRITE INHIBIT	OPER-ATION INL
3410,3411	NOISE	TU STAT A	TU STAT B		AT LOAD POINT	WRT STAT	FILE PROT	NOT CAPA-BLE
3420, 2803	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				

UCB Sense Information (con't)

BYTE 2

DEVICE \ BIT	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	BITS 0-7 INDICATE A TRACK IS IN ERROR						6 & 7 INDICATE NO ERROR OR MULTI-ERROR	
3211	CARR FAILED TO MOVE	CARR SEQ	CARR STOP	PLATEN FAILED	PLATEN FAILED	FORMS JAM	RIBBON MOTION	TRAIN OVER-LOAD
3330, 3333		COR-RECT ABLE		ENV DATA PRESENT				
3410, 3411	TRACK IN ERROR BITS							
3420, 3803	TRACK IN ERROR BITS							
3505, 3525	USED FOR DIAGNOSTIC PURPOSES ONLY							

BYTE 3

DEVICE \ BIT	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 CFN	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	COM-PARE
3211	UCSB PARITY	PLB PARITY	FCB PARITY	COIL PROT CHK	HAM-MER FIRE	FIELD ENG	USCAR SYNC CHK	SEP SYNC CHK
3330, 3333	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	1600 BPI	BKWD	COM-PARE
3505, 3525	USED FOR DIAGNOSTIC PURPOSES ONLY							

UCB Sense Information (con't)

BYTE 4

DEVICE \ BIT	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	ECHO ERR	RES TAPE UNIT	READ CLOCK ERR	WRITE CLOCK ERR	DELAY CNTR	SEQ IND C	SEQ IND B	SEQ IND A
3211								
3330, 3333	PHYSICAL DRIVE IDENTIFICATION							
3410, 3411	TU POSIT 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

BYTE 5

DEVICE \ BIT	0	1	2	3	4	5	6	7
2250, 2260								
2305	CYLINDER ADDRESS							
2314	COMMAND IN PROGRESS WHEN OVERFLOW INCOMPLETE OCCURS							
2319								
2400	COMMAND IN PROGRESS WHEN OVERFLOW INCOMPLETE OCCURS OR ZERO							
3211								
3330, 3333	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

BYTE 6

DEVICE \ BIT	0	1	2	3	4	5	6	7
2305	CURRENT HEAD ADDRESS							
3330, 3333	REVERSE	CYL HIGH	DIFFER HIGH	HEAD ADDRESS				
3410, 3411		SHRT GAP	DUAL DENSITY	ALT DENSITY	TAPE UNIT MODEL			
3420, 3803	7 TRK	WRT	DUAL	NRZI	TAPE UNIT MODEL DEFINED			

UCB Sense Information (con't)

BYTE 7

BIT / DEVICE	0	1	2	3	4	5	6	7
2305	ENCODED ERROR MESSAGE							
3330, 3333	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

BYTE 8

BIT / DEVICE	0	1	2	3	4	5	6	7
3410, 3411		WRT FEED THRU CHK		END VEL CHK	RD BK DATA NOT DET	START VEL CHK		MAR-GINAL VELOC
3420, 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

BYTE 9

BIT / DEVICE	0	1	2	3	4	5	6	7
3420, 3803	JDR CNTR	VLCTY CHNG ON WRT	SDR COUNTERS					TAPE CTL RESD

BYTE 10

BIT / DEVICE	0	1	2	3	4	5	6	7
3420, 3803	CMD STATUS REJ		CNTRL STATUS REJ	NO BLK ON RCD RD BKCK	WTM NOT DETECT	TACH START FAIL		VELO-CITY CHK

BYTE 11

BIT / DEVICE	0	1	2	3	4	5	6	7
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

UCB Sense Information (con't)

Byte 12

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	B BUS PAR ERR 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

Byte 13

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	CONTROL UNIT DENSITY		CONTROL UNIT UNIQUE ID - HIGH ORDER					

Byte 14

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	CONTROL UNIT UNIQUE ID - LOW ORDER							

Byte 15

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	TAPE UNIT UNIQUE ID - HIGH ORDER							

UCB Sense Information (con't)

Byte 16

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	TAPE UNIT UNIQUE ID - LOW ORDER							

Byte 17

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	2 CHAN SWTCH	CONTROL UNIT WITH DEVICE SWITCH FEATURES			EC LEVEL OF TAPE CONTROL UNIT			

Byte 18

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

Byte 19

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	PRIMED FOR DEVICE END TU 7	TU 6	TU 5	TU 4	TU 3	TU 2	TU 1	TU 0

UCB Sense Information (con't)

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

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

Byte 22

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	FRU IDENTIFIERS FOR ALU 1							

Byte 23

DEVICE \ BIT	0	1	2	3	4	5	6	7
3420, 3803	FRU IDENTIFIERS FOR ALU 2							

OS/VS1 Completion Code Summary

Group	Completion Code	Operation or Macro Instruction	Explanation
BISAM/ BSAM/ QSAM/ BDAM	001	CHECK, GET, PUT	I/O error; terminate specified or no SYNAD specified.
BSAM/ QSAM/ QISAM/ ISAM	002		Record is greater than 32,768 bytes, exceeds maximum track length or stated block size; block could not be contained in one extent; too many tracks specified for cylinder overflow; BDW or RDW (SDW) invalid; record to be transferred larger than track capacity.
BSAM/ QSAM/	003	EOB for 3525	3525 associated data set I/O sequence error.
	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.
BSAM	008	CHECK while creating data set	SYNAD returned to CHECK routine, but save area was destroyed.
BDAM	020	OPEN	Invalid DCBMACRF field.
	025		Address in DCBSQND field outside task.
	026	Processing with exclusive control	Invalid DCBXARG field or exclusive control status not indicated.
BISAM/ QISAM	030	OPEN	Invalid DCBMACRF field.
	032	OPEN	Invalid DCBMACRF field.
	033	OPEN	I/O error in reading highest level index or while reading the last prime data block or in validating last record pointers; address in DCBMSHI field outside task or under incorrect protection key.
	036	OPEN	No prime area specified.
	037	OPEN	User supplied buffers to small.
	03A	CLOSE	I/O error writing updated format 2 DSCB.
BISAM	034	OPEN	DCBSMSI field specifies area too small for highest level index; invalid address in DCBMSWA.
	035	OPEN	DCBSMSW and DCBMSWA fields specify area too small for one track.
QISAM	031		QISAM I/O error; no SYNAD specified.
	038	OPEN	Index area too small or crosses volumes.
	039	Scanning	End of data set; no exit routine address in DCBEODAD field.
	03B	OPEN	ISAM data set to be processed, but not created or its DCB not closed after creation; invalid DCBRKP field; DCBKEYLE field was zero; OPEN macro not issued for output; BLKSIZE or LRECL specified incorrectly.
	03E	OPEN	No space available for resume loading.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
BDAM/ BISAM/ QISAM	03D	OPEN	Missing format 2 DSCB, or serial numbers for SER in DD not in order or not all present.
TCAM	040	OPEN	Error in opening a TCAM line group data set.
	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.
	046	CLOSE	TCAM MCP is scheduled to be terminated, application program data set is active. Completion code is for the application program data set.
Graphics access method	056	Graphics attention service routine	ANALYZ or GSERV specified DCB, which pointed to DEB, which pointed to invalid UCB.
(GAM)	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 for GACB that was not specified (via SPAR) for 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 pressed alphameric keyboard CANCEL key and selected DUMP or TERMINATE option to terminate program.
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	DCBBFTEK field specified dynamic buffer allocation, but DCBBUFCB, DCBBUFNO, and DCBBUFL fields not specified.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
BTAM (cont'd)	097	OPEN	Device I/O directory full.
	098	OPEN	Transmission control unit not a 2701 or the Dual Communication Interface, or Dual Code Feature not specified in UCB.
Job sched-uler	0B0		I/O error in reading or writing SYS1.SYSJOBQE or SWADS.
	1B0		Invalid TTR for SYS1.SYSJOBQE found by system conversion routine.
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.
	0F1		Program interrupt in I/O interrupt handler.
	0F2		Program interrupt in type 1 SVC routine.
	0F3		Machine-check interrupt; MCH able to abnormally terminate job step and continue operating system.
	0F5		Program interrupt occurred while loading transient area for type 3 or 4 SVC.
EXCP (SVC 00)	100	I/O Operation	Device not operational.
	200	I/O Operation	Invalid ECB, IOB, DCB protect key.
	300	I/O Operation	Invalid DEB protect key; not enough extents in DEB.
	400	I/O Operation	Invalid DCB pointers.
	500	I/O Operation	Invalid UCB address.
WAIT (SVC 01)	101	WAIT	More events than ECBs.
	201	WAIT	Invalid ECB address.
	301	WAIT	ECB wait flag already on.
POST (SVC 02)	102	POST	Invalid ECB address.
	202	POST	Invalid RB address in ECB.
Task termin-ation (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.
	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.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
GETMAIN (SVC 04)	604	GETMAIN	Address in A or LA operand is outside task; address of parameter list invalid.
	704	GETMAIN	List request; not VS2 system.
	804	GETMAIN	Request for zero bytes of virtual storage or not enough virtual storage available.
	B04	GETMAIN	Subpool number greater than 127.
	E04	GETMAIN from program in super-visor mode	Not enough SQA available.
FREEMAIN (SVC 05)	605	FREEMAIN	Address in A or LA operand is outside task; address of parameter list invalid.
	705	FREEMAIN	List request; not VS2 system.
	905	FREEMAIN	Address of area to be freed not multiple of 8.
	A05	FREEMAIN	Area to be freed overlaps existing free area.
	B05	FREEMAIN	Subpool number greater than 127.
Contents supervi-sor (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; module specified by entry point defined by IDENTIFY macro.
	506	LINK, LOAD, ATTACH, XCTL	Not enough virtual storage for module and overlay supervisor.
	606	LINK, LOAD, ATTACH, XCTL	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 or I/O error during directory search.
	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		Abnormally terminating transient area task reinstated; user task abnormally terminated.
XCTL (SVC 07)	207	XCTL	Asynchronous exit routine attempted to execute XCTL.
LOAD (SVC 08)	308	LOAD	Module specified by entry point defined by IDENTIFY macro.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
GETMAIN, FREEMAIN with R operand (SVC 0A)	60A	FREEMAIN	Invalid specification of an area to be freed; address of area to be freed (in register 1) not multiple of 8.
	80A	GETMAIN	Request for zero bytes of virtual storage or not enough virtual storage available.
	90A	FREEMAIN	Address of area to be freed not a multiple of 8.
	A0A	FREEMAIN	Area to be freed overlaps an existing free area.
	B0A	GETMAIN, FREEMAIN	Subpool number greater than 127.
ABEND (SVC 0D)	D0D	ABEND	Invalid ABEND recursion during abnormal termination of subtask; job step task 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 not found; no directory allocation subparameter in DD.
	113	OPEN, OPEN with TYPE=J	I/O error in reading or writing JFCB or in reading JFCB extension block; no exit code provided.
	213	OPEN	DSCB not found; I/O error in reading or writing DSCB; unable to locate PASSWORD data set.
	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; ASCII tape accessibility error, ASCII tape security error.
	A13	OPEN	File sequence number in LABEL in DD incorrect.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
OPEN (SVC 13) (cont'd)	B13	OPEN for UCS printer	Operator cancelled UCS load; incorrect UCS image; space unavailable for DCB and DEB; SYS1.IMAGELIB not mounted or cataloged; permanent I/O error detected.
	C13	OPEN	I/O error in reading JFCB or DSCB for concatenated data set; DSCB not found for one data set in concatenation; graphic device already opened by another task; output data sets concatenated.
	D13	OPEN for graphics	DCB for other than graphics device.
	E13	OPEN for graphics	DCBGNCP field not 1 through 99.
CLOSE (SVC 14)	214	CLOSE	I/O error in tape positioning or volume disposition.
	314	CLOSE	I/O error reading DSCB.
	414	CLOSE	I/O error writing DSCB.
	514	CLOSE	I/O error reading JFCB.
	614	CLOSE	I/O error writing file mark.
	714	CLOSE	I/O error processing label, or tape mark.
	A14	CLOSE	I/O error releasing unused direct access 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	Graphic device not opened by closing task.
TCLOSE (SVC 17)	117	BSAM CLOSE with TYPE=T	I/O error in tape positioning 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 or tape mark.
Master scheduler (SVC 22)	122		Operator cancelled job; requested dump.
	222		Operator cancelled job; did not request dump.
	322		Execution of job step or cataloged procedure taking longer than time specified.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
Master scheduler (SVC 22) (cont'd)	422		Job required too much queue space for initiation.
	522		All tasks in SVC wait state for 30 consecutive minutes or for time specified in JWT parameter (in systems with SMF).
WTO/ WTOR (SVC 23)	D23	WTO, WTOR	Parameter list not begin on proper boundary; no buffers available; text length equal to or less than zero (WTOR only).
	E23	Reply command processing (in response to WTOR)	Invalid ECB, RB, or reply address.
EXTRACT (SVC 28)	128	EXTRACT	Output list not on fullword boundry 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, or not within bounds of partition.
	62A	ATTACH	Exceeded allowed number of tasks.
CHAP (SVC 2C)	12C	CHAP	Address for subtask TCB does not point to TCB of immediate subtask, or points to a task that has terminated.
	22C	CHAP	Address for subtask TCB not multiple of 4.
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.
DEQ (SVC 30)	130	DEQ without RET=HAVE	DEQ for resource not enqueued by prior ENQ.
	230	DEQ	Invalid length specified for name of resource.
	330	DEQ	Invalid option specified by task with non-zero protection key.
	430	DEQ	Invalid parameter list.
	530	DEQ	Task does not yet control specified resource.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
EOV (SVC 37)	137	End of volume	I/O error in label or tape mark processing or tape mark positioning.
	237	End of volume	Verification error in label processing; tape label block count not same as DCB block count.
	337	End of data set	No address specified in DCBEODAD field.
	437	End of volume	Protect key different in TCBPKF field of TCB and DEBDEBID field of DEB.
	637	End of volume	I/O error in 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	Direct access I/O error; DSCB not found for multi-volume or concatenated data set.
	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.
	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 for SPACE in DD.
	E37	Output operation	More space needed but not enough volumes specified in SER, volume count, or REF in DD.
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	Invalid option specified by task with non-zero protection key.
	438	ENQ	Invalid parameter list.
DETACH (SVC 3E)	13E	DETACH	Subtask being detached not yet terminated.
	23E	DETACH	TCB address not on word boundary; subtask TCB not on word boundary; subtask TCB=0 or not an immediate subtask.
CHKPT	13F		Error during execution of checkpoint restart.
RDJFCB	140	RDJFCB	I/O error in reading JFCB.
	240	RDJFCB	No foundation extension block in DCB; EXLST address in DCB; JFCB exit in DCB exit list; JFCB buffer not in user's virtual storage.
SWAP (SVC 55)	155		SVC 85 (in decimal) issued by user's task, but is restricted for use by Dynamic Device Reconfiguration.

OS/VS1 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
System Restart	2F3		Job was being executed when system failure occurred; a system restart was performed.
Supervisor Call (SVC nn)	Fnn		Invalid operand, nn, in SVC instruction.
ESR (SVCs 109, 116, 117)	16D		Invalid ESR code in register 15.
Paging	028		Page file I/O error.
	0D0		Invalid segment translation interrupt.
	0D1		Invalid page-fault interrupt.
Misc.	160		Attention exit routine invalid.
	2FF	ABEND appendage	Step terminated at request of user appendage III.
	622		Execution of a task entered from a TSO terminal was stopped for one of these reasons: error while constructing control blocks for TSO; operator issued STOP TSO; terminal user signaled ATTN; job submitter disconnected his terminal from the system.
	722		OUTLIM keyword specified on SYSOUT DD statement exceeded.
	822		V=R region not obtained.

OS/VS2 Completion Code Summary

Group	Completion Code	Operation or Macro Instruction	Explanation
BISAM/ BSAM/ QSAM/ BDAM	001	CHECK, GET, PUT	I/O error; terminate specified or no SYNAD specified
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
BSAM/ QSAM	003	EOB for 3525	3525 associated data set I/O sequence error
	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
BSAM	008	CHECK while creating data set	SYNAD returned to CHECK routine, but save area was destroyed
BDAM	020	OPEN	Invalid DCBMACRF field or data set contained zero extents
	025		Address in DCBSQND field outside task
	026	Processing with exclusive control	Invalid DCBXARG field or exclusive control status not indicated
BDAM/ BISAM/ QISAM	03D	OPEN	Missing format 2 DSCB or serial numbers not in order or not all present
BISAM/ QISAM	030	OPEN	Invalid DCBMACRF field
	032	OPEN	Invalid DCBMACRF field.
	033	OPEN	I/O error reading highest level index or while reading the last prime data block or validating last record pointers; address in DCBMSHI field outside task or under incorrect protection key
	036	OPEN	No prime area specified
	037	OPEN	User supplied buffers too small
	03A	CLOSE	I/O error writing updated format 2 DSCB
BISAM	034	OPEN	DCBSMSI field specifies area too small for highest level index; invalid address in DCBMSWA
	035	OPEN	DCBSMSW fields specify area to small for one track
QISAM	031		QISAM I/O error; no SYNAD specified

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
QISAM (con't)	038	OPEN	Index area too small or crosses volumes
	039	Scanning	End of data set; no address in DCBEODAD field
	038	OPEN	ISAM data set to be processed, but not created or its DCB not closed after creation; invalid DCBRKP field; DCBKEYLE field was zero; OPEN macro not issued for output; BLKSIZE or LRECL specified incorrectly.
	03E	OPEN	No space available for resume loading
TCAM	040	OPEN	Error in opening a TCAM line group data set
	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 (TOTE)
	043	OPEN	Error in opening a TCAM application program data set
	044	Processing	Error in processing the FE Common Write (COMWRITE)
	045	Message Control Program (MCP)	I/O error or logical read error
	046	CLOSE	TCAM MCP is scheduled to be terminated, application program data set active. Completion code is for the application program data set.
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 for GACB that was not specified (via SPAR) by the closing task
	062	Graphics Subroutine Package for FORTRAN IV, COBOL, and PL/I	Return code equal to or greater than absolute value of null argument produced
	063	Execution of a graphic program	2250 operator pressed alphameric keyboard CANCEL key and selected DUMP or TERMINATE option to terminate program
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

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
BTAM (con't)	093	OPEN	UCB specified invalid or unsupported type of 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	DCBBFTEK field specified dynamic buffer allocation, but DCBBUFCB,DCBBUFNO, and DCBBUFL fields not specified
	097	OPEN	Device I/O directory full
	098	OPEN	Transmission control unit not a 2701 or the Dual Communication Interface, or Dual Code Feature not specified in UCB
Job Scheduler	00B	Scheduler Initialization Program	System failure or ABEND (dump given)
	00D	Scheduler Initialization Program	System failure or ABEND (no dump given)
	0B0		I/O error in reading or writing SYS1.SYSJOBQE or SWADS
	1B0		Invalid TTR for SYS1.SYSJOBQE found by system conversion routine
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
	0F1		Program interrupt in I/O interrupt handler
	0F2		Program interrupt in type 1 SVC routine or SVC first level interrupt handler (SVC FLIH)
	0F3		Machine-check interrupt; MCH able to abnormally terminate job step and continue operating system
	0F7		Program interruption while processing another program interruption
EXCP (SVC 00)	100	I/O operation	Device not operational or invalid use of pseudo device
	200	I/O operation	Invalid ECB, IOB,DCB protect key
	300	I/O operation	Invalid DEB protect key; not enough extents in DEB.
	400	I/O operation	Invalid DCB pointers
	500	I/O operation	Invalid UCB address
	600	I/O operation	Invalid EXCPVR request

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
	800	I/O operation	Error on fix operation or page fault occurred within IOS when it should not
	900	I/O operation	A TIC could not be inserted in construction of a real channel program
	A00	I/O operation	PCI modify function required a page to be fixed or invalid IAL entry in PCI appendage
	B00	I/O operation	Overfix threshold exceeded and pages could not be fixed
	C00	I/O operation	Number of page fix entries greater than 7
	D00	I/O operation	A reference could not be found in the validity map
WAIT (SVC 01)	101	WAIT	More events than ECBs
	201	WAIT	Invalid ECB address
	301	WAIT	ECB wait flag already on
POST (SVC 02)	102	POST	Invalid ECB address
	202	POST	Invalid RB address in ECB
Task Termination (SVC 03)	A03	RETURN or branch to return address in register 14	Subtasks not yet terminated
	C03	RETURN or branch to return address in register 14	TCBDEB points to DEB that is associated with on invalid DCB. WARNING: all data sets not closed
	E03	RETURN or branch to return address in register 14	Termination of task did not reset must-complete status
GETMAIN (SVC 04)	104	GETMAIN for program in supervisor mode	Not enough available real storage for local system queue area
	504	GETMAIN with LA and A operands	Length and address lists overlap
	604	GETMAIN	Address in A or LA operand is outside task or is not multiple of 4; address of parameter lists invalid
	804	GETMAIN or language processor	More virtual storage requested than was was available
	B04	GETMAIN	Subpool number greater than 127
FREEMAIN	305	FREEMAIN	Area to be released not within correct subpool or not described by DQE
	505	FREEMAIN with LA and A operands	Length and address lists overlap
	605	FREEMAIN	Address in A or LA operand is outside task; address of parameter list invalid
	905	FREEMAIN	Address of area to be freed not multiple of 8

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
FREEMAIN (con't)	A05	FREEMAIN	Area to be freed overlaps existing free area
	B05	FREEMAIN	Subpool number greater than 127
	D05	FREEMAIN	Attempt to free system queue space storage not owned by task
Contents Supervisor (SVC 06)	106	LINK, LOAD, ATTACH, XCTL	Error while loading module into virtual storage; invalid record type, invalid address, I/O error
	206	LINK, LOAD, XCTL, DELETE	Address of parameter list invalid, address of name or directory entry invalid, or DE (if specified) not long enough
	406	LINK, ATTACH, XCTL	Module was only loadable
Contents Supervisor (SVC 06)	506	LINK, LOAD, ATTACH, XCTL	Not enough virtual storage for module and overlay supervisor
	606	LINK, LOAD, ATTACH, XCTL	Not enough virtual storage for module; or not enough real storage to fix requested module
	706	LINK, LOAD, XCTL, DELETE	Module marked "NOT EXECUTABLE"
	806	LINK, LOAD ATTACH, XCTL	BLDL detected error; module not found or I/O error during directory search. Also, SVC routine not in LPA
	906	LINK	More than 4096 users of a reenterable or serially reusable module
	A06	LINK LOAD, XCTL, ATTACH	Task already waiting for serially reusable module
	B06	I/O activity	Abnormally terminating system error task reinstated; user task abnormally terminated
GETMAIN, FREEMAIN with R operand (SVC 0A)	10A	GETMAIN for program in supervisor mode	Not enough available storage in local system queue space
	20A	Freeing region for new job step	Storage still allocated to previous step
	30A	FREEMAIN	Area to be released not within correct subpool or not described by DQE
	40A	FREEMAIN	Attempt to release all of subpool zero storage
	80A	GETMAIN	Not enough virtual storage available
	90A	FREEMAIN	Address of area to be freed not a multiple of 8

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
GETMAIN, FREEMAIN with R operand (SVC 0A) (con't)	A0A	FREEMAIN	Area to be freed overlaps on existing free area
	B0A	GETMAIN, FREEMAIN	Subpool number greater than 127
	D0A	FREEMAIN	Attempt to free system queue area storage not owned by task
ABEND (SVC 0D)	10D		Job step task terminated
	30D		Task enqueued for a resource that become permanently unavailable
	D0D	ABEND	Invalid ABEND recursion during abnormal termination of subtask; job step terminated
SPIE (SVC 0E)	10E	SPIE	PICA address invalid
	20E	SPIE	PIE address invalid
	30E	SPIE	Unauthorized program attempted program interruption code 17
OPEN (SVC 13)	013	OPEN	Conflicting or unsupported paramters 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	DSCB not found; I/O error in reading or writing DSCB; unable to locate PASSWORD data set
	313	OPEN	I/O error 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 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; ASCII tape accessibility error, ASCII tape security error; unauthorized job step or job-step task attempted to open the VTOC for output
	A13	OPEN	File sequence number in LABEL on DD not correct

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
OPEN (SVC 13) (con't)	B13	OPEN for UCS printer	Operator canceled UCS load; permanent I/O error detected; space unavailable for DCB and DEB; SYS1.IMAGELIB not mounted or cataloged.
	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; output data sets concatenated
	D13	OPEN for graphics	DCB for other than graphics device
	E13	OPEN for graphics	DCBGNCP field not 1 through 99
CLOSE (SVC 14)	214	CLOSE	I/O error in tape positioning or volume disposition
	314	CLOSE	I/O error reading DSCB
	414	CLOSE	I/O error writing DSCB
	514	CLOSE	I/O error reading JFCB
	614	CLOSE	I/O error writing file mark
	714	CLOSE	I/O error writing label
	A14	CLOSE	I/O error releasing unused direct access 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	Graphic device not opened by closing task
TCLOSE (SVC 17)	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 or tape mark
Master Scheduler (SVC 22)	122		Operator cancelled job; requested dump
	222		Operator cancelled job; did not request dump
	322		Execution of job step or cataloged procedure taking longer than time specified

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
Master Scheduler (SVC 22) (con't)	422		Job required too much queue space for initiation
	522		All tasks in SVC wait state for the length of time specified in the SMF parameter JWT
	622		Initiation of execution of task entered from TSO terminal terminated
	822		Not enough real storage available for problem program request for V=R region
WTO/ WTOR (SVC 23)	A23	Queue waiting for ORE (operator reply element)	ABEND of communications task occurred
	B23	WTOR	Request still unanswered when communications task ABEND occurred
	D23	WTO, WTRO	Parameter list does not begin on proper boundary; no buffers available; WTOR/MLWTO parameter list specified
	E23	Reply command processing (in response to WTOR)	Invalid ECB, RB, or reply address
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)	12A	ATTACH	Attempt to give shared subpool to new subtask
	22A	ATTACH	Subpool number greater than 127
	32A	ATTACH	Attempt to give job pack queue, which contains active programs, to new subtask
	42A	ATTACH	Address for ECB to be posted upon subtask termination is not multiple of 4; not within bounds of partition
	52A	ATTACH	Insufficient LSQA storage to copy necessary STAI information
	72A	ATTACH	Issuer specified invalid parameter address
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 multiple of 4, higher than hightest virtual storage, or does not have same protect key as CHAP issuer

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
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
DEQ (SVC 30)	130	DEQ without RET=HAVE	DEQ for resource not enqueued by prior ENQ
	230	DEQ	Invalid length specified for name of resource
	330	DEQ	Invalid option specified by task with non-zero protection key
	430	DEQ	Invalid parameter list
	530	DEQ	Task does not yet control specified resource
EOV (SVC 37)	137	End of volume	I/O error in label processing or tape positioning
	237	End of volume	Verification error in label processing; tape label block count not same as DCB count
EOV (SVC 37)	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
	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	Direct Access I/O error; DSCB not found for multi-volume or concatenated data set
	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
	B37	End of volume	Volumes must be demounted from a device allocated to the data set, but system unable to demount volume; DOS VTOC could not be converted to OS format
	D37	Output operation	More space needed but no secondary quantity specified for SPACE in DD

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
EOV (SVC 37) (con't)	E37	Output operation	More space needed but no more volumes specified in SER, volume count, or REF in DD; end-of-volume has found DSCB with a duplicate data set name on next volume
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	Invalid option specified by task with non-zero protection key
	438	ENQ	Invalid parameter list
	538	ENQ	Task requested one-at-a-time access to a permanently unavailable resource
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
	33E	DETACH	Subtask being detached not yet terminated (STAE=YES was specified)
CHKPT	13F		Error during execution of checkpoint restart
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
SWAP (SVC 55)	155		SVC 85 (in decimal) issued by user's task but is restricted for use by Dynamic Device Reconfiguration
System Restart	2F3		Job was being executed when system failure occurred; a system restart was performed
Supervisor Call (SVC nn)	Fnn		Invalid operand, nn, in SVC instruction
ESR (SVCs 109, 116, 117)	16D		Invalid ESR code in register 15
Paging	028		Paging supervisor detected a system error
	0D0		Invalid segment translation interrupt
	0D1		Invalid page-fault interrupt
	0F4		Page translation exception while supervisor lock set

OS/VS2 Completion Code Summary (con't)

Group	Completion Code	Operation or Macro Instruction	Explanation
Misc	047		Unauthorized program requested a restricted SVC
	14F	STATUS	Program issued STATUS macro instruction for other than STOP/START function
	160		Attention exit routine invalid
	16E	DEBCHK	Function performed on a DEB obtained from the DCB passed by the program could not be completed
	2FF	ABEND appendage	Step terminated at request of user appendage III
	3FE		ABEND occurred while teleprocessing I/O are active or pending
	4FE		ABEND occurred while non-teleprocessing I/O are active or pending

PLEASE COMMENT

Use Reader's Comment Form

Source Publications

Additonal information about the supervisor macro outlines and SVCs contained here is in *Supervisor Services and Macros*, GC27-6979.

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] [,GSPV=number / ,GSPL=address of list] [,SHSPV=number / ,SHSPL=address of list] [,SZERO={YES / NO}] [,TASKLIB=dcb address] [,STAI=(exit address[,parameter list address])] [,PURGE={NONE / HALT / QUIESCE}] [,ASYNCH={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 [,GSPV=number / ,GSPL=address of list] [,SHSPV=number / ,SHSPL=address of list] [,SZERO={YES / NO}] [,TASKLIB=dcb address] [,STAI=(exit address[,parameter list address])] [,PURGE={NONE / HALT / QUIESCE}] [,ASYNCH={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] {,MF=(E,{problem program list address / (1)}) / ,SF=(E,{control program list address / (15)}) / ,MF=(E,{address / (1)}),SF=E,{address / (15)})} [,GSPV=number / ,GSPL=address of list] [,SHSPV=number / ,SHSPL=address of list] [,SZERO={YES / NO}] [,TASKLIB=dcb address]

ATTACH (execute form) (cont'd)	[,STAI=(exit address[,parameter list address])] [,PURGE={NONE / HALT / QUIESCE}] [,ASYNCH={YES / NO}]
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 [,STAE={YES / NO}]*
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]

*VS2 only

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] L,LA=address,A=address[,SP=number]* R,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)	[L][,LA=address][,A=address][,SP=number]* [E][,LV=number][,A=address][,SP=number] [V][,A=address][,SP=number] ,MF=L
FREEMAIN (execute form)	[L][,LA=address][,A=address][,SP=number]* [E][,LV=number][,A=address][,SP=number] [V][,A=address][,SP=number] ,MF=(E,{control program list address / (1)})

* VS2 only

GETMAIN	EC,LV=number,A=address[,SP=number][,BNDRY={DBLWD|PAGE}] EU,LV=number,A=address[,SP=number][,BNDRY={DBLWD|PAGE}] LC,LA=address,A=address[,SP=number][,BNDRY={DBLWD|PAGE}]* LU,LA=address,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] [LC][,LA=address]* [LU][,LA=address]* [VC][,LA=address] [VU][,LA=address]}[,A=address][,SP=number] [,BNDRY={DBLWD|PAGE}],MF=L
GETMAIN (execute form)	{[EC][,LV=number] [EU][,LV=number] [LC][,LA=address]* [LU][,LA=address]* [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)	[DATA=address][,LNG=length][,FID=number] ,ID=value,MF=(E,{parameter list address|(1-12)})
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

*VS2 only

Supervisor Macros (con't)

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]
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 / (reg)}][,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=dcb address] [,ID=number] [,SDATA=(code)] [,PDATA=(code)] [,STORAGE=(address,address,...) ,LIST=address] ,MF=L
SNAP (execute form)	[DCB=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)]

*VS2 only

Supervisor Macros (con't)

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) })
STATUS (VS2 only)	{ START / STOP } [,TCB=subtask tcb address]
STIMER	{ REAL[,timer completion exit address] / TASK[,timer completion exit address] / WAIT } [,DINTVL=address ,BINTVL=address ,TUINTVL=address ,TOD=address ,MIC=address*]
TIME	[DEC BIN TU MIC,address]
TTIMER	[CANCEL] [, { TU* / MIC,address }]*
WAIT	[number of events,] { ECB=address / ECBLIST=address }
WAITR	[number of events,] { ECB=address / ECBLIST=address }
WTL	'message'
WTL (list form)	'message',MF=L

*VS2 only

Supervisor Macros (con't)

WTL (execute form)	MF=(E, {control program list address / (1)})
WTO (VS1 only)	'message'[,ROUTCDE=(number[,number...])] [,DESC=number]
WTO (list form) (VS1 only)	'message'[,ROUTCDE=(number[,number...])] [,DESC=number],MF=L
WTO (VS2 only)	{'message' / ('text'[,line type]),...} [,ROUTCDE=(number[,number],...)] [,DESC=(number[,number],...)][,AREAID=char]
WTO (list form) (VS2 only)	{'message' / ('test'[,line type]),...} [,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)})}

Valid Sup. Macro Parm Notation

Abbreviation		Meaning
SYM		Any symbol valid in the assembler language.
DEC 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 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	ASYNCH=	YES or NO						
	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
	GSPL=			SE			E	SL
	GSPV=	SLE	SLE	SE				
	LPMOD=	SLE	SLE	SE				
	PARM=			SE			E	S
	PURGE=	QUIESCE, HALT, or NONE						
	SHSPL=			SE			E	SL
	SHSPV=	SLE	SLE	SE				
	STAI=			SE			E	SL
	SZERO=	YES or NO						
	TASKLIB=			SE			E	SL
	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	

Summary of Supervisor Operands (con't)

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						
DETACH	tcb location address	S		S	S		S	
	STAE=	YES or NO						
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, or HAVE						
FREEMAIN	E, L, R or V	written as shown						
	A=(with E, L, or V)			SE			E	SL
	A=(with R)			S	S		S	
	LA=			SE			E	SL
	LV=(with E)	SLE	SLE	SE				
	LV=(with R)	S	S	S		S		
	SP=(with E, L, or V)	SLE	SLE	SE				
	SP=(with R)	S	S	S		S		
GETMAIN	EC, EU, LC, LU, VC, or VU	refer to macro description						
	A=			SE			E	SL
	BNDRY=	DBLWD or PAGE						
	LA=			SE			E	SL

Summary of Supervisor Operands (con't)

MACRO INSTRUCTION	OPERANDS	WRITTEN AS						
				Register				
		Sym	Dec Dig	(2-12)	(1)	(0)	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, L, 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	
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 Operands (con't)

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					
STATUS	STOP or START	written as shown						
	TCB=			S			S	
STIMER	REAL, TASK or WAIT	written as shown						
	timer completion exit addr			S		S	S	
	BINTVL=			S	S		S	
	DINTVL=			S	S		S	
	MIC=			S	S		S	
	TOD=			S	S		S	
	TUINTVL=			S	S		S	
TIME	DEC or BIN or TU	written as shown						
	MIC	written as shown						
	address			S		S	S	
TTIMER	CANCEL	written as shown						
	TU or MIC	written as shown						

Summary of Supervisor Operands (con't)

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						
	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
	HIARCHY=		SLE					
	PARAM=			E			E	

Programming Conversion 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 interruptions	No	Yes	Yes	Yes
Entry point	Must be the first byte of the routine or load module, and must be on a double-word boundary			
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 pass control to what other types of SVC routines	None	Any	Any	Any
Type of linkage with other SVC routines	Not applicable	Issue supervisor call (SVC) instruction		
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 Summary for OS/VS1

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 of 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 Summary for OS/VS1 (con't)

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)	III	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
43(2B)	II	DIRB		

SVC Summary for OS/VS1 (con't)

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	EOB 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		QPOST
68(44)	IV	SYNADAF	Same as reg 0 on entry to SYNAD	Same as reg 1 on entry to SYNAD
68(44)	IV	SNYADRLS		
69(45)	III	BSP		DCB address
70(46)	II	GSERV		Parameter list address

SVC Summary for OS/VS1 (con't)

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)		Reserved		
78(4E)	IV	DSCAN		
80(50)		Reserved		
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	XOMNGR	Address of list of ECB/IOB pointers (optional)	QMPA address
91(5B)	IV	VOLSTAT	DCB address	Zero: issued by CLOSE Non-zero: issued by EOV
92(5C) - 101(65)		Reserved		
102(66)	I	AQCTL		Parameter list address
103(67)	III	XLATE		
104(68)	IV	TOPCTL		
105(69)	III	IMAGLIB		
106(6A)		Reserved		
107(6B)	I	MODESET		Parameter list address
108(6C)		Reserved		

SVC Summary for OS/VS1 (con't)

Dec (hex) No.	Type	Macro	Register 0	Register 1
109(6D)	II	Extended SVC Router (ESR)	Parameters to ESR	Parameters to ESR
110(6E)	Reserved			
111(6F)	II	JECS		
112(70)	I	PGRLSE	Low address	High address
113(71)	I	SIR	ECB address or pointer to parameter list or contents ignored	Parameter word or PA
114(72)	I	EXCPVR		
115(73)	Reserved			
116(74)	I	Extended SVC Router (ESR)	Parameters to ESR	Parameters to ESR
117(75)	IV	Extended SVC Router (ESR)	Parameters to ESR	Parameters to ESR
118(76)	I	AT		

SVC Summary for OS/VS2

Dec (hex) No.	Type	Macro	Register 0	Register 1
0(0)	I	EXCP		IOB address
0(0)	I	XDAP		
1(1)	II	WAIT	Event count	ECB address
1(1)	II	WAITR	Event count	2's complement of ECB address
2(2)	II	PRTOV		
2(2)	II	POST	Completion code	ECB address
3(3)	I	EXIT		
4(4)	II	GETMAIN		Parameter list address
5(5)	II	FREEMAIN		Parameter list address
6(6)	II	LINK		(Register 15 has Parameter list address)
7(7)	II	XCTL		(Register 15 has Parameter list address)
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 (R Operand)	Subpool number (byte 0) Length (bytes 1-3)	Address of area to be freed
10(A)	I	FREEPOOL		
11(B)	II	TIME		Time units code
12(C)	II	SYNCH	(Register 15 contents branch address)	
13(D)	IV	ABEND		Completion code
14(E)	II	SPIE		PICA address
15(F)	I	ERREXCP		Request queue address element
16(10)	III	PURGE		Parameter list address
17(11)	III	RESTORE		IOB chain address
18(12)	II	BLDL	Address of build list	DCB address
18(12)	II	FIND	Parameter list address	DCB address
19(13)	IV	OPEN		Parameter list address of DCB addresses
20(14)	IV	CLOSE		Parameter list address of DCB addresses
21(15)	III	STOW	Parameter list address	DCB address

SVC Summary for OS/VS2 (con't)

Dec (hex) No.	Type	Macro	Register 0	Register 1
22(16)	IV	OPEN TYPE = J		Parameter list address of DCB addresses
23(17)	IV	CLOSE TYPE=J		Parameter list address of DCB addresses
24(18)	III	DEVTYPE	Output area address	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)	III	OBTAIN		Parameter list address
28(1C)	IV	CVOL		DCB or UCB address
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		UCB list address
33(21)	III	IOHALT		UCB address
34(22)	IV	MGCR	Indicator	Indicator
34(22)	IV	QEDIT		
35(23)	IV	WTO	Console source ID	Message address
35(23)	IV	WTOR	Console source ID	Message address
36(24)	IV	WTL		Message address
37(25)	II	SEGLD	Entry code	Parameter list address
37(25)	II	SEGWT	Entry code	Parameter list address
38(26)		Reserved		
39(27)	III	LABEL		Parameter list address
40(28)	II	EXTRACT		Parameter list address
41(29)	III	IDENTIFY	Entry point name address	Entry point address
42(2A)	II	ATTACH	(Register 15 has supervisor parameter list address)	Problem program parameter list address
43(2B)	II	CIRB	Entry point address	Size of work area in doublewords
44(2C)	II	CHAP	+ Increase priority - Decrease priority	Address of TCB address

SVC Summary for OS/VS2 (con't)

Dec (hex) No.	Type	Macro	Register 0	Register 1
45(2D)	II	OVLYBRCH		(Register 15 has entry point address)
46(2E)	I	TTIMER		1: Cancel time interval
47(2F)	II	STIMER	Exit address	Timer interval address
48(30)	II	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		DCB address
55(37)	IV	EOV	IOB address	DCB address
56(38)	II	ENQ		Parameter list address
56(38)	II	RESERVE		Parameter list address
57(39)	III	FREEDBUF	DECB address	DCB address
58(3A)	I	RELBUF		DCB address
58(3A)	I	REQBUF	Request count	DCB address
59(3B)	III	OLTEP	Parameter list address	Code
60(3C)	III	STAE	0 Create SCB 4 Cancel SCB 8 Overlay	Parameter list address
61(3D)		Reserved		
62(3E)	II	DETACH		TCB address
63(3F)	IV	CHKPT		Parameter list address
64(40)	III	RDJFCB		Parameter list address of DCB addresses
65(41)	II	QWAIT	(Register 2 has QUEUE address)	
66(42)	IV	BTAMTEST		IOB address
67(43)	II	ENDREADY		QPOST
68(44)	IV	SYNADAF	DECB or ICB address	DCB address
68(44)	IV	SYNADRLS		
69(45)	III	BSP		DCB address
70(46)	II	GSERV		Parameter list address

SVC Summary for OS/VS2 (con't)

Dec (hex) No.	Type	Macro	Register 0	Register 1
71(47)	III	ASGNBFR		Parameter list address
71(47)	III	BUFINQ		Parameter list address
71(47)	III	RLSEBFR		Parameter list address
72(48)	IV	CHATR		Parameter list address
73(49)	III	SPAR		Parameter list address
74(4A)	III	DAR		Parameter list address
75(4B)	III	DQUEUE	IQE address	Parameter list address
76(4C)	III	IFBSTAT	RDE code	Record address
77(4D)	IV	QTAMTEST	Request address	Parameter list address
78(4E)	III	WSCAN	UCB address	
79(4F)	I	STATUS	Start/Stop code	TCB address
80(50)	IV	IKASVC	Parameter list address	Console address
81(51)	IV	SETPRT		Parameter list address
82(52)	IV	DASDR		Parameter list address
83(53)	IV	SMFWTM		Message address
84(54)	I	GRAPHICS	UCB address and buffer restart address	
85(55)	IV	DDRSWAP		
86(56)	IV	ATLAS		Parameter list address
87(57)	III	DOM	If zero If negative	A DOM message I.D. Address of list of DOM message IDs
88(58)	III	MOD88		DCB address
89(59)	III	EMSRV		Parameter list address
90(5A)	IV	XQMNGR		QMPA address
91(5B)	III	VOLSTAT	DCB address	Issued code
92(5C)	I	TCBEXCP	TCB address	IOB address
93(5D)	III	TGET TPUT	TJID & buffer	Address of User's Buffer
94(5E)	III	STCC	Entry code	Indicator
95(5F)	I	TSEVENT	TJID/Entry Code or 0	
96(60)	III	STAX		Parameter List Address
97(61)	III	TEST-TSO	TCB,PRB,IRB address	Values

SVC Summary for OS/VS2 (con't)

Dec (hex) No.	Type	Macro	Register 0	Register 1
98(62)	IV	PROTECT		Parameter List Address
99(63)	IV	DDDYNAM		Parameter List Address
100(64)	III	SUBMIT		
101(65)	I	QTIP	Entry code	Save area Address
102(66)	I	AQCTL		Parameter List Address
103(67)		XLATE	Field length	Action byte and field address
104(68)	IV	TOPCTL	Subroutine indicator	Address of operator control work area
105(69)	III	IMAGLIB		Action indication
107(6B)	I	MODESET		Parameter list
109(6D)	IV	ESR	(Register 15 contains a routing code to appropriate SVC routine)	
110(6E)		DSTATUS	(Register 15 contains an indication)	Address of CSCB
112(70)	I	RELEASE	Starting address of virtual area	End address of virtual area +1
113(71)	I	SIR	+ Address of ECB - Address of ECB indirect area	High order bit on - address of virtual subarea list High order bit off - first word of virtual subarea list
114(72)	I	EXCPVR	Function code	Address of IOB
115(73)	I	BLKPAGE		Address of SPCT
116(74)	I	ESR	Address of HASP vector table	Address of UCB or password
117(75)	II	ESR	Function code	Address of parameter list of DCB, or DEB
119(77)	I	TESTAUTH	+ Authorization code - No code	Function code

Explanation of Style

Explanation of style	Footnotes:
Words in all capitals are coded as shown; appropriate values are to be substituted for words in lower case letters. Shaded operands may only be used in a VS2 system. 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

Load Module Control

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	[(register (s) 2-12)], { 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

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 [,0 to 16,777,215]
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]

TIME AND TIME INTERVALS FOR TIME, TTIMER, AND STIMER

Decimal (DEC and DINTVL operands):
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 or 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)

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.

Program Interruption Control

(see explanation of style - page 3-26)

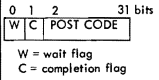
Enable and disable program interruptions and transfer control to interruption exit routine	SPIE	[interruption exit routine address] [,(interruption type[,interruption type]...)]

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			

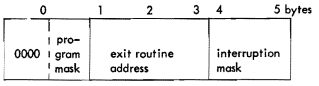
CONTROL BLOCKS

Event control block (ECB):

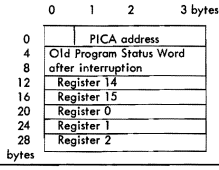


W = wait flag
C = completion flag

Program interruption control area (PICA):



Program interruption element (PIE):



(see explanation of style - page 3-26)

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' [,ROUTCDE=(number [,number] ,...)] {'message' {'text'[,line type]}, ...} [DESC=message descriptor code(s)]
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 AND TIME INTERVALS FOR TIME, TTIMER, AND STIMER

Decimal (DEC and DINTVL operands):
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 or 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 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 (con't)

(see explanation of style - page 3-26)

Save register contents — SAVE

(register(s) 14 through 12) [,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

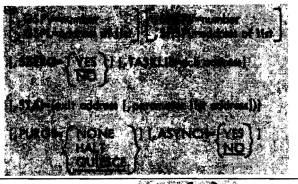
[(register(s) 14 through 12)] [,T] [{ ,RC=0 to 4095 / ,RC=(15) }]

In RETURN, T means: place all ones in high-order byte of save area word 4.

Terminate abnormally — ABEND

0 to 4095, [DUMP] [,STEP]

(see explanation of style - page 3-26)

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] 
Delete	DETACH	address of tcb address* [,STAE={YES / NO}]
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.
	STATUS	{START / STOP} [,TCB=subtask tcb address]

Virtual Storage Allocation

(see explanation of style - page 3-26)

Allocate storage	GETMAIN	R,LV=length° [,SP=0 to 127]
	GETMAIN	{EC / EU},LV=length° / {VC / VU},LA=address of length° list / [redacted] ,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]
	FREEMAIN	{E,LV=length° / V / [redacted]},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
L=virtual area(s), variable length(s)
U=unconditional
C=conditional

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

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.

Data Management Macros

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) (0)} , {buffer length (2-12)}}	
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)}]	

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 (2-12)],[number of buffers],[buffer length] ,[record area address (2-12)],[record area length] ,MF=(E,{control program list address (1)})	
BDAM BISAM BPAM BSAM	CHECK	{decb address (1-12)}[,DSORG={IS / 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
	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 (con't)

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)})	
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]}	Not available to user program

Data Management Macros (con't)

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 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])]	
QISAM	ESETL	{dcb address / (1-12)}	
BSAM QSAM	FEOV	{dcb address / (1-12)} [,{REWIND / LEAVE}]	

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
BPAM	FIND	{dcb address / (1-12)} , { {name address / (2-12) / (0)} , D / {relative address list / (2-12) / (0)} , C }	00=A 04=B 08=C Note: reladr, C always returns CC of 00
BDAM BISAM BPAM BSAM	FREEBUF	{dcb address / (1-12)} ,register* *Note: Reg, any of 2 to 12, contains addr of buffer.	
BDAM BISAM	FREEDBUF	{decb address / (2-12) / (0)} ,{K / D} ,{dcb address / (1-12)}	
BDAM BISAM BPAM BSAM QISAM QSAM	FREEPOOL	{dcb address / (1-12)}	
QISAM QSAM	GET	{dcb address / (1-12)} [,{area address / (2-12) / (0)}]	

Data Management Macros (con't)

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)} , {buffer length / (2-12)} / (0)}	
BPAM BSAM	NOTE	{dcb address / (1-12)}	
BDAM BISAM BPAM BSAM QISAM QSAM	OPEN	({dcb address / (2-12)} ,[(options)],...)	

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
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 / (1)})	

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 / OUTPUT / RDBACK]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / OUTPUT / UPDAT]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / OUTPUT]	—
BSAM	[INPUT / OUTPUT / INOUT / OUTIN / RDBACK]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / OUTPUT / INOUT / OUTIN / UPDAT]	[,REREAD / ,LEAVE / ,DISP]	[INPUT / OUTPUT]	—
QISAM (Load Mode)	—	—	[OUTPUT]	—	—	—
BPAM, BDAM	—	—	[INPUT / OUTPUT / UPDAT]	—	—	—

Optionally select one from vertical stack within [ ]

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
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)}]	

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
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}, {dcb address (2-12)}, {area address (2-12)}, {length (2-12) 'S'}	
	READ (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 parameters (e.g., K, SF, DI) from the applicable standard form of the READ macro.	

Data Management Macros (con't)

A/M	Macro	Parameters	Condition 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 macro.	
BDAM	RELEX	D, {dcb address (2-12)}, {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 (con't)

A/M	Macro	Parameters	Completion Codes
BSAM QSAM	SETPRT	{dcb address (2-12)} {, UCS=(csc [, F[OLD] / , F[OLD], V[ERIFY] / , , V[ERIFY]]) [, FCB=(image-id [, V[ERIFY]] [, A[LIGN]])] [, OPTCD={B / U}] / , FCB=(image-id [, V[ERIFY]] [, A[LIGN]])] [, OPTCD={B / U} [, F[OLD] / , U[NFOLD]]] / , OPTCD={B / U} [, F[OLD] / , U[NFOLD]])}	

SETPRT Completion Codes

FCB Bits 16-23	UCS Bits 24-31	
00	00	Successful completion
04	04	Operator cancellation
08	08	Permanent I/O error in image library
0C	0C	Permanent I/O error during load
10	10	Permanent I/O error during image display
14	14	Operator cancellation: incorrect image

FCB/UCS Bits 24-31	
18	NOP: incorrect specification
1C	NOP: error during previous I/O attempt
20	Insufficient space for IMAGELIB blocks
24	Unable to open SYS1.IMAGELIB

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
BSAM QSAM	SETPRT (list form)	[dcb address { ,UCS=(csc [,F [OLD] / ,F [OLD],V [ERIFY] / ,V [ERIFY]]) [,FCB=(image-id [,V [ERIFY] / ,A [LIGN]])] [,OPTCD={B / U}] ,FCB=(image-id [,V [ERIFY] / ,A [LIGN]])[,OPTCD=({B / U}[,F [OLD] / ,U [NFOLD]])] ,OPTCD={B / U}[,F [OLD] / ,U [NFOLD]]) }] ,MF=L	

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
BSAM QSAM	SETPRT (execute form)	[{dcb address / (2-12)} { ,UCS=(csc [,F [OLD] / ,F [OLD],V [ERIFY] / ,V [ERIFY]]) [,FCB=(image-id [,V [ERIFY] / ,A [LIGN]])] [,OPTCD={B / U}] ,FCB=(image-id [,V [ERIFY] / ,A [LIGN]])[,OPTCD=({B / U}[,F [OLD] / ,U [NFOLD]])] ,OPTCD={B / U}[,F [OLD] / ,U [NFOLD]]) }] ,MF=(E,{data management list address / (1-12)})	

Data Management Macros (con't)

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		--	--
10	Permanent I/O error in directory			
14	Specified data control block not open			
18	Conditional GETMAIN unsuccessful			

A/M	Macro	Parameters	Completion Codes
BDAM BISAM BPAM BSAM QISAM QSAM EXCP	SYNADAF	{ACSMETH=BDAM / ACSMETH=BPAM / ACSMETH=BSAM / ACSMETH=QSAM} [,PARM1=parm reg*][,PARM2=parm reg**] {ACSMETH=BISAM / 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)}	
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 (con't)

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.	

Data Management Macros (con't)

A/M	Macro	Parameters	Completion Codes
BDAM BISAM BPAM BSAM QISAM QSAM	XLATE	{area address (2-12)}, {length (2-12)} [, TO={A E}]	

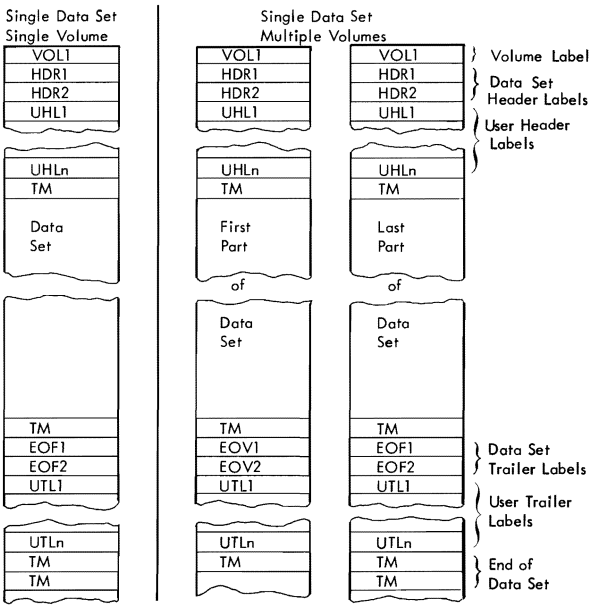
Direct Access Device Capacities

Device Type	Volume Type	Track Capacity	Tracks/ Cylinder	No. of Cylinders	Total Capacity
2314/ 2319	Disk	7294	20	200	29,176,000
3330	Disk	13030	19	404	101,751,270
2305-1	Drum	14136	8	48	5,428,224
2305-2	Drum	14660	8	96	11,258,880

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	191+KL+DL	191+KL+DL	135+DL	135+DL
2305-1	632+KL+DL	632+KL+DL	430+DL	430+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

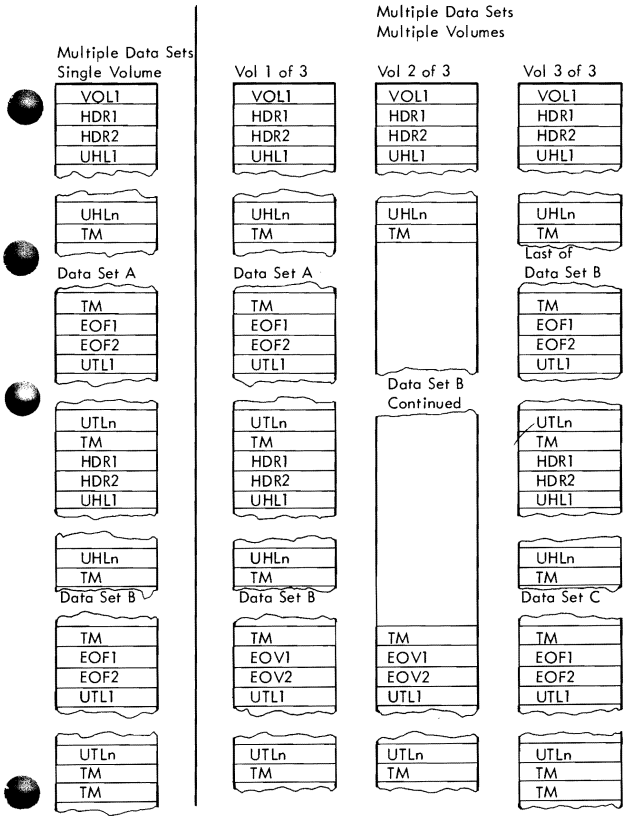
Volume Organization with IBM Standard Labels



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 (con't)



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.

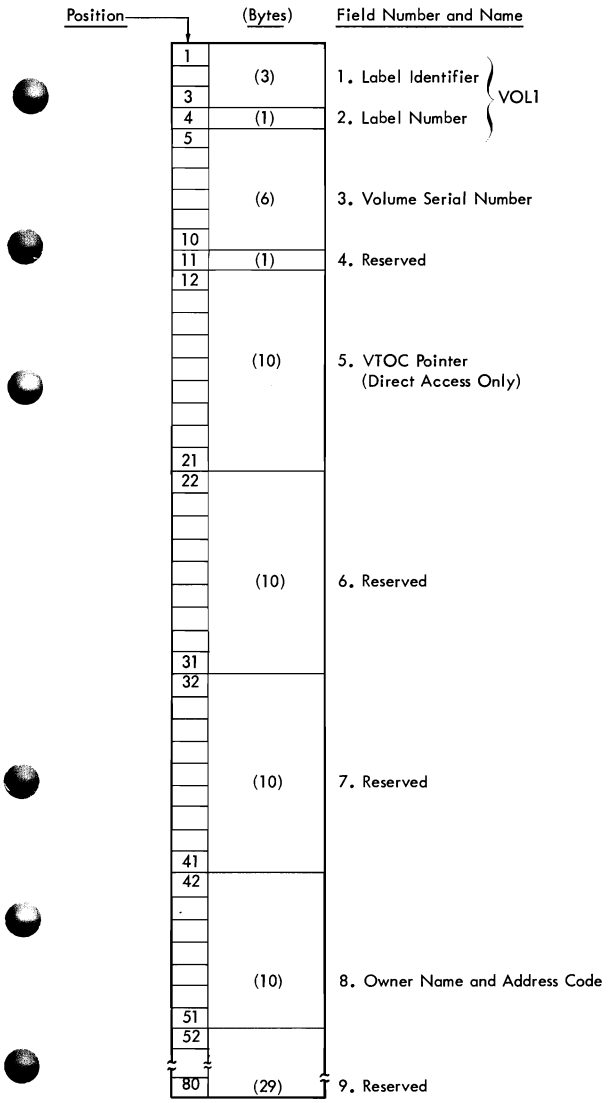
IBM Standard Label Processing by Data Management Routines

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:							
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:							
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

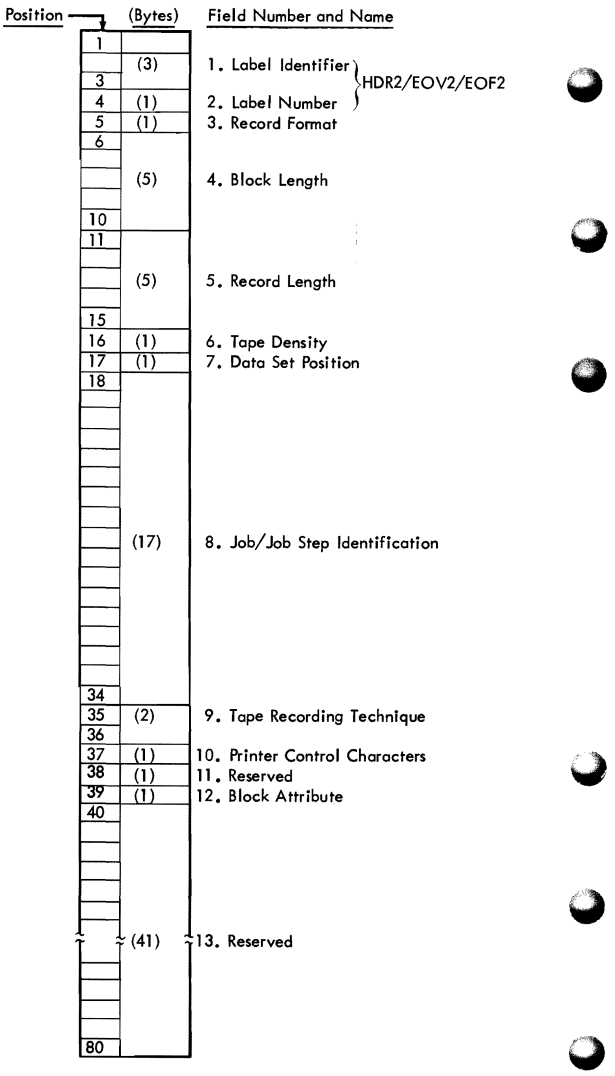
Notes:

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 exsiting labels).

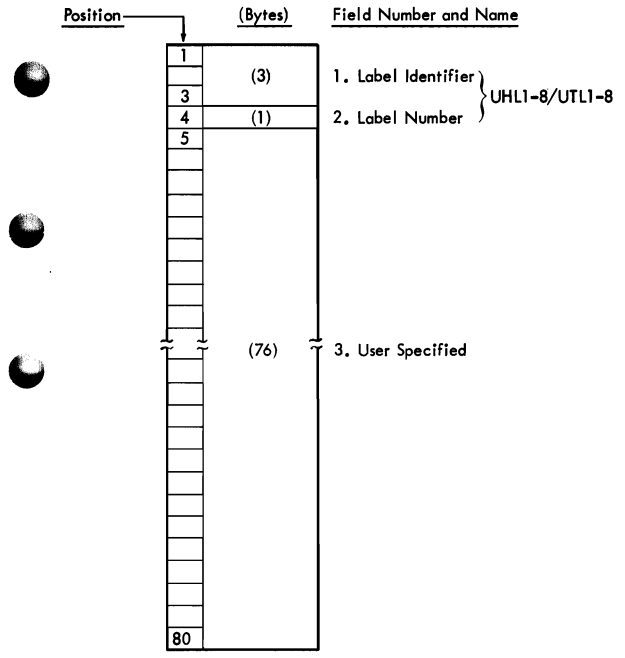
Format of IBM Standard Volume Label



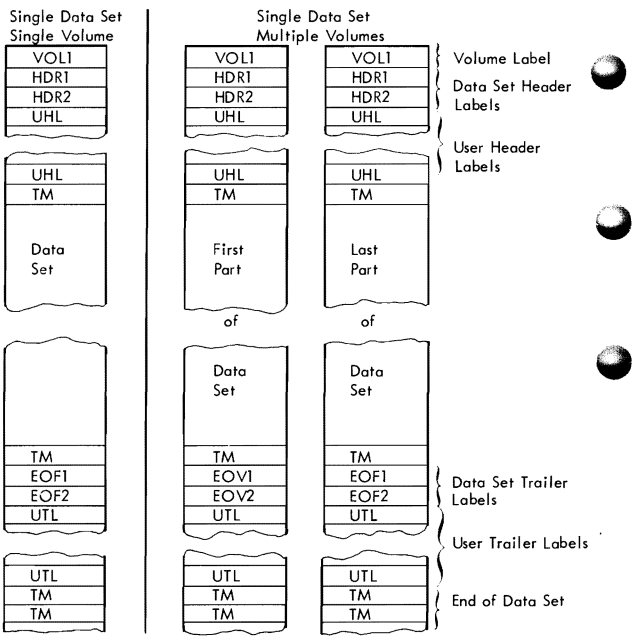
Format of IBM Standard Data Set Label 2



Format of User Label



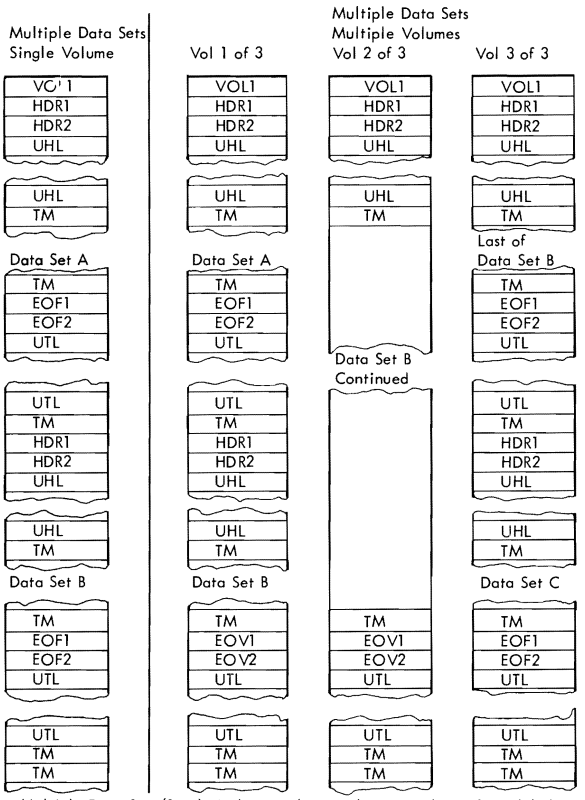
Volume Organization with ANSI Standard Labels



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 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 (con't)



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.

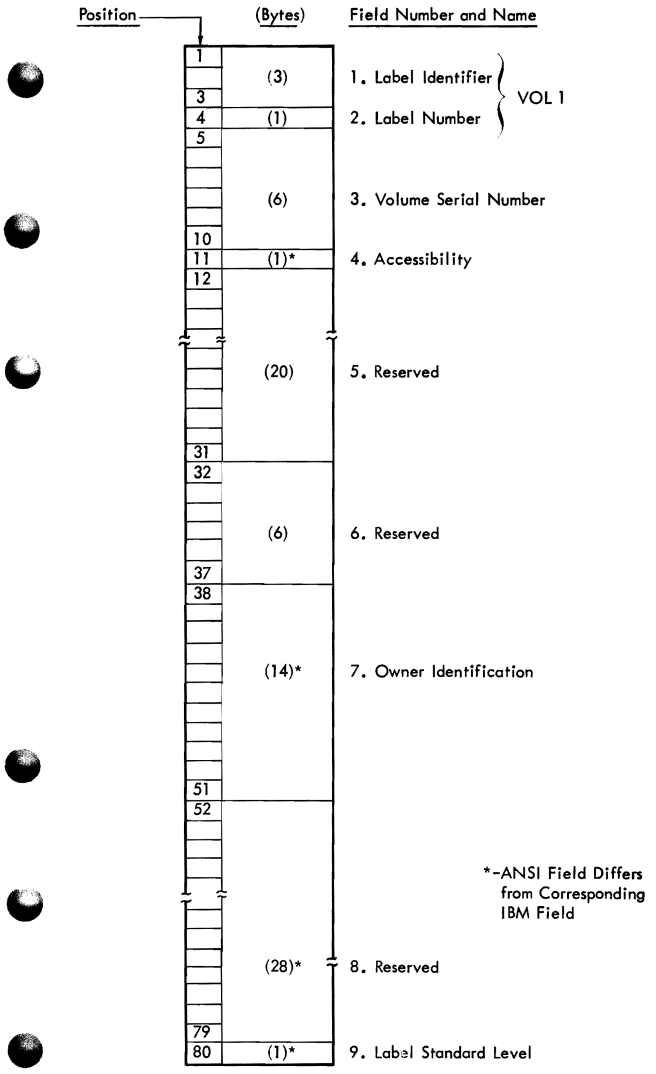
Multiple Data Sets/Multiple Volumes: More than one volume is needed to contain the multiple data set aggregate. The last volume is organized the same as a multiple data set/single volume layout. On the other volumes, the last data set trailer labels are identified as EOV instead of EOF, and the last 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

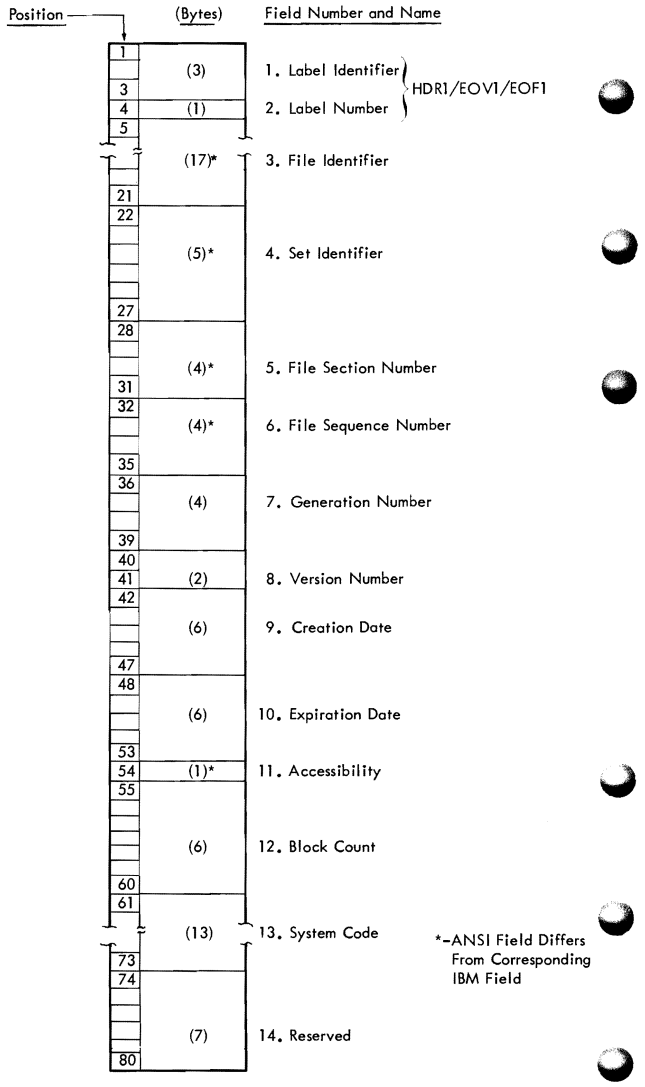
Processing	Volume Label		Header Labels[1]				Trailer Labels[1]			
	VOL1	USER VOLUME LABELS	HDR1	HDR2	HDR3-9	UHL	EOF1 or EOV1	EOF2 or EOV2	EOF3-9 or EOV3-9	UTL
First or Only Volume[2]:										
Checks labels on input tape.	Open	Ignored	Open	Open	Ignored	Open	EOV	bypassed	ignored	EOV
Checks existing lables 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 actian 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 label routine to process user trailer labels (before overwriting the existing labels).

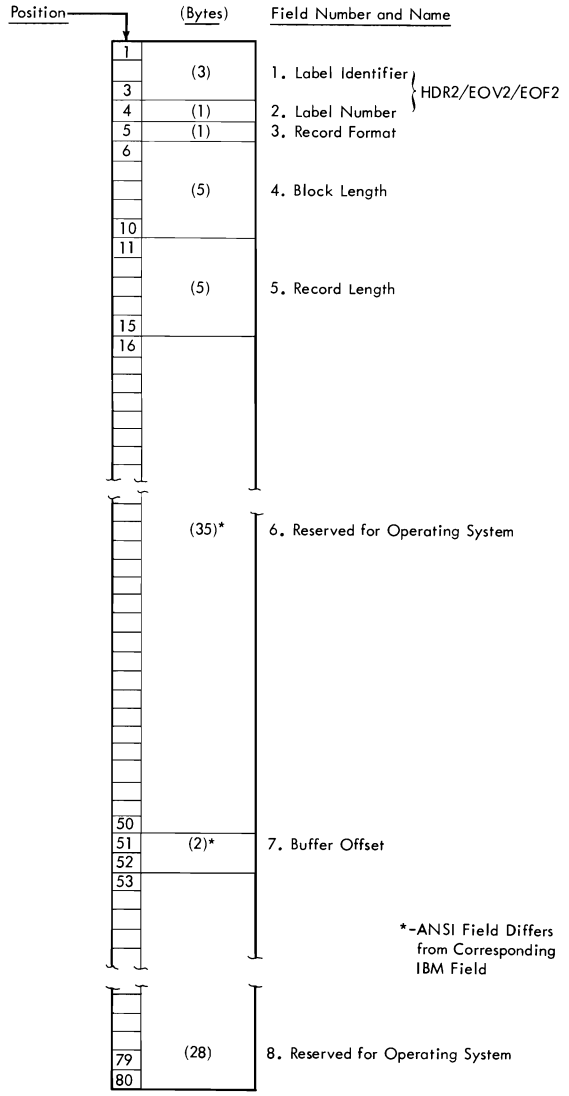
Format of ANSI Standard Volume Label



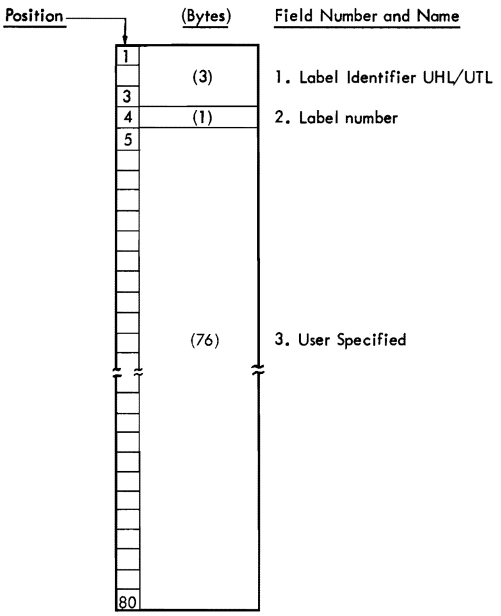
Format of ANSI Header 1 and Trailer 1 Labels



Format of ANSI Header 2 and Trailer 2 Labels



Format of ANSI User Labels



Component Support of Label Processing Features

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

Source Publications

Additional JCL reference information is contained in *OS/VS JCL Reference,* GC28-0618.

Additional operator commands information can be found in Operator's Library: *OS/VS1 Reference,* GC38-0110 or Operator's Library: *OS/VS2 Reference,* GC38-0210.

Additional SMF information is contained in *OS/VS System Management Facility (SMF),* GC35-0004.

Additional RES information is contained in Operator's Library: *OS/VS1 RES,* GC38-0330, and *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.

PLEASE COMMENT

Use Reader's Comment Form

Job Statement

The JOB Statement

//Name	Operation	Operands	P/K	Comments
//jobname	JOB	([account number][,additional accounting information,...])	P	Can be made mandatory
		[programmer's name]	P	Can be made mandatory
		[ADDRSPC= {VIRT / REAL}]	K	Requests storage type
		[CLASS=jobclass]	K	Assigns A-O
		[COND=((code,operator),...)]	K	Specifies a maximum of 8 tests
		[MSGCLASS=output class]	K	Assigns A-Z,0-9
		[MSGLEVEL=([0 / 1 / 2][,0 / ,1])]	K	
		[NOTIFY=user identification]2	K	For TSO only
		[PRTY=priority]	K	Assigns 0-13
		[RD= {R / RNC / NC / NR}]	K	Restart definition
		[REGION=valueK]	K	Specifies amount of storage space
		[RESTART=({* / stepname / stepname.procstepname} [,checkid])]	K	For deferred restart
		[TIME= {(minutes,seconds) / 1440}]	K	Assigns job CPU time limit
		[TYPERUN= {HOLD / SCAN1}]	K	Holds a job in job queue, or, for OS/VS1 only, scans JCL for syntax errors

Legend:
P Positional parameter.
K Keyword parameter.
{ } Choose one.
[ ] Optional; if more than one line is enclosed, choose one or none.
1 For OS/VS1 only.
2 For OS/VS2 only.

EXEC Statement

The EXEC Statement

//Name	Operation	Operands	P/K	Comments
//[stepname]	EXEC	{PGM={program name / *.stepname.ddname / *.stepname.procstepname.ddname} / {[PROC=]procedure name}}	P	Identifies program or cataloged procedure
		[ACCT=(accounting information,...) / ACCT.procstepname=(accounting information,...)]	K	Accounting information for step
		[ADDRSPC={VIRT / REAL}]	K	Requests storage type
		[COND=([(code,operator) / (code,operator,stepname) / (code,operator,stepname.procstepname)],...[,][EVEN / ONLY])	K	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])	K	Specifies a maximum of 8 tests, or 7 tests if EVEN or ONLY is coded
		[DPRTY=(value1,value2)]2	K	Specifies a dispatching priority for a job step
		[PARM=value / PARM.procstepname=value]	K	Parentheses or apostrophes enclosing value may be required
		[RD={R / RNC / NC / NR} / RD.procstepname={R / RNC / NC / NR}]	K	Restart definition
		[REGION=valueK]		Specifies amount of storage space
		[TIME={(minutes,seconds) / 1440} / TIME.procstepname={(minutes,seconds) / 1440}]	K	Assigns step CPU time limit

Legend:
P Positional parameter.
K Keyword parameter.
{ } Choose one.
[ ] Optional; if more than one line is enclosed, choose one or none.
1 For OS/VS1 only.
2 For OS/VS2 only.

DD Statement

The DD Statement

//Name	Operation	Operands	P/K	Comments
//[ddname / procstepname.ddname]	DD	[* / DATA[,DLM=xx]]	P	Defines data set in the input stream
		[DUMMY]	P	Bypasses I/O operations on a data set (BSAM and QSAM)
		[DYNAM]2	P	
		[AFF=ddname]	K	Requests channel separation
		[COPIES=nnn]	K	For use with the SYSOUT parameter
		[DCB=(list of attributes) / DCB=(ddname / *.ddname / *.stepname.ddname / *.stepname.procstepname.ddname [,list of attributes])]	K	Completes the data control block
		[DDNAME=ddname	K	Postpones the definition of a data set
		[DEST=userid]'	K	Specifies remote destination for output data set
		[DISP=([NEW / OLD / SHR / MOD] [,DELETE / ,KEEP / ,PASS / ,CATLG / ,UNCATLG] [,DELETE / ,KEEP / ,CATLG / ,UNCATLG])]	K	Assigns a status, disposition, and conditional disposition to the data set
		[DLM=delimiter]	K	Assigns delimiter other than */
		[{DSNAME / DSN} = {dsname / dsname(member name) / dsname(generation number) / dsname(area name) / &&dsname / &&dsname(member name) / &&dsname(area name) / *.ddname / *.stepname.ddname / *.stepname.procstepname.ddname}]	K	Assigns a name to a new data set or to identify on existing data set. An unqualified name is 1-8 characters, beginning with an alphabetic or national character.
		[FCB=(image-id [,ALIGN / ,VERIFY])]	K	Specifies forms control information. The FCB parameter is ignored if the data set is not written to a 3211 printer.
		[HOLD={YES / NO}]1	K	Specifies whether output processing is to be deferred or processed normally

DD Statement (con't)

The DD Statement (con't)

//Name	Operation	Operands	P/K	Comments
//[ddname / procstepname.ddname]	DD (con't)	[LABEL=([data set seq #][,SL / ,SUL / ,AL / ,AUL / ,NSL / ,NL / ,BLP / ,LTM / ,][,PASSWORD / ,NOPWREAD / ,][,IN / ,OUT][,][EXPDT=yyddd / RETPD=nnnn])	K	Supplies label information
		[OUTLIM=number][1]	K	Limits the number of logical records you want included in the output data set
		[QNAME=process name]	K	Specifies the name of a TPROCESS macro which defines a destination queue for messages received by means of TCAM
		[SEP=(ddname,...)][1]	K	Requests channel separation
		[[1]SPACE=(TRK / CYL / blocklength,(primary[,secondary][,directory][,index])[,RLSE][,CONTIG / ,MXIG / ,ALX][,ROUND])	K	Assigns space on a direct access volume for a new data set
		[[2]SPACE=(ABSTR,(primary quantity,address[,directory][,index]))	K	Assigns specific tracks on a direct access volume for a new data set
		[SPLIT=(n,CYL,(primary quantity[,secondary quantity])) / (percent,blocklength,(primary quantity[,secondary quantity])) / percent]	K	Assigns space on a direct access volume for a new data set. Data sets share cylinders
		[SUBALLOC=(TRK / CYL / blocklength,(primary[,secondary][,directory])[,ddname / ,stepname.ddname / ,stepname.procstepname.ddname])]	K	Requests part of the space on a direct access volume assigned earlier in the job
		[SYSOUT=(classname[,program name][,form number]),[OUTLIM=number]]	K	Routes a data set through the output stream. For classname, assign A-Z or 0-9
		[TERM=RT][1]	K	Indicates that on RTAM device is in use

DD Statement (con't)

The DD Statement (con't)				
//Name	Operation	Operands	P/K	Comments
(continued from previous page)		[TERM=TS] 2	K	Identifies TSO user
		[UCS=(character set code[,FOLD] [,VERIFY])]	K	Requests a special character set for a 1403 printer
		[{UNIT=({unit address / device type / group name}[,unit count / ,P /,] [,DEFER][,SEP=(ddname,...)]) / UNIT=AFT=ddname}]	K	Provides the system with unit information
		[{VOLUME / VOL}=([PRIVATE][,RETAIN][,volume seq#] [,volume count] [,{SER=(serial number,...) / REF=dsname / REF=*.ddname / REF=*.stepname.ddname / REF=*.stepname.procstepname.ddname}])]	K	Provides the system with volume information

Legend:
P Positional parameter.
K Keyboard parameter.
{ } 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 For OS/VS1 only.
2 For OS/VS2 only.

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.

Operation	Operand
{CANCEL / C}	{ jobname [DUMP[,ALL] / ,IN=[i / HOLD] / ,OUT=[s / HOLD]] / unitaddr / devicetype / [procname.] identifier }
{DEFINE / N}	[LIST / PARM=membername]
{DISPLAY / D}	{ T / A / U [,TP / ,GRAPHIC / ,TAPE / ,DASD / ,UR] [,ONLINE / ,OFFLINE] [,cuu] [,nnn] / R / Q[=qclass] / N[=qclass] / jobname[,HOLD] / CONSOLES / SQA }
DUMP	[text]
{HALT / Z}	EOD
{HOLD / H}	{ jobname / Q[=inclass] }
{LOG / L}	'text'
MODE	{ STATUS / RETRY{RECORD / QUIET} / MAIN{RECORD / QUIET} / CONTROL{THRESHOLD / QUIET} }
{MODIFY / F}	{ [procname.]{identifier / Pnn} / unitaddr } {[,CLASS=outclass] / [,CLASS=jobclass]} [,'text'] [,PAUSE={FORMS / DATASET}] {[,JOBCLASS=jobclass][,OUTCLASS=s] / [,OUTCLASS=s]}
{MONITOR / MN}	{ JOBNAMES[,T] / DSNAME / SPACE / STATUS / A }
{MOUNT / M}	unitaddr,VOL={(NL,volserial) / (SL,volserial) / (AL,volserial)} [,USE={STORAGE / PUBLIC / PRIVATE}]

VS1 Operator Command Outlines (con't)

Operation	Operand
{MSGRT / MR}	{(D=(display-operand,...)[,MN=A][,K]) / REF}
{RELEASE / A}	{jobname / Q[=inclass]}
{REPLY / R}	id,{'text' / text}
{RESET / E}	jobname,{PRTY=nn / CLASS=c}[,OUT=s]
{SET / T}	DATE=yy.ddd[CLOCK=hh.mm.ss]
{START / S}	procname[.{identifier / Pnn / ALL}][,[{unitaddr / devicetype}][,volserial], [jobname outclass jobclass (jobclass,RESV=nn) (RESV=nn) (JOBCLASS=class,OUTCLASS=s) (parm) ([MODE={INT / (INT,S) / EXT}][,TIME=YES[,DEBUG=YES][,BUF=nnn])] [,keyword=option,...]* * The keyword=option parameter(s) can follow after the last positional parameter.
{STOP / P}	{[procname][,identifier / ,Pnn] unitaddr jobname JOBNAMES DSNAME SPACE STATUS}
{STOPMN / PM}	{JOBNAMES / DSNAME / SPACE / STATUS}
{SWAP / G}	{OFF / ON / unitaddr,cuu}
{SWITCH / I}	SMF
{UNLOAD / U}	unitaddr
{VARY / V}	{unitaddr / (unitaddr,unitaddr...)}{,ONLINE / ,OFFLINE / ,PATH,cuu,{ONLINE / OFFLINE}}

VS1 Operator Command Outlines (con't)

Operation	Operand
{VARY / V}	{unitaddr / (I-cuu,O-cuu)},MSTCONS
{VARY / V}	{unitaddr / SYSLOG},HARDCPY[{,CMDS / ,NOCMDS / ,OFF / ,INCMDS / ,STCMDS}] [,ROUT={ALL / NONE / (routecode[,routecode]...)}]
{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={nnn / DS} / BSP={nnn / DS / JOB} / LSP={n / C} / HOLD / REPEAT={(nnn,JOB) / nnn}}[,JBN=jobname]

Operation	Operand
{CANCEL / C}	{ identifier / devicetype / unitaddr / devicename / jobname [[, DUMP] [, ALL] / , IN[=i] / , OUT[=s]] }
{CONTROL / K}	C, D, idd[, L=cc]
{DISPLAY / D}	{ SQA / A / T / U [, TP / , GRAPHIC / , TAPE / , DASD / , UR] [, OFFLINE / , ONLINE] [, cuu] [, nnn] / CONSOLES / jobname / R / Q[=qclass] / N[=qclass] / C, K }
DUMP	COMM=(comment)
{HALT / Z}	EOD
{HOLD / H}	{ Q[=inclass] / jobname }
{LOG / L}	'text'
MODE	{ STATUS / RETRY[,] {RECORD / QUIET} / MAIN[,] {RECORD / QUIET} / CONTROL[,] {THRESHOLD / QUIET} }
{MODIFY / F}	{ jobname, parm / [procname.]identifier { , CLASS=jobclass / , CLASS=outclass / , PAUSE= {FORMS / DATASET} } }
{MONITOR / MN}	{ JOBNAMES[, T] / DSNAME / SPACE / STATUS }

VS2 Operator Command Outlines (con't)

Operation	Operand
{MOUNT / M}	{unitaddr / devicename} {,VOL=(SL,serial) / ,VOL=(AL,serial) / ,VOL=(NL,serial)} [,USE={STORAGE / PUBLIC / PRIVATE}]
{MSGRT / MR}	{D=(display-operand,...) / REF} [,L={a / cc / cca}]
{RELEASE / A}	{Q[=inclass] / jobname}
{REPLY / R}	id,['text']
{REPLY / R} (used for DUMP)	{U / STOR=(startaddr,endaddr,...)[,SDATA] / SDATA}
{RESET / E}	jobname {,PRTY=nn / ,CLASS=c / ,PRTY=nn,CLASS=c} [,OUT=s]
{SET / T}	DATE=yy.ddd[,CLOCK=hh.mm.ss]
{START / S}	procname[.identifier] [,cuu] [,volumeserial] [,parmvalue] [,jobname] [,LSQA=nn] [,keyword=option,...] {GTF / GTFSNP} [.identifier] [,[cuu],[volumeserial]] [,([MODE={INT / (INT,S)}] [,BUF=nnn] [,TIME={YES / NO}] [,DEBUG={YES / NO}])] [,REG=nnn]
{STOP / P}	{[procname.]identifier / jobname}
{STOPMN / PM}	{JOBNAMES / DSNAME / SPACE / STATUS}
{SWAP / G}	{OFF / ON / unitaddr,cuu}
{SWITCH / I}	SMF
{UNLOAD / U}	unitaddr
{VARY / V}	({unitaddr / O-cuu / (I-cuu,O-cuu)} [{,unitaddr / ,O-cuu / ,(I-cuu,O-cuu)}]...) ,CONSOLE [,AUTH={ALL / INFO / ([SYS][,IO][,CONS])}] [,ROUT={ALL / NONE / (routecode[,routecode]...)}] [,ALTCONS={unitaddr / O-cuu / (I-cuu,O-cuu)}]

VS2 Operator Command Outlines (con't)

Operation	Operand
{VARY / V}	{unitaddr / (unitaddr,unitaddr...)} [,ONLINE / ,OFFLINE / ,PATH,cuu, {ONLINE / OFFLINE}]
{VARY / V}	{unitaddr / (1-cuu,0-cuu)} ,MSTCONS
{VARY / V}	{unitaddr / SYSLOG} ,HARDCPY [,CMDS / ,NOCMDS / ,OFF / ,INCMDS / ,STCMDS]
	[,ROUT= {ALL / NONE / (routecode[,routecode]...)}]
{WRITELOG / W}	{s / CLOSE}

Definitions of Substitutional Operands

a	an area on a graphics console.
c	one input (A-O) or output (A-Z,0-9) class.
cc	console identification number.
class	one to fifteen job classes (A-O) without priorities.
comment	a 1-100 character identifier.
cuu	the channel and unit address (cuu) on an I/O device.
devicename	a device that was specified, such as 231401 or 231400.
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.
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.
idd	a three digit identification number of the status display.
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.
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-O). 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.
n	a single digit decimal number.
nn	a two digit number from 00 to 03.
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.
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-O 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.
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 (60-99) and Julian day (000-366).

SMF

SMFDEFLT 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=nnnnn]	nnnnn - buffer size in bytes (max for VS1 is 2048, for VS2 is 8192)
SID=xx	xx - system type
MDL=nn	nn - user 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
[PRM={vol ser no | (,dev addr)}]	vol - volume serial number of SMF volume dev - device address of SMF volume (cuu)
[ALT={vol ser no | (,dev addr)}]	vol - volume serial number of alternate volume dev - device address of alternate volume

RES Central Operator Commands

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

Operator commands that use additional operands for RES.

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

*Command not valid from workstation.

New operator commands for RES.

LISTBC	ROUTE
LOGON	SEND
LOGOFF	

RES Central Operator Command Outline

Operation	Operand
{CANCEL / C}	{[JBN=]jobname [,DUMP] [,ALL] [,USER=userid] / ,IN[=inclass|HOLD] / ,OUT[=outclass|HOLD]} / {[DEV=]id / unitaddr / devicetype} [,USER=userid]
{DISPLAY / D}	{jobname[,HOLD] / Q[=qlist] / N[=qlist]} [,ALLQ / ,USER=userid] / USER [,L / =userid] / R [,USER=userid / ,ALL] / RT, {ALL / ACT / INACT} [,L] / TERM{=term-id / =nnn.aam[,nnn.aam,...]}
{HOLD / H}	{jobname[,USER=userid] / Q[=list]}
{LISTBC / LB}	[NOTICES] [,MAIL=userid]
LOGON	CENTRAL [/password] [PROC(procname)]
LOGOFF	{CENTRAL / userid}
{MODIFY / F}	[procname.], {START / STOP / RESTART} = {ALL / (n,...)}
{MONITOR / MN}	SESS[,T]
{RELEASE / A}	{jobname[,USER=userid] / Q[=list]}
{REPLY / R}	[R]msgno,'text'
{RESET / E}	jobname[,PRTY=pp] [,CLASS=class] [,OUT=outclass] / ,USER=userid
{ROUTE / RO}	{JBN=jobname[,USER=userid] / ALL,USER=userid / {JBN=jobname / ALL},GROUP=list[,USER=userid]} [,CLASS=outclass] / [,DEST=userid] [,HOLD={YES / NO}]

RES Central Operator Command Outline (con't)

Operation	Operand
{SEND / SE}	{ 'text' {SAVE / LOGON} messageno [,LIST / ,DELETE] 'text' [,ALL / ,USER=(userid, ...) / ,OPERATOR=route-code] [,NOW / ,LOGON / ,SAVE] }
{START / S}	{ procname[.id] {wtr / rdr} [.id], USER=userid }
{STOP / P}	[procname.]id {blank / ,USER=userid}
STOPMN	SESS
{WRITER / WTR}	unit, { FSP={nfs / DS} BSP={nbs / DS / JOB} LSP={n / C} HOLD REPEAT={nnn / (nnn, JOB)} } [,JBN=jobname] [,USER=userid]

RES Workstation Operator Command Outline

Operation	Operand
{CANCEL / C}	{ [JBN=] jobname { [,DUMP] [,ALL] / ,IN[={inclass|HOLD}] / ,OUT[={outclass|HOLD}] } / [DEV=] id }
{DISPLAY / D}	{ T / R / jobname[,HOLD] / Q[=qlist] / N[=qlist] / USER [,L / =userid] }
{HOLD / H}	jobname
{LISTBC / LB}	[NOTICES[,MAIL] / MAIL[,NOTICES]]
{LOG / L}	'text'
LOGOFF	
LOGON	userid[/password] TERM(term-id) [PROC(procname)] [NOTICES / NONOTICES] [MAIL / NOMAIL]
{MODIFY / F}	{ [procname.]id[,CLASS=list] [,PAUSE={FORMS / DATASET}] / procname.Pnn,'data' }
{MONITOR / MN}	JOBNAMES[,T]
{RELEASE / A}	jobname
{REPLY / R}	msgno { BLANK / ,'text' / ,text }
{RESET / E}	jobname { ,PRTY=pp[,CLASS=outclass] / ,CLASS=outclass[,PRTY=pp] } [,OUT=outclass]
{ROUTE / RO}	{ JBN=jobname[,GROUP=list] / ALL[,GROUP=list] / GROUP=list } [,CLASS=outclass] [,DEST=userid] [,HOLD=(YES|No)]
{SEND / SE}	'text' [,USER=(userid[,userid]...) [,{NOW / LOGON / SAVE}] / ,OPERATOR=route-code]

RES Workstation Operator Command Outline (con't)

Operation	Operand
{STOP} {P}	[procname.]id
STOPMN	JOBNAMES
{WRITER} {WTR}	unit { ,HOLD / ,FSP={DS | nfs} / ,BSP={DS | JOB | nbs} / ,REPEAT={nnn | (nnn,JOB)} / ,LSP={n | C} } [,JBN=jobname]

Definition of Substitutional Operands - RES

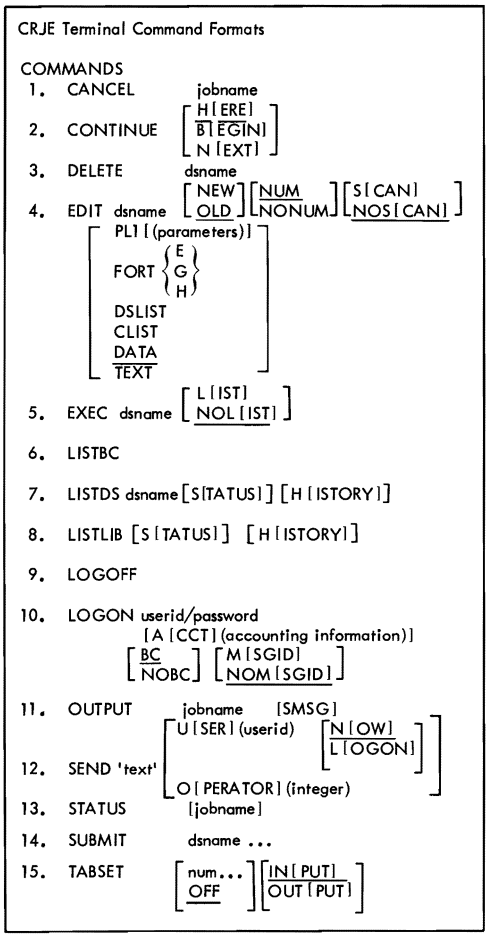
class	specifies an input or output class.
data	specifies information to be passed to the procedure.
devicetype	specifies a device type (for example, PR1).
id	specifies any unique one to eight character name that starts with a letter (except for Pnn or ALL).
inclass	specifies an input queue class.
jobname	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 decimal digit from 1 to 255.
nnn.aam	nnn specifies a workstation (1-200), aa identifies a device type (RD, PR, PU), m identifies a particular device.
nbs	specifies a decimal digit from 1 to 100 (indicates the number of pages to be backspaced).
nfs	specifies a decimal digit from 1 to 255 (indicates the number of pages to be spaced forward).
outclass	specifies an output class.
password	specifies an assigned sequence of one to eight alphameric characters.
Pnn	specifies the VS1 partition number in which the procedure was started.
pp	specifies numerical priority (decimal number from 0 to 13).
procname	specifies the name of a cataloged procedure.
qlist	specifies one to four queue classes (A-O for input queues, SOUT for the output queue, HOLD for the hold queue).
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 workstation.
text	specifies information to be entered in response to a message.
unit	specifies the symbolic unit address (for example, PR1) of an I/O device.
unitaddr	specifies the channel and unit address (cuu) 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.

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



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]]] [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]]

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 (con't)

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.

Syntax Checkers

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] }

PLEASE COMMENT

Use Reader's Comment Form

Linkage Editor

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
	//SYSIN	DD	dataset reference	
	//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 //SYSIN 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:

HIAR	-	hierarchy format; may be specified, but ignored as unsupported by OS/VS1.
SCTR	-	scatter loading; may be specified, but ignored as unsupported by OS/VS1.
DC	-	downward compatable; only needed when max record size required or grouping of output load module CSECTs not desired.
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 CSECT must be refreshable or REFER is ignored.
TEST	-	test; for TESTRAN or the TSO test command. Modules using TESTRAN should not be marked RENT, REUS, or REFR.
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=($value_1$, $value_2$)	-	size; $value_1$ is virtual storage available for linkage editor with minimum of 65,536 and default of 196,608. $Value_2$ 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 cannot 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.

Linkage Editor Execute Statement (con't)

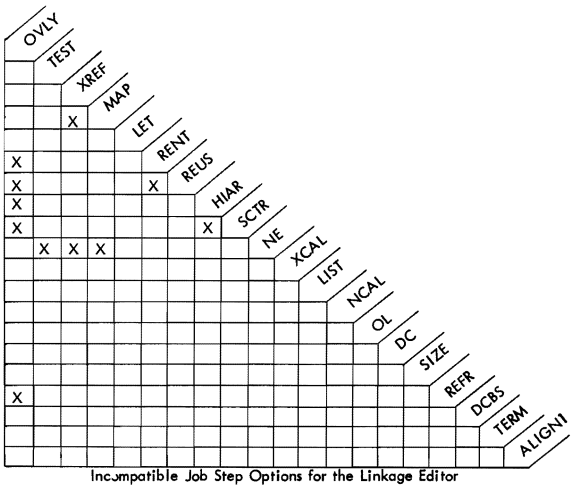
PARM default attributes for the linkage editor:

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, REGION value must at least value $_1$ +8K.



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 level F 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
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}, ...
SETSSI	xxxxxxxx

Linkage Editor Record Formats

The following record formats are used with the linkage editor:

- F -- The records are fixed length.
- FB -- The records are fixed length, and blocked.
- FBM -- The records are fixed length, blocked, and contain machine code control characters.
- FBS -- The records are fixed length, blocked, and written in standard blocks.
- FM -- The records are fixed length and contain machine code 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 ASCII control characters.

Linkage Editor 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 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 three loader names are:

1. LOADER
2. HEWLDRGO
3. IEWLDRGO

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 main 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 module. For FORTRAN, ALGOL, and PL/I, these entry points must be MAIN, IHIFSAIN, and IHENTRY, respectively.

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

NOTERM

Numbered diagnostic messages are not to be sent to the SYSTEM 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

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=FDSA 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 OPTCD=C is used to specify chained scheduling, an additional 2K (2048 bytes) of main 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 statement and data required by the loaded program must be included in the input deck.

Loader Macro

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 Main Storage Requirements (in bytes)	Comments
Loader Code	Control	400	--
	Processing	13250	--
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	--

Source Publications

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

- *OS/VS BTAM*, GC27-6980
- *OS/VS TCAM Programmer's Guide*, GC30-2034
- *OS/VS2 TSO Guide*, GC28-0644

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={I / M}][,MSG='text']	2715 ONLY
(Omit)	ASMTRTAB	tablename,...	
[symbol]	CHGNTRY	listaddr, listype, listposition, numchars, action	
[symbol]	{OPEN / CLOSE}	({dcb},...) [MF=L / MF=(E, listname)]	
[symbol]	CONFIGUR	[CORE={16 / 32}] [,PC={NO / YES}]	2715 ONLY
[symbol]	CTRGROUP	ctrno, [sro], [cttest], ID=absexp [,SROENAB={NO / YES}] [,CTINIT={NULL / NCT / UNASP}]	2715 ONLY
[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	

BTAM Macros (con't)

Name	Operation	Operand	
[symbol]	DEULIST	[DIGIT=(absexp1,absexp2)] [,LENGTH=absexp1] [,MSG='text']	2715 ONLY
symbol	DFTRMLST	list type, device-dependent operands	
[symbol]	IODEVICE	UNIT=type, ADDRESS=address, ADAPTER=type, TCU=teleprocu [,FEATURE=(feature1,feature2,...)] [,SETADDR=type] [,OBRCNT=n]	
symbol	LERB	nlines[,{([transmct] [,datack],[,intreq] [,nontto])}...]	
[symbol]	LERPRT	dcbaddr [,rln] [,cid] [,CLEAR=YES] [,CLEAR=NO]	
[symbol]	LOPEN	decbaddr	
[symbol]	ONLIST	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]	{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]	STEND		2715 ONLY
symbol	TGROUP	[TCn=(symboln [,E])]	2715 ONLY

BTAM Macros (con't)

Name	Operation	Operand	
[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 IN-SCRIBER ONLY
symbol	TRLIST	ROUTE= ([{DISK / CPU}][,LOG][{,NULL / ,absexp1}]) [TEXT={NO / YES}],TRID=absexp2	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	
[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					
	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
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
	entrylength		x										
	userlength		x										
	idcount		x										
	idsent											x	
	authsequence											x	
	controlvalue		x										
	userdata							x					

*See macro description for allowable values.

BTAM Macro Instruction Format (con't)

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*
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
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

*See macro description for allowable values.

BTAM Macro Instruction Format (con't)

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 (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
TRANSLATE	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								
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

*See macro description for allowable values.

BTAM Macro Instruction Format (con't)

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 (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			
CONFIGUR	CORE=												x
	PC=												x
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			
STEND	no operands												
TGROUP	TCn= tcode E	x											As Shown

*See macro description for allowable values.

2715 User Table Macro Instructions (con't)

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*
TRLIST	TRID=								x				
	ROUTE=												x
	LOG												As Shown
	NULL												As Shown
	asaddr								x				
	TEXT=												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): 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 (con't)

Binary Synchronous Communications

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

   IBM System/360[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

2. 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

[3]The remote System/360 may be a Model 25, 30, 40, 50, 65, 67 (operating in 65 mode), 75, 85, or 91.

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]	COUNTER	opfield
[symbol]	CUTOFF	integer
[symbol]	DATETIME	[DATE={NO / YES}] [,TIME={NO / YES}]
[symbol]	ERRORMSG	[mask] [,CONNECT={AND / OR}] [,DEST={destination name / opfield / ORIGIN / DESTIN}] ,DATA={message / fieldname} [,EXIT=name of routine]
[symbol]	FORWARD	[DEST={destname / opfield / (number) / PUT / * *}] [,EOA=characters] [,EXIT=name] [,THRESH=nn]
[symbol]	GET	dcbname [,areaname]
[symbol]	HOLD	[mask] [,RELEASE] [,INTVL=integer] [,CONNECT={AND / OR}] [,LEVEL={BLK / MSG}]
[symbol]	ICHNG	grpname,rln {areaname / ACT / DEACT [,PASSWRD=chars]}
[symbol]	ICOPY	grpname,rln,areaname
[symbol]	INBLOCK	[PATH=(opfield,switch)]
[symbol]	INBUF	[PATH=(opfield,switch)]
[symbol]	INEND	(no operands)
[symbol]	INHDR	[PATH=(opfield,switch)]

Name	Operation	Operands
[symbol]	INITIATE	[conchars[,BLANK={char|NO|YES}]]
[symbol]	INMSG	[PATH=(opfield,switch)]
[symbol]	INTRO	keyword operands ...,FEATURE=(...,{CONC|CONCO|NOCONC}) ,CPB={integer|0} ,ENVIRON={TSO|MIXED|TCAM} ,FEATURE=({NODIAL|DIAL},{NO2741|2741},{NOTIMER|TIMER}) ,MSMAX={integer|70} ,MSMIN={integer|50} ,MSUNITS={integer|0}
symbol	INVLIST	ORDER=(entry,...)[,EOT=hexchars][,CPUID=addr]
[symbol]	LOCK	{EXTEND|MESSAGE}[,conchars[,BLANK={YES|NO|char}]]
[symbol]	LOCOPT	opfield,{(register)|(15)}
[symbol]	LOG	{dcbname|typename}
typename	LOGTYPE	dcbname,BUFSIZE=size,QUEUES=form
[symbol]	MCOUNT	DCB={name|(r)}
[symbol]	MCPCLOSE	{QUICK|FLUSH}[,PASSWRD=chars]
[symbol]	MRELEASE	statname[,PASSWRD=chars]
[symbol]	MSGEDIT	((group1),(group2),...),BLANK={NO|char|YES}
[symbol]	MSGFORM	[BLOCK=integer][,SUBBLCK=integer] [,COUNT=integer][,SENDTRP={YES|NO}] [,ENDCHAR=subblock delimiter]
[symbol]	MSGGEN	[mask],{message|fieldname} [,CONNECT={AND|OR}] [,CODE={tablename|NO}]
[symbol]	MSGLIMIT	{integer|opfield}

TCAM Macros (con't)

Name	Operation	Operands
[symbol]	MSGTYPE	[conchars,[BLANK={YES/NO/char}]]
[symbol]	OPEN	(dcbname,,...)[,MF={L/E,listname/(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)]
[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)]
[symbol]	POINT	dcbname,address
[symbol]	PRIORITY	[integer][,conchars[,BLANK={YES/NO/char}]]
[symbol]	PUT	dcbname[,areaname]
[symbol]	QCOPY	termname,areaname
[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

Name	Operation	Operands
[symbol]	SCREEN	{WRE / WLA / WDC} [conchars [,BLANK={YES / NO / char}]]
[symbol]	SEQUENCE	(no operands)
[symbol]	SETEOF	[conchars [,BLANK={YES / NO / char}]]
[symbol]	SETEOM	[ENDCHAR={chars / opfield}] [,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)}]
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}][,TSOMH={YES / NO}]
[symbol]	TCHNG	termname,areaname [,PASSWRD=chars]
[symbol]	TCOPY	statname,areaname
[symbol]	TERMINAL	QBY={T / L},DCB=dcbname,RLN=integer,TERM=type ,QUEUES=form [,DIALNO={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 / NO}] [,DVCID={(CONC [,integer]) / chars / NONE}] [,QCNTRL=({ALL / (MSG[,{msgcount / 1}]) / bytecount} [,L] [,STATUS] [,char])]

TCAM Macros (con't)

Name	Operation	Operands
[symbol]	TERMINAL (continued)	[,CTBMAX=integer] [,SCRSIZE=(n,m)] [,FEATURE=([BREAK|NOBREAK],[ATTN|NOATTN][,TOSUPPR])]
[symbol]	TERRSET	(no operands)
[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,...)]
[symbol]	TTABLE	LAST=name [,MAXLEN=integer]
[symbol]	UNLOCK	[conchars[,BLANK={YES|NO|char}]]
[symbol]	WRITE	decbname,SF,dcbname,areaname,{length|'S'}

TCAM Macros (con't)

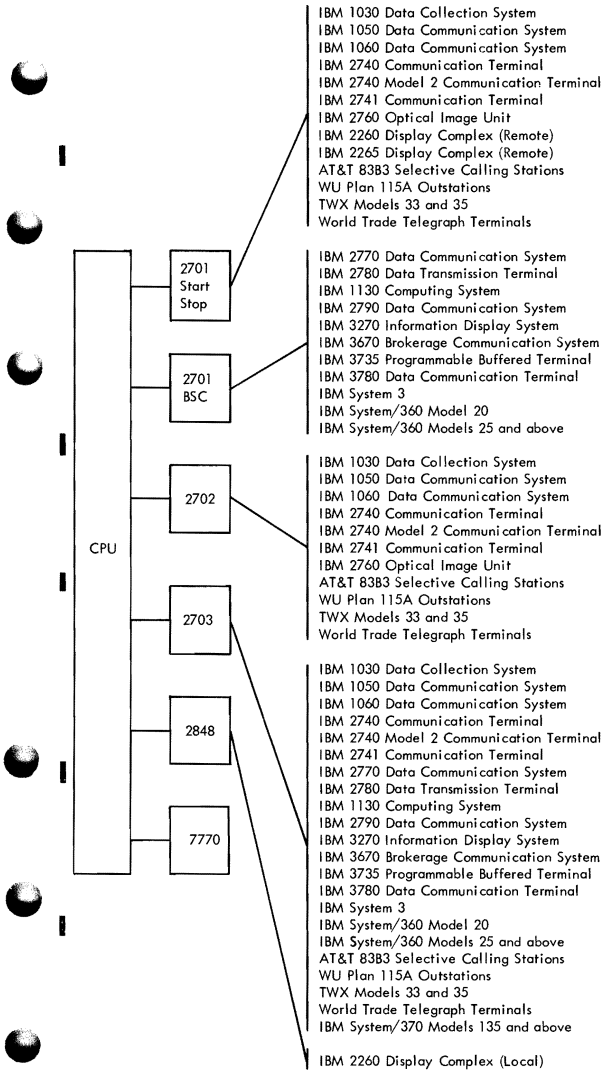
Name	Operation	Operands
[symbol]	CTBFORM	[opfield] [,DVCID={NO / YES}] [,ENDCHAR={NO / YES}]
[symbol]	INTRO	...,FEATURE=(..., {CONC / CONCO / NOCONC})
[symbol]	MRELEASE	statname [,PASSWRD=chars]
[symbol]	ORIGIN	[,FORM={ID / NAME}]
[symbol]	QACTION	TYPE={A / V},EXIT=name
symbol	TERMINAL	[,DVCID={(CONC [,integer]) / chars / NONE}]
		[,QCNTRL=({ALL / (MSG [,{msgcount / 1}]) / bytecount} [,L][,STATUS][,char])] [,CTBMAX=integer]
[symbol]	TGOTO	MH={name of MH / opfield}

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,LIST,{grpname,rln} {address} TP,OPTION,statname,opfldname TP,PRITERM TP,QUEUE,statname TP,SECTERM TP,TERM,statname
control chars	{HALT} {Z}	TP, {QUICK} {FLUSH}
control chars	{HOLD} {H}	TP=statname
control chars	{MODIFY} {F} {MODIFY} {F}	id,AUTOPOLL= {grpname,rln},OFF {address} id,AUTOPOLL= {grpname,rln},ON {address} id,INTERVAL=POLL,statname,data id,INTERVAL=SYSTEM id,INTERVAL=SYSTEM,data id,INTENSE=LINE,{grpname,rln}, {address} sense,count id,INTENSE=TERM,statname,sense,count id,OPERATOR={statname} {SYSCON} id,OPT=statname,opfldname,data id,TRACE={grpname,rln},OFF {address} id,TRACE={grpname,rln},ON {address}

TCAM Operator Commands (con't)

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

Device Configurations Supported by TCAM



Device Configurations Supported by TCAM

PLEASE COMMENT

Use Reader's Comment Form

Device Configurations Supported by TCAM (con't)

Station Type		Channel Type		TCU			Audio Response Unit	Line Type		Notes
		Multiplexer	Selector	IBM 2701 Data Adaptor Unit	IBM 2702 Transmission Control	IBM 2703 Transmission Control	IBM 7770 Model 3	Switched	Nonswitched	
IBM 1030 Data Collection System	Auto Poll	x			x	x			x	The IBM Digital Time Out feature cannot be attached through an IBM 2701 TCU.
		x		x	x	x			x	
IBM 1050 Data Communication System	Auto Poll	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	
IBM 2260-2848 Display Complex (Remote)		x		x					x	
IBM 2260-2848 Display Complex (Local)		x	x							
IBM 2265-2845 Display Complex (Remote)		x		x					x	
IBM 2740 Model 1 Communication Terminal	Auto Poll	x			x	x			x	Two Types: 2740 with station control 2740 with station control and record checking
		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		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	Four Types: 2740 2740 with record checking 2740 with buffer receive 2740 without buffer receive (requires lines slowdown feature)
		x		x	x	x			x	Four types: 2740 2740 with record checking 2740 with buffer receive 2740 without buffer receive
IBM 2741 Communication Terminal		x		x	x	x		x	x	

Device Configurations Supported by TCAM

Device Configurations Supported by TCAM (con't)

Station Type	Channel Type		TCU			Audio Response Unit	Line Type		Notes
	Multiplexor	Selector	IBM 2701 Data Adapter Unit	IBM 2702 Transmission Control	IBM 2703 Transmission Control	IBM 7770 Model 3	Switched	Nonswitched	
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	BSC transmission using either ASCII or EBCDIC
IBM 2780 Data Transmission Terminal	x		x		x		x	x	BSC transmission using ASCII, EBCDIC, or 6 bitcode.
IBM 2790 Data Communication System	x		x		x		x	x	
IBM 3270 Information Display System	x		x		x			x	
IBM 3670 Brokerage Communication System	x		x		x			x	BSC transmission using EBCDIC
IBM 3735 Programmable Buffered Terminal	x		x		x		x	x	Either ASCII or EBCDIC
IBM 3780 Data Communication Terminal	x		x		x		x	x	BSC transmission using ASCII or EBCDIC
IBM 1130 Computing System	x		x		x		x	x	BSC transmission
IBM System 3	x		x		x		x	x	Code TERM=202A or TERM= 202B on TERMINAL macro: inquiry/response not supported
IBM System/360 Model 20	x		x		x		x	x	BSC transmission using either ASCII or EBCDIC
IBM System/360 Models 25 and above	x		x		x		x	x	BSC transmission and point-to point lines only
AT&T 8383 Selective Calling Stations	x		x	x	x			x	
Western Union Plan 115A Outstations	x		x	x	x			x	
TWX Models 33 and 35	x		x	x	x		x		Teletype terminal, dial service (8-level code)
World Trade Telegraph Terminals	x		x	x	x			x	Control unit must incorporate a WTTA
Audio Terminals	x					x	x		Example: IBM 2721 Portable Audio Terminal

Device Configuration Supported by TCAM

TCAM Macros Modified for TSO		
Name	Operation	Operand
linedcb	DCB	MH=mhname,CPRI={R/E/S}[,TRANS={table/list}]
[symbol]	INTRO	ENVIRON={TSO/MIXED/TCAM},CPB={integer/0} ,MSMIN={integer/50},MSMAX={integer/70} ,MSUNITS={integer/0} ,FEATURE=({NODIAL/DIAL},{NO2741/2741},{NOTIMER/TIMER})
[symbol]	LOG	dcbname
[symbol]	PATH	switch,opfield
symbol	TERMINAL	TERM=type,QUEUES=form[,SCRSIZE=(n,m)] [,FEATURE=([BREAK/NOBREAK],[ATTN/NOATTN],[TOSUPPR])]

TCAM Macro Instructions not used in TSO MH				
CHECKPT	DATETIME	ERRORMSG	FORWARD	HOLD
INITIATE	LOCK	MSGEDIT	MSGFORM	MSGLIMIT
MSGTYPE	ORIGIN	PRIORITY	REDIRECT	SCREEN
SEQUENCE	SETEOF	SETSCAN	UNLOCK	RETRY
				SETEOM

TSO/TCAM (con't)

TSO Macro Instructions		
Name	Operation	Operands
	ATTEN	(no operands)
	CARRIAGE	(no operands)
	HANGUP	(no operands)
	LOGON	(no operands)
	SIMATTN	(no operands)
mhname	STARTMH	[,TSOMH = { YES / NO }]
symbol	TRANSLIST	{ LIST = ({ table / username },...) / L2741 = ({ table / username },...),L1050 = ({ table / username },...) } ,CHARS = (control chars,...) ,OPFLD = opfldname
[symbol]	TSINPUT	(no operands)

TSO Operator Commands		
control chars	operation	operands
cont chars	{ DISPLAY / D }	TP,ACT,{ grpname,rln / address }
cont chars	{ DISPLAY / D }	TP,QUEUE,statname
cont chars	{ HALT / Z }	TP,{ QUICK / FLUSH }
cont chars	{ MODIFY / F }	{ [procname.]identifier / jobname } ,INTERVAL = SYSTEM
cont chars	{ VARY / V }	{ (grpname,rln) / grpname / address },OFFTP,{ C / I }

Source Publications

Additional information about OS/VS Utilities is contained in *IBM System OS/VS Utilities,* GC35-0005.

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-data-group 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 (con't)

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 volume	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	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
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
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:

- Analyzes tracks on direct access volumes.
- Assigns alternate tracks to a direct access volume.
- Gets alternate tracks on a direct access volume.
- Initializes a direct access volume.

Job Control Statements

Because IBCDASDI is an independent utility, 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 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
2314	25
2319	25
2305-1	18
2305-2	34
3330	39

IBCDASDI (con't)

Format

Name	Operation	Operands
[label]	JOB	[user-information]
[label]	MSG	TODEV=xxxx TODADDR=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, 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.

Job Control Statements

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

Control Statements

JOB	indicates the beginning of an ICAPRTBL job.
DFN	defines the address of the 2311.
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=$\begin{cases} Y \\ N \end{cases}$
[ucsname]	UCS	ucs-image
[fcbname]	FCB	LPI=$\begin{cases} 6 \\ 8 \end{cases}$ LNCH=((l,c)[,(l,c)...]) FORMEND=x
[label]	END	[user-information]

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.	B1A	Printer not ready.
B07	Incorrect END statement.	B1B	Printer unit check (display low main storage location 2 through 7 for sense information).
B0A	External interrupt.		
B0B	Program check interrupt.		
B0C	Machine check interrupt.	B1C	Printer channel error.
B11	Reader not online.	B1D	No device end on printer.

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	[INHDR=routinename] [INTLR=routinename] [ERROR=routinename] [PRECOMP=routinename]
[label]	LABELS	DATA={YES / NO / ALL / ONLY}

IEBCOPY

IEBCOPY, a data set utility:

- Compresses in place, a partitioned data set.
- Converts to partitioned, a sequential 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.
- 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 (con't)

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={[(]name1[,(name2,,R[,newname[,R])]...[)] [,(name2,newname[,R])] ((name1{,newname[,R] ,,R} [,name2]...)}
[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	bc.<(+|&|$*); -/,%__>?:#@'="¬ A...Z 0...9
Randon 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	{[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]}*

*Use at least one of the optional parameters.

IEBDG (con't)

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 -- alphabetic.	
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	[START=jobname] [TYPE={POSITION / INCLUDE / EXCLUDE}] [STEPNAME=({name / name-name}[,{name / name-name}],...)] [NOPRINT]

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 (con't)

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	[MAXNAME=n] [MAXFLDS=n] [MAXGPS=n] [MAXLITS=n]
[label]	EXITS	[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]

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[,N1 [,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.
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.

IEBPTPCH (con't)

Format

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

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)
//SYUT2       DD         data set definition (output data set-valid records)
//SYUT3       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.

IEBTCRIN (con't)

MTDI Codes from TCR

Bit Positions 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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					!	¬	?	"								

Bit Positions 0, 1

Bit Positions 2, 3

First Hexadecimal Digit

Special Control:
LZ = Left zero fill
DUP = Duplicate
LZS = Left zero s
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.

MTST Codes from TCR

Bit Positions 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	z	cr	5	0	l	tab	'	s	src							
0001	1	2	sw	6	9	.	as	i	w								
0010	2	t		e	h	¡	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	,									

Bit Positions 0, 1
Bit Positions 2, 3
First Hexadecimal Digit

This figure represents the character set and control codes as read from an MTST created cartridge.

cr and CR = Carrier return code
sw and SW = Switch code
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 = Backspace
stx and STX = Stop transfer
src and SRC = Search

IEBTCRIN (con't)

MTST Codes after Translation by IEBTCRIN with TRANS = STDCL

Bit Position 4, 5, 6, 7	Second Hexadecimal Digit	00				01				10				11			
		00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	x		G	P	X	7
1000	8									h	q	y		H	Q	Y	8
1001	9									i	r	z		I	R	Z	9
1010	A					¢	!		:								
1011	B					.	$	,	#								
1100	C						*	%	@								
1101	D	CR				(	)	_	'								
1110	E		SRC			+	;		=		±						
1111	F							?	"								

Bit Positions 0,1

Bit Positions 2,3

FirstHexadecimalDigit

TAB = Tab code
CR = Carrier return
BSP = Backspace
SRC = Search
STX = Stop transfer
SP = Space

Note: The STDUC option permits translating both lowercase and uppercase alphabetic characters to uppercase.

IEBTCRIN (con't)

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	[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.

Name	Operation	Operands	Comments
[label]	EXITS	[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.

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)
//SYSUT2    DD      data set definition (output data set)
//SYSIN     DD      {*   }
                    {DATA}

./          "IEBUPDTE control statements"
(optional data or label statements)
/*
```

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 / REPL / CHANGE / REPRO}	[LIST=ALL] [SEQFLD=dd1] [NEW=PO] [NEW=PS] [MEMBER=cccccccc] [COLUMN=dd] [UPDATE=INPLACE]	Applicable to partitioned or sequential organization.
			[INHDR=cccccccc] [INTLR=cccccccc] [OUTHDR=cccccccc] [OUTTLR=cccccccc] [TOTAL=(routinename,size)]	Applicable to sequential organization only.
			[NAME=cccccccc] [LEVEL=hh] [SOURCE=x] [SSI=hhhhhhhh]	Applicable to partitioned organization only.

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		Operands	
./	[label]	{NUMBER / DELETE}	[SEQ1=cccccccc] [SEQ2=cccccccc]	Used with the NUMBER or DELETE statement.
			[SEQ1=ALL] [NEW1=cccccccc] [INCR=cccccccc] [INSERT=YES]	Used only with the NUMBER statement.

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.

IEBUPDTE (con't)

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=bbbbccccchhhhrrkkdddd[S]
```

or

```
VTOC=bbbbccccchhhhrrkkdddd
```

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}]
[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=ccchhhh
[label]	DUMP	FROMDD=ddname TODD=(ddname,...) [CPYVOLID={YES / NO}] [BEGIN=ccchhhh] [END=ccchhhh] [PURGE={YES / NO}]
[label]	RESTORE	TODD=(ddname,...) FROMDD=ddname [CPYVOLID={YES / NO}] [PURGE={YES / NO}]
[label]	IPLTXT	

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 (VS1 only):

- 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	[VOL=device=serial] [NODE=name]
[label]	LISTPDS	DSNAME=(name[,name]...) [VOL=device=serial] {DUMP } {FORMAT}
[label]	LISTVTOC	{DUMP } {FORMAT} [DATE=dddyy] [VOL=device=serial] [DSNAME=(name[,name]...)

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.
- 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.

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.

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]

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.
- Creates an index.
- Deletes a password.
- Deletes an index structure.
- Lists a password entry.
- Password-protects add, delete, or lists password operations.
- Renames a partitioned data set member.
- Renames a sequential or partitioned data set.
- 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 card. 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  }'] ]
                                 {,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 (con't)

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 (con't)

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

IEHSTATR

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.
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=	ccchhhh	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 (con't)

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.

Definition of Operands (con't)

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.
ERRORT=	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.
FIELD=	conversion	specifies a two-byte code that indicates the type of conversion to be performed on this field.

Definition of Operands (con't)

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 (con't)

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 (con't)

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 (1-280), 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 (con't)

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 (con't)

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 (con't)

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 (con't)

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 of 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 (con't)

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.
=	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.
TARTLOC=	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 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,...) ddname (ddname,...)	specifies the channel and unit address of the direct access device containing the volume to be processed. 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 (con't)

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.
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 (con't)

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.

PLEASE COMMENT

Use Reader's Comment Form

Source Publications

This section provides a listing of the base publications that support OS/VS1 and OS/VS2. Applicable TNL's and suffix numbers are not included.

If additional information is required, refer to the *OS/VS1 Release 2 Guide*, GC24-5097; the *OS/VS2 Release 1 Guide*, GC28-0601; or the *IBM System/360 and System/370 Bibliography*, GA22-6822.

OS/VS1 Publication Support

General Publications

IBM Data Processing Glossary	GC20-1699
Introduction to Virtual Storage in System/370	GR20-4260
IBM System/370 System Summary	GA22-7001

System Publications

DOS to OS/MFT, OS/MVT, OS/VS1 Management Planning Guide	GC24-5082
DOS to OS/VS1 Implementation Guide	GC24-5095
DOS to OS/VS1 Planning and Use Guide	GC24-5090
OS/VS Programmer's Reference Digest	GC24-5091
OS/VS1 Release 2 Guide	GC24-5097
OS/VS1 Storage Estimates	GC24-5094
OS/VS1 System Data Areas	SY28-0605

Operator's Library Publications

OS/VS Console Configurations	GC38-0120
OS/VS1 CRJE	GC38-0335
OS/VS Display Consoles	GC38-0255
OS/VS1 Reference	GC38-0110
OS/VS1 RES	GC38-0330
S/370 Model 135 Operating Procedures	GC38-0005
S/370 Model 145 Operating Procedures	GC38-0015
S/370 Model 158 Operating Procedures	GC38-0025

Job Management Publications

OS/VS JCL Services	GC28-0617
OS/VS JCL Reference	GC28-0618
OS/VS System Management Facilities (SMF)	GC35-0004
OS/VS1 Job Management Logic	SY24-5161

Supervisor Publications

OS/VS Supervisor Services and Macro Instructions	GC27-6979
OS/VS1 IPL and NIP Logic	SY24-5160
OS/VS1 Supervisor Logic	SY24-5155

Data Management Publications

OS/VS I/O Supervisor Logic	SY24-5156
OS/VS Checkpoint/Restart	GC26-3784
OS/VS Data Management for System Programmers	GC28-0631
OS/VS Data Management Macro Instructions	GC26-3793
OS/VS Data Management Services Guide	GC26-3783
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
OS/VS Tape Labels	GC26-3795
OS/VS BDAM Logic	SY26-3789
OS BSAM Logic for IBM 1419/1275	GY21-0012
OS/VS Catalog Management Logic	SY35-0003
OS/VS Checkpoint/Restart Logic	SY24-5159
OS/VS DADSM Logic	SY26-3787
OS Data Management Macro Logic for IBM 1285/1287/1288	GY21-0013
OS/VS ISAM Logic	SY26-3786
OS/VS Open/Close/EOV Logic	SY26-3785
OS/VS SAM Logic	SY26-3788

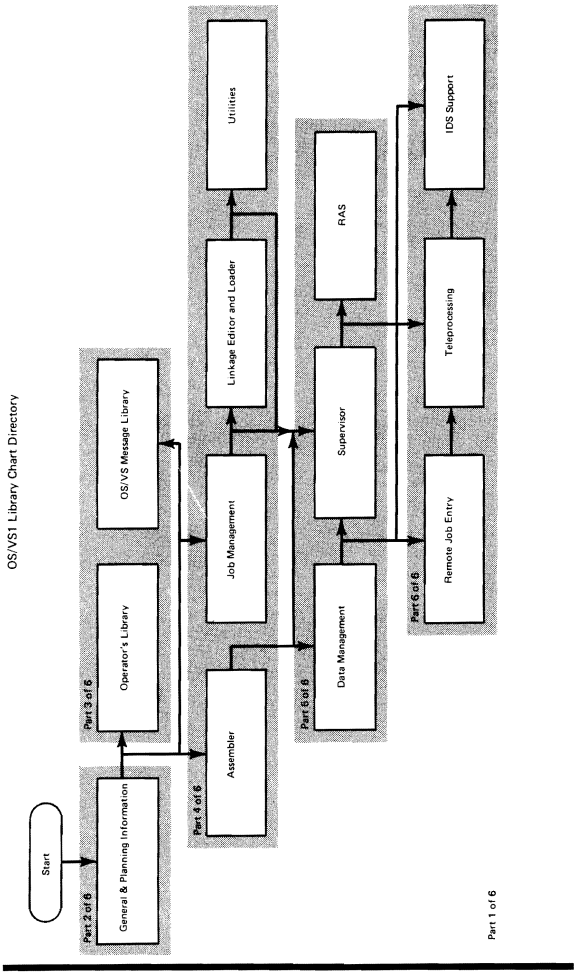
RAS Publications

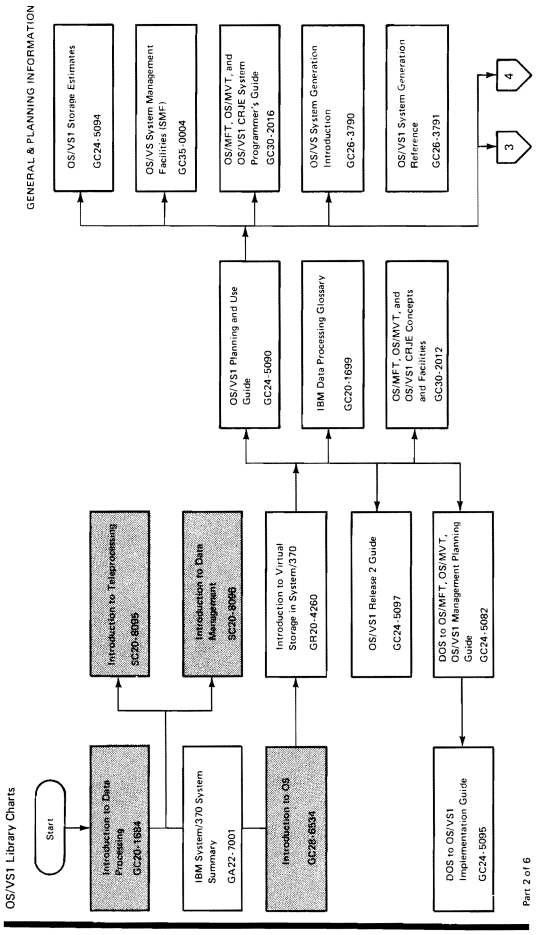
OS/VS1 Debugging Guide	GC24-5093
OS/VS1 OLTEP	GC28-0666
OS/VS Service Aids	GC28-0633
OS/VS1 OLTEP Logic	SY28-0662
OS/VS Service Aids Reference Summary	GX28-0634
OS/VS Recovery Management Support Logic	SY27-7239
OS/VS1 Service Aids Logic	SY28-0635
OS/VS SYS1.LOGREC Error Recording	GC28-0638
OS/VS SYS1.LOGREC Error Recording, LOGIC	SY28-0639

OS/VS1 Publication Support (con't)

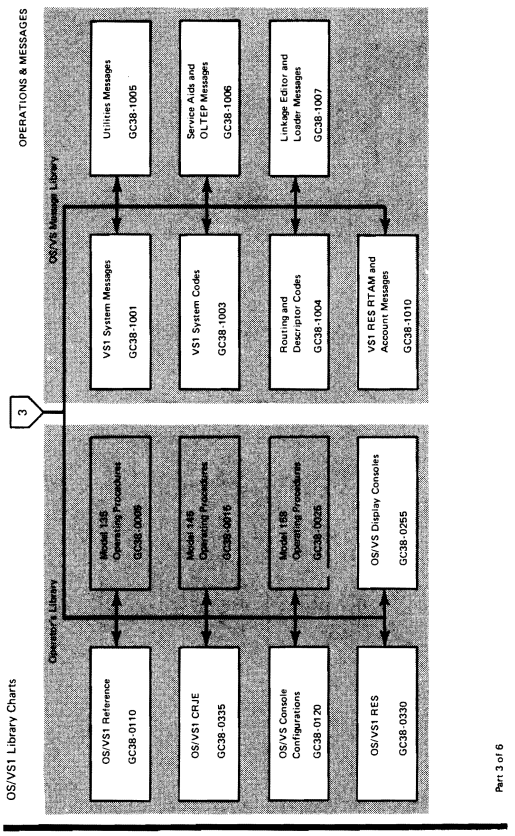
Message Library Publications	
Linkage Editor and Loader Messages	GC38-1007
Routing and Descriptor Codes	GC38-1004
Service Aids and OLTEP Messages	GC38-1006
VS1 System Codes	GC38-1003
VS1 System Messages	GC38-1001
Utilities Messages	GC28-1005
VS1 RES RTAM and Account Messages	GC38-1010
Support Component Publications	
OS/VS and DOS/VS Assembler Language	GC33-4010
OS/VS Assembler Programmer's Guide	GC33-4021
OS/VS Linkage Editor and Loader	GC26-3813
OS/VS System Generation Introduction	GC26-3790
OS/VS1 System Generation Reference	GC26-3791
OS/VS Utilities	GC35-0005
OS/VS Assembler Logic	SY33-8041
OS/VS Linkage Editor Logic	SY26-3815
OS/VS Loader Logic	SY26-3814
Utilities Logic	SY35-0005
Teleprocessing Publications	
OS/VS BTAM	GC27-6980
OS/VS BTAM Logic	SY27-7246
Information Display System Publications	
OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit	GC27-6971
OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment)	GC27-6972
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I	GC27-6973
OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP	GC27-6974
OS/VS Graphic Access Method Logic	SY27-7240
OS/VS Graphics Problem-Oriented Routines Logic	SY27-7241
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I Logic	SY27-7242
Remote Entry Publications	
OS/MFT, and OS/MVT, and OS/VS1 CRJE Terminal user's Guide	GC30-2014
OS/MFT, OS/MVT, and OS/VS1 CRJE Concepts and Facilities	GC30-2012
OS/MFT, OS/MVT, and OS/VS1 CRJE System Programmer's Guide	GC30-2016
OS/MFT, OS/MVT, and OS/VS1 CRJE Logic	GY30-2011
OS/VS1 RES Workstation User's Guide	GC28-6879
OS/VS1 RES System Programmer's Guide	GC28-6878
OS/VS1 RES RTAM and Workstation Support Logic	SY28-6849
OS/VS1 RES Account Facility Logic	SY28-0660

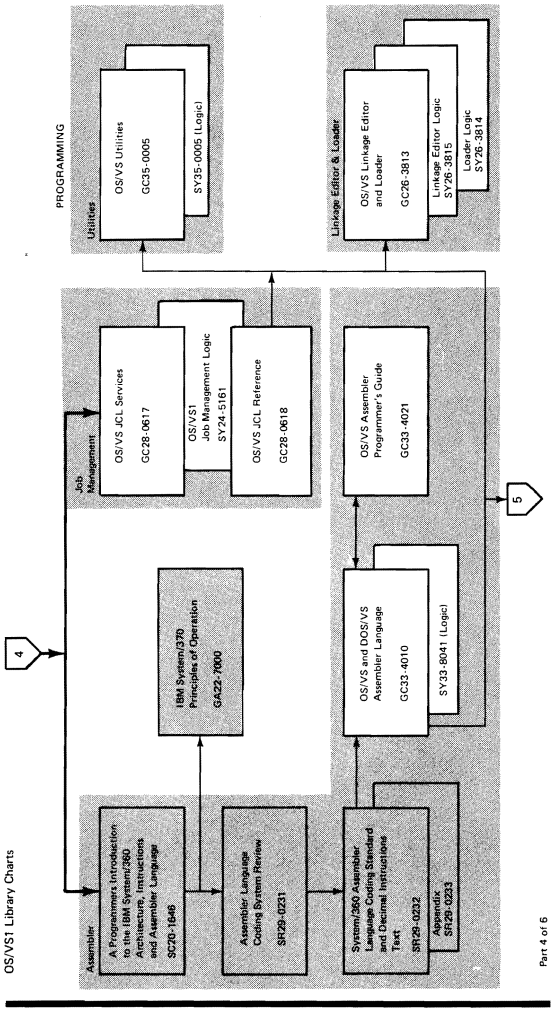
OS/VS1 Library Chart Directory

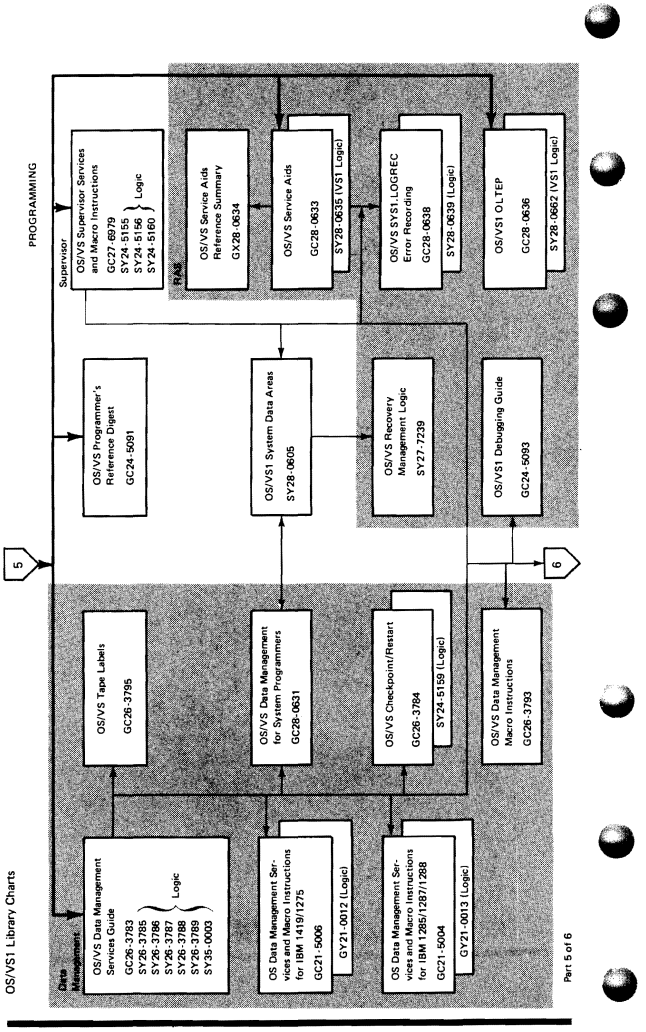


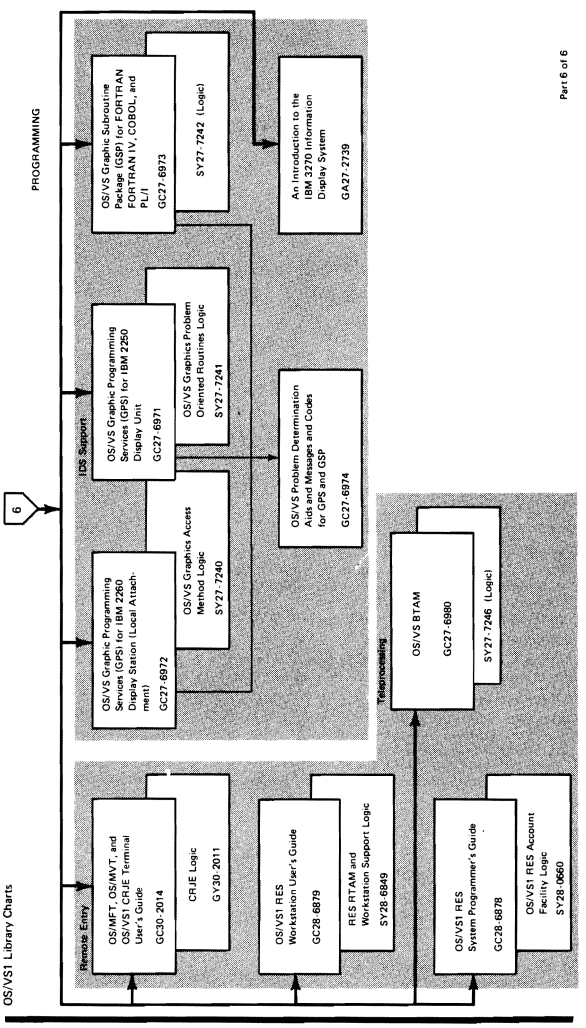

GENERAL & PLANNING INFORMATION
OS/VS1 Storage Estimates GC24-5094
OS/VS System Management Facilities (SMF) GC35-0004
OS/MFT, OS/MVT, and OS/VS1 CRJE System Programmer's Guide GC30-2016
OS/VS System Generation Introduction GC26-3790
OS/VS1 System Generation Reference GC26-3791
4
3
OS/VS1 Planning and Use Guide GC24-5090
IBM Data Processing Glossary GC20-1699
OS/MFT, OS/MVT, and OS/VS1 CRJE Concepts and Facilities GC30-2012
Introduction to Teleprocessing SC20-8095
Introduction to Data Management SC20-8096
Introduction to Virtual Storage in System/370 GR20-4260
OS/VS1 Release 2 Guide GC24-5097
DOS to OS/MFT, OS/MVT, OS/VS1 Management Planning Guide GC24-5082
OS/VS1 Library Charts
Start
Introduction to Data Processing GC20-1684
IBM System/370 System Summary GA22-7001
Introduction to OS GC28-6534
DOS to OS/VS1 Implementation Guide GC24-5095
Part 2 of 6

OS/VS1 Library Charts (con't)




OS/VS1 Library Charts
PROGRAMMING
Utilities
OS/VS Utilities
GC35-0005
SY35-0005 (Logic)
Linkage Editor & Loader
OS/VS Linkage Editor and Loader
GC26-3813
Linkage Editor Logic
SY26-3815
Loader Logic
SY26-3814
Job Management
OS/VS JCL Services
GC28-0617
OS/VS1 Job Management Logic
SY24-5161
OS/VS JCL Reference
GC28-0618
OS/VS Assembler Programmer's Guide
GC33-4021
4
IBM System/370 Principles of Operation
GA22-7000
5
OS/VS and DOS/VS Assembler Language
GC33-4010
SY33-8041 (Logic)
Assembler
A Programmer's Introduction to the IBM System/360 Architecture, Instructions and Assembler Language
SC20-1646
Assembler Language Coding System Review
SR29-0231
System/360 Assembler Language Coding Standard and Decimal Instructions Text
SR29-0232
Appendix
SR29-0233
Part 4 of 6


OS/VS1 Library Charts
PROGRAMMING
Supervisor
OS/VS Supervisor Services and Macro Instructions
GC27-6979
SY24-5155
SY24-5156
SY24-5160
Logic
RAS
OS/VS Service Aids Reference Summary
GX28-0634
OS/VS Service Aids
GC28-0633
SY28-0635 (VS1 Logic)
OS/VS SYS1.LOGREC Error Recording
GC28-0638
SY28-0639 (Logic)
OS/VS1 OLTEP
GC28-0636
SY28-0662 (VS1 Logic)
OS/VS Programmer's Reference Digest
GC24-5091
OS/VS1 System Data Areas
SY28-0605
OS/VS Recovery Management Logic
SY27-7239
OS/VS1 Debugging Guide
GC24-5093
5
6
Data Management
OS/VS Tape Labels
GC26-3795
OS/VS Data Management for System Programmers
GC28-0631
OS/VS Checkpoint/Restart
GC26-3784
SY24-5159 (Logic)
OS/VS Data Management Macro Instructions
GC26-3793
OS/VS Data Management Services Guide
GC26-3783
SY26-3785
SY26-3786
SY26-3787
SY26-3788
SY26-3789
SY35-0003
Logic
OS Data Management Services and Macro Instructions for IBM 1419/1275
GC21-5006
GY21-0012 (Logic)
OS Data Management Services and Macro Instructions for IBM 1285/1287/1288
GC21-5004
GY21-0013 (Logic)
Part 5 of 6


OS/VS1 Library Charts
PROGRAMMING
6
Remote Entry
OS/MFT, OS/MVT, and OS/VS1 CRJE Terminal User's Guide GC30-2014
CRJE Logic GY30-2011
OS/VS1 RES Workstation User's Guide GC28-6879
RES RTAM and Workstation Support Logic SY28-6849
IDS Support
OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment) GC27-6972
OS/VS Graphics Access Method Logic SY27-7240
OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit GC27-6971
OS/VS Graphics Problem Oriented Routines Logic SY27-7241
OS/VS Graphic Subroutine Package (GSP) for FORTRAN FORTRAN IV, COBOL, and PL/I GC27-6973
SY27-7242 (Logic)
An Introduction to the IBM 3270 Information Display System GA27-2739
OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP GC27-6974
Telecommunications
OS/VS BTAM GC27-6980
SY27-7246 (Logic)
OS/VS1 RES System Programmer's Guide GC28-6878
OS/VS1 RES Account Facility Logic SY28-0660
Part 6 of 6

OS/VS2 Publication Support

Planning and Implementing a VS2 System

OS/VS2 Planning and Use Guide	GC28-0600
OS/VS2 Release 1 Guide	GC28-0601
OS/VS2 Storage Estimates	GC28-0604
OS/VS System Generation Introduction	GC26-3790
OS/VS2 System Generation Reference	GC26-3792
OS/VS2 TSO Guide	GC28-0644
OS/VS System Management Facilities (SMF)	GC35-0004
OS/VS Checkpoint/Restart	GC26-3784
OS/VS Data Management for System Programmers	GC28-0631
OS/VS Tape Labels	GC26-3795
OS/VS Virtual Storage Access Method (VSAM) Planning Guide	GC26-3799
OS TCAM Concepts and Facilities	GC30-2022

Operating a VS2 System

Operator's Library: OS/VS2 Reference	GC38-0210
Operator's Library: OS/VS Console Configurations	GC38-0120
Operator's Library: OS/VS2 Display Consoles	GC38-0260
Operator's Library: OS/VS TCAM	GC38-0305
Operator's Library: OS/VS2 TSO	GC38-0220
Operator's Library: S/370 Model 145 Procedures	GC38-0015

Application Programming in VS2

OS/VS Programmer's Reference Digest	GC24-5091
OS/VS JCL Services	GC28-0617
OS/VS JCL Reference	GC28-0618
OS/VS Data Management Services Guide	GC26-3783
OS/VS Data Management Macro Instructions	GC26-3793
OS/VS Supervisor Services and Macro Instructions	GC27-6979
OS/VS and DOS/VS Assembler Language	GC33-4010
OS/VS Assembler Programmer's Guide	GC33-4021
OS/VS Linkage Editor and Loader	GC26-3813
OS/VS Utilities	GC35-0005

Teleprocessing Applications

OS/VS BTAM	GC27-6980
OS/VS TCAM Programmer's Guide	GC30-2034
IBM 3735 Programmer's Guide (OS and DOS Systems)	GC30-3001

Graphics Applications

OS/VS Graphic Programming Services (GPS) for IBM 2250 Display Unit	GC27-6971
OS/VS Graphic Programming Services (GPS) for IBM 2260 Display Station (Local Attachment)	GC27-6972
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I	GC27-6973

OCR/MICR Applications

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

TSO Applications

OS/VS TSO Terminal User's Guide	GC28-0645
OS/VS TSO Command Language Reference	GC28-0646
OS/VS TSO Command Language Reference Summary	GX28-0647
OS/MVT and OS/VS2 TSO Terminals	GC28-6762
OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor	GC28-0648

Problem Determination

OS/VS Message Library: VS2 System Messages	GC38-1002
OS/VS Message Library: VS2 System Codes	GC38-1008
OS/VS Message Library: Service Aids and OLTEP Messages	GC38-1006
OS/VS Message Library: Utilities Messages	GC38-1005
OS/VS Message Library: Linkage Editor and Loader Messages	GC38-1007
OS/VS Message Library: VS2 TSO Messages	GC38-1009
OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP	GC27-6974
OS/VS Message Library: Routing and Descriptor Codes	GC38-1004
OS/VS Service Aids	GC28-0633
OS/VS Service Aids Reference Summary	GX28-0634
OS/VS2 Debugging Guide	GC28-0632
OS/VS OLTEP	GC28-0636
OS/VS SYS1. LOGREC Error Recording	GC28-0638

OS/VS2 Publication Support (con't)

Maintaining and Modifying a VS2 System (Program Logic)

Control Program Logic

OS/VS Master Index of Logic	GY28-0603
OS/VS2 Planning and Use Guide	GC28-0600
OS/VS2 Job Management Logic	SY28-0620
OS/VS2 IPL and NIP Logic	SY27-7243
OS/VS2 Supervisor Logic	SY27-7244
OS/VS2 Checkpoint/Restart Logic	SY26-3820
OS/VS2 System Data Areas	SY28-0606

Data Management Logic

OS/VS SAM Logic	SY26-3788
OS/VS ISAM Logic	SY26-3786
OS/VS BDAM Logic	SY26-3789
OS/VS2 I/O Supervisor Logic	SY26-3823
OS/VS Open/Close/EOV Logic	SY26-3785
OS/VS DADSM Logic	SY26-3787
OS/VS Catalog Management Logic	SY35-0003
OS BSAM Logic for IBM 1419/1275	GY21-0012
OS Data Management Macro Logic for IBM 1285/1287/1288	GY21-0013

Teleprocessing Logic

OS/VS BTAM Logic	SY27-7246
OS/VS TCAM Logic	SY30-2039
IBM 3735 Programmable Buffered Terminal Form Description Macro Instructions and Form Description Utility Program Logic Manual IOS and DOS Systems	GY30-3000

Graphics Logic

OS/VS Graphics Access Method Logic	SY27-7240
OS/VS Graphics Problem Oriented Routines Logic	SY27-7241
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I	SY27-7242

RAS Logic

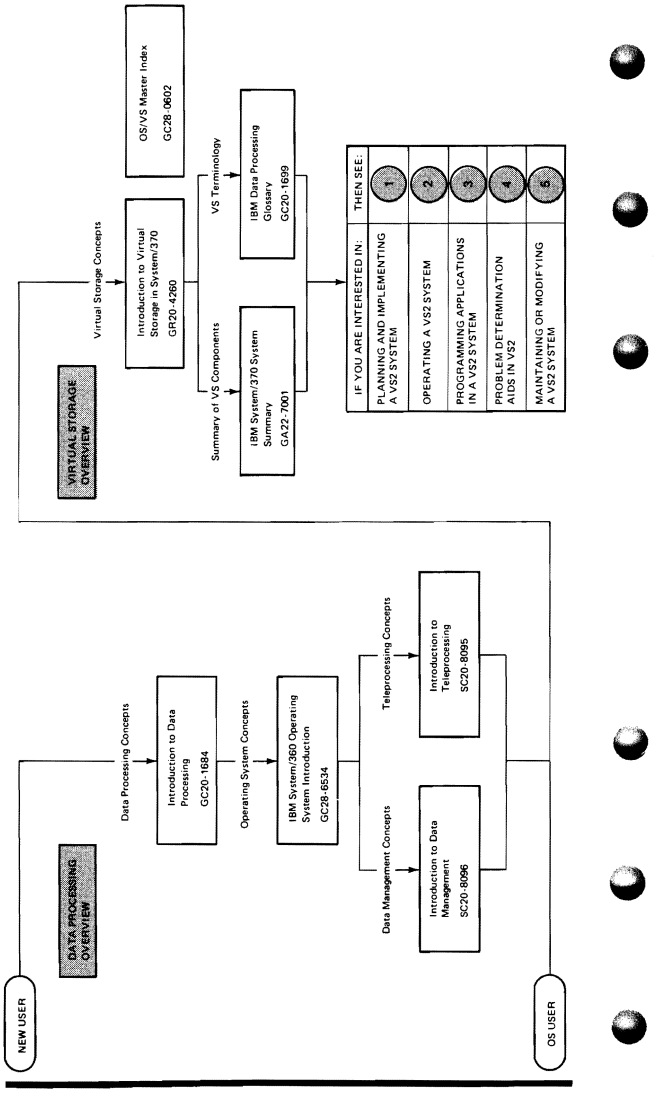
OS/VS2 Recovery Management Support Logic	SY27-7252
OS/VS2 Service Aids Logic	SY28-0643
OS/VS OLTEP Logic	SY28-0637
OS/VS SYS1.LOGREC Error Recording Logic	SY28-0639

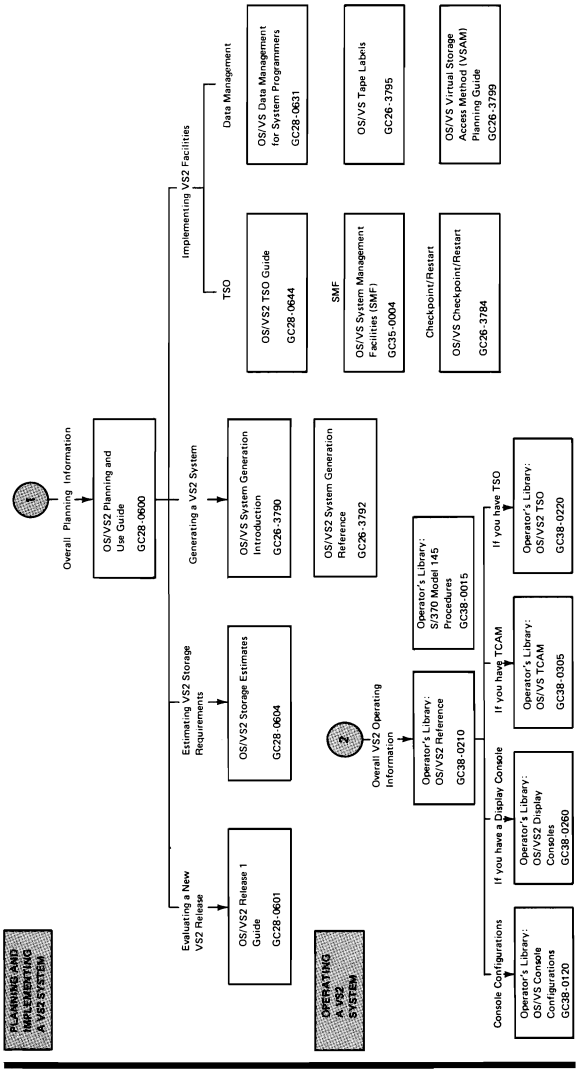
Support Program Logic (Assembler, Linkage Editor, Loader, and Utilities)

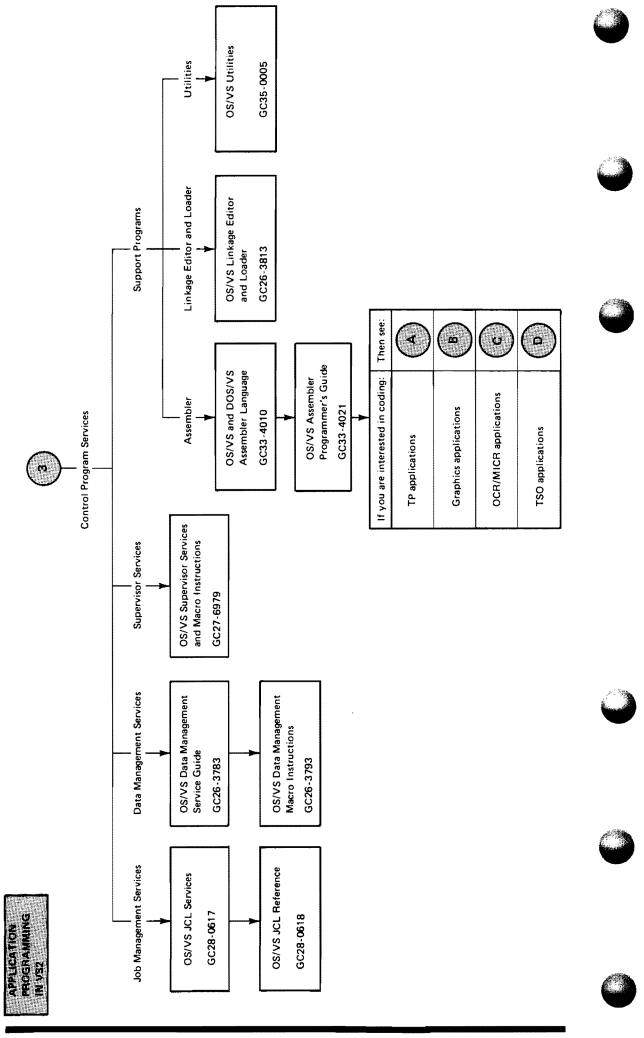
OS/VS Assembler Logic	SY33-8041
OS/VS Linkage Editor Logic	SY26-3815
OS/VS Loader Logic	SY26-3814
OS/VS Utilities Logic	SY35-0005

TSO Logic

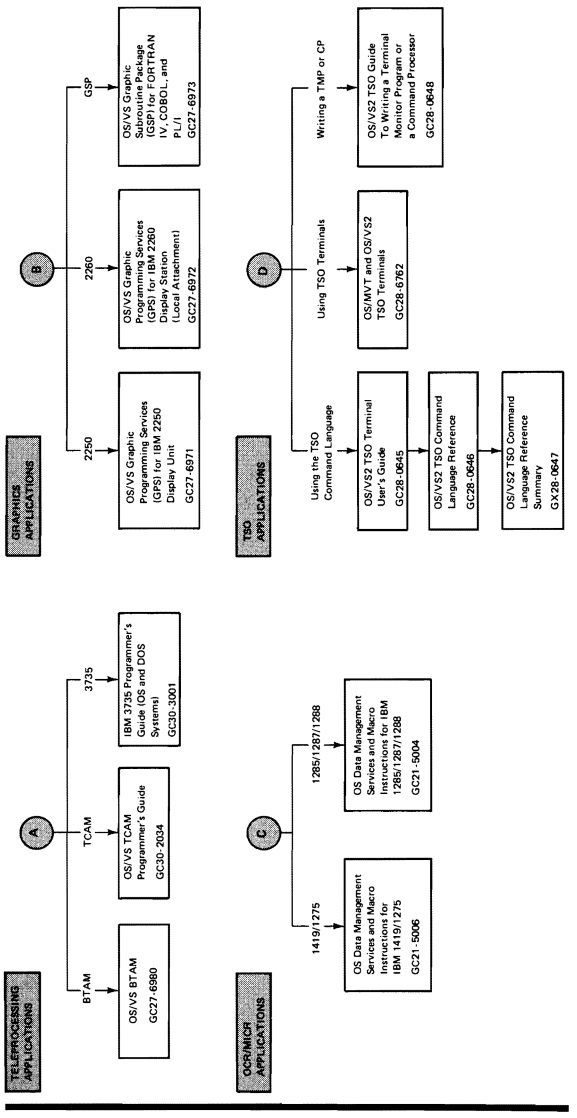
OS/VS2 TSO Control Program Logic	SY28-0649
OS/VS2 TSO Terminal Monitor Program and Service Routines Logic	SY28-0650
OS/VS2 TSO Command Processor Logic Volume I - ACCOUNT	SY28-0651
OS/VS2 TSO Command Processor Logic Volume II - EDIT	SY33-8548
OS/VS2 TSO Command Processor Logic Volume III - TEST	SY35-0004
OS/VS2 TSO Command Processor Logic Volume IV	SY28-0652

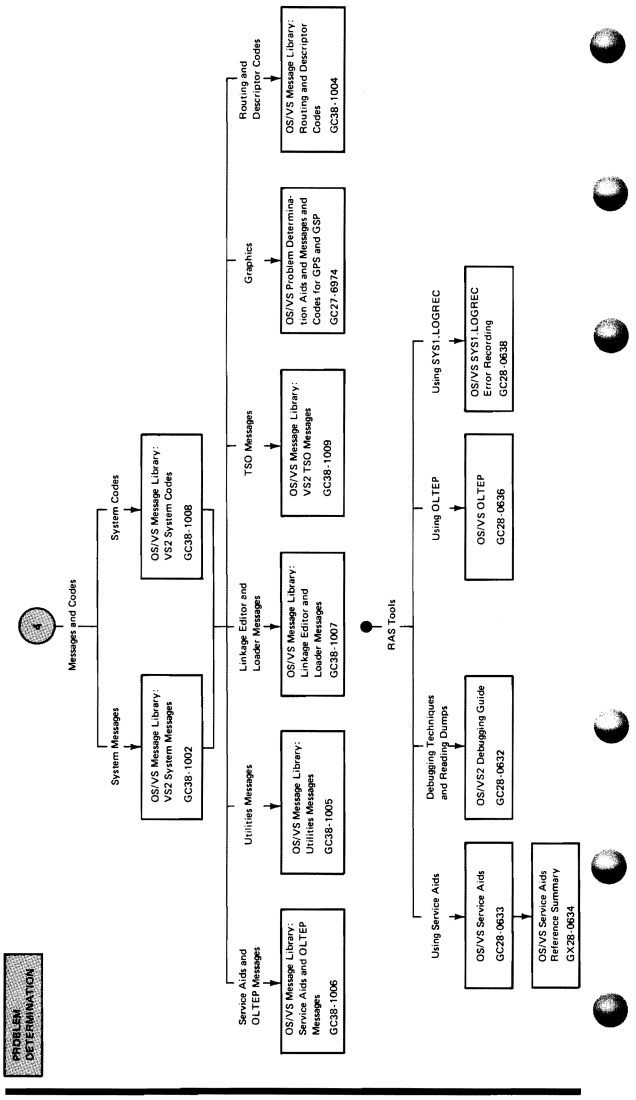

NEW USER
DATA PROCESSING OVERVIEW
Data Processing Concepts
Introduction to Data Processing GC20-1684
Operating System Concepts
IBM System/360 Operating System Introduction GC28-6534
Data Management Concepts
Introduction to Data Management SC20-8096
Teleprocessing Concepts
Introduction to Teleprocessing SC20-8095
OS USER
VIRTUAL STORAGE OVERVIEW
Virtual Storage Concepts
Introduction to Virtual Storage in System/370 GR20-4260
OS/VS Master Index GC28-0602
Summary of VS Components
IBM System/370 System Summary GA22-7001
VS Terminology
IBM Data Processing Glossary GC20-1699
IF YOU ARE INTERESTED IN:
THEN SEE:
PLANNING AND IMPLEMENTING A VS2 SYSTEM
1
OPERATING A VS2 SYSTEM
2
PROGRAMMING APPLICATIONS IN A VS2 SYSTEM
3
PROBLEM DETERMINATION AIDS IN VS2
4
MAINTAINING OR MODIFYING A VS2 SYSTEM
5

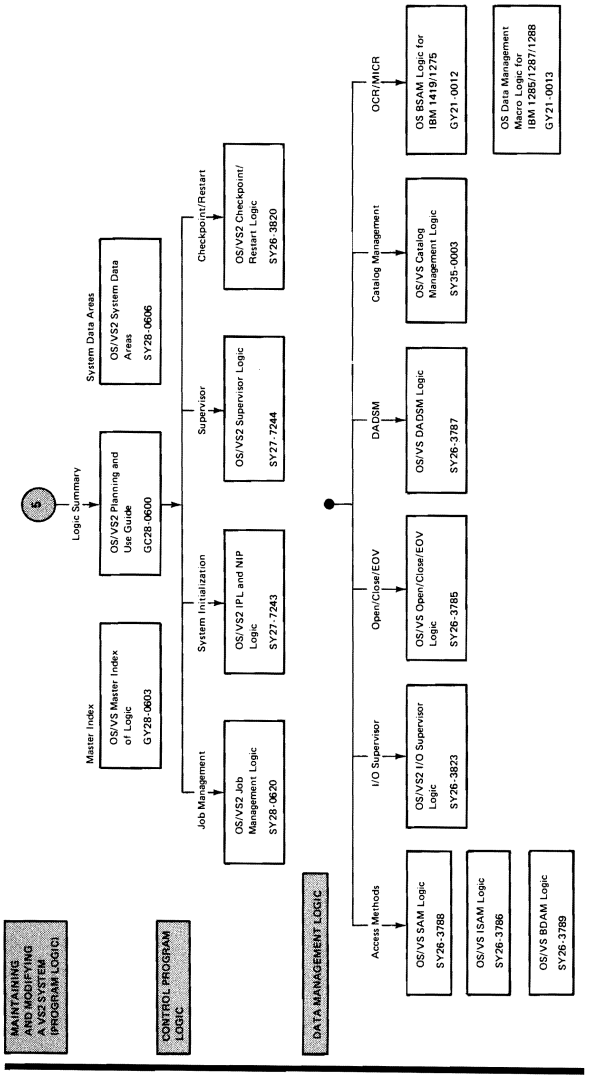

PLANNING AND IMPLEMENTING A VS2 SYSTEM
1
Overall Planning Information
OS/VS2 Planning and Use Guide GC28-0600
Evaluating a New VS2 Release
OS/VS2 Release 1 Guide GC28-0601
Estimating VS2 Storage Requirements
OS/VS2 Storage Estimates GC28-0604
Generating a VS2 System
OS/VS System Generation Introduction GC26-3790
OS/VS2 System Generation Reference GC26-3792
Implementing VS2 Facilities
Data Management
OS/VS Data Management for System Programmers GC28-0631
TSO
OS/VS2 TSO Guide GC28-0644
SMF
OS/VS System Management Facilities (SMF) GC35-0004
Checkpoint/Restart
OS/VS Checkpoint/Restart GC26-3784
OS/VS Tape Labels GC26-3795
OS/VS Virtual Storage Access Method (VSAM) Planning Guide GC26-3799
OPERATING A VS2 SYSTEM
2
Overall VS2 Operating Information
Operator's Library: OS/VS2 Reference GC38-0210
Operator's Library: S/370 Model 145 Procedures GC38-0015
Console Configurations
If you have a Display Console
If you have TCAM
If you have TSO
Operator's Library: OS/VS Console Configurations GC38-0120
Operator's Library: OS/VS2 Display Consoles GC38-0260
Operator's Library: OS/VS TCAM GC38-0305
Operator's Library: OS/VS2 TSO GC38-0220

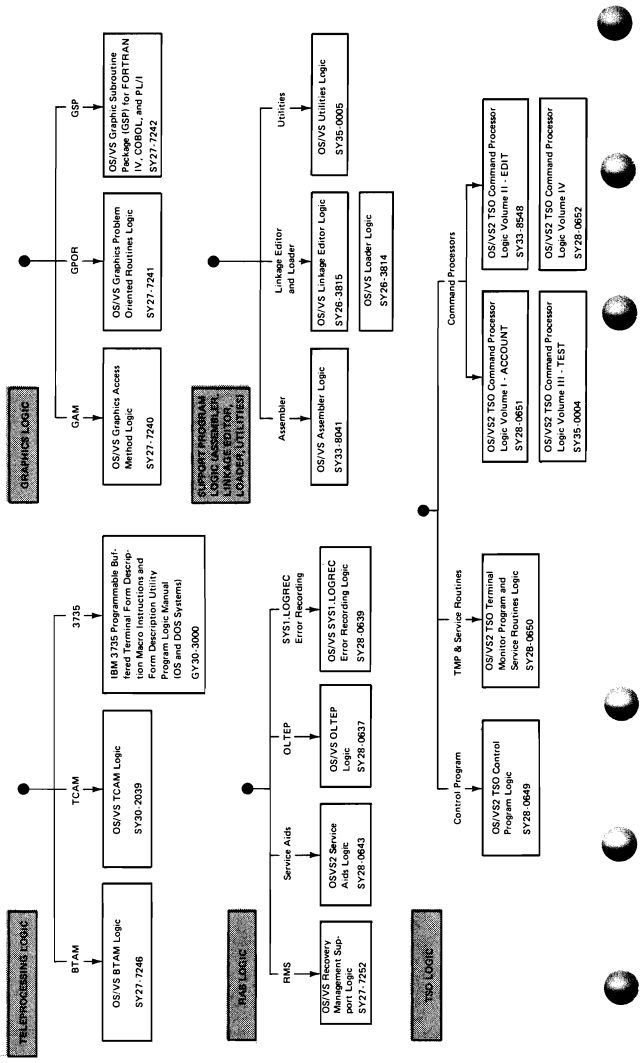

APPLICATION PROGRAMMING IN VS2
Control Program Services
Job Management Services
OS/VS JCL Services GC28-0617
OS/VS JCL Reference GC28-0618
Data Management Services
OS/VS Data Management Service Guide GC26-3783
OS/VS Data Management Macro Instructions GC26-3793
Supervisor Services
OS/VS Supervisor Services and Macro Instructions GC27-6979
Support Programs
Assembler
OS/VS and DOS/VS Assembler Language GC33-4010
OS/VS Assembler Programmer's Guide GC33-4021
Linkage Editor and Loader
OS/VS Linkage Editor and Loader GC26-3813
Utilities
OS/VS Utilities GC35-0005
If you are interested in coding:
Then see:
TP applications
A
Graphics applications
B
OCR/MICR applications
C
TSO applications
D

OS/VS2 Library Charts (con't)




PROBLEM DETERMINATION
Messages and Codes
System Messages
OS/VS Message Library: VS2 System Messages GC38-1002
System Codes
OS/VS Message Library: VS2 System Codes GC38-1008
Utilities Messages
OS/VS Message Library: Utilities Messages GC38-1005
Linkage Editor and Loader Messages
OS/VS Message Library: Linkage Editor and Loader Messages GC38-1007
TSO Messages
OS/VS Message Library: VS2 TSO Messages GC38-1009
Graphics
OS/VS Problem Determination Aids and Messages and Codes for GPS and GSP GC27-6974
Routing and Descriptor Codes
OS/VS Message Library: Routing and Descriptor Codes GC38-1004
Service Aids and OLTEP Messages
OS/VS Message Library: Service Aids and OLTEP Messages GC38-1006
RAS Tools
Using Service Aids
OS/VS Service Aids GC28-0633
OS/VS Service Aids Reference Summary GX28-0634
Debugging Techniques and Reading Dumps
OS/VS2 Debugging Guide GC28-0632
Using OLTEP
OS/VS OLTEP GC28-0636
Using SYS1.LOGREC
OS/VS SYS1.LOGREC Error Recording GC28-0638


MAINTAINING AND MODIFYING A VS2 SYSTEM (PROGRAM LOGIC)
CONTROL PROGRAM LOGIC
DATA MANAGEMENT LOGIC
Master Index
OS/VS Master Index of Logic GY28-0603
Logic Summary
OS/VS2 Planning and Use Guide GC28-0600
System Data Areas
OS/VS2 System Data Areas SY28-0606
Job Management
OS/VS2 Job Management Logic SY28-0620
System Initialization
OS/VS2 IPL and NIP Logic SY27-7243
Supervisor
OS/VS2 Supervisor Logic SY27-7244
Checkpoint/Restart
OS/VS2 Checkpoint/ Restart Logic SY26-3820
Access Methods
OS/VS SAM Logic SY26-3788
OS/VS ISAM Logic SY26-3786
OS/VS BDAM Logic SY26-3789
I/O Supervisor
OS/VS2 I/O Supervisor Logic SY26-3823
Open/Close/EOV
OS/VS Open/Close/EOV Logic SY26-3785
DADSM
OS/VS DADSM Logic SY26-3787
Catalog Management
OS/VS Catalog Management Logic SY35-0003
OCR/MICR
OS BSAM Logic for IBM 1419/1275 GY21-0012
OS Data Management Macro Logic for IBM 1285/1287/1288 GY21-0013


TELEPROCESSING LOGIC
BTAM
OS/VS BTAM Logic
SY27-7246
TCAM
OS/VS TCAM Logic
SY30-2039
3735
IBM 3735 Programmable Buffered Terminal Form Description Macro Instructions and Form Description Utility Program Logic Manual (OS and DOS Systems)
GY30-3000
RAS LOGIC
RMS
OS/VS Recovery Management Support Logic
SY27-7252
Service Aids
OS/VS2 Service Aids Logic
SY28-0643
OLTEP
OS/VS OLTEP Logic
SY28-0637
SYS1.LOGREC Error Recording
OS/VS SYS1.LOGREC Error Recording Logic
SY28-0639
TSO LOGIC
Control Program
OS/VS2 TSO Control Program Logic
SY28-0649
TMP & Service Routines
OS/VS2 TSO Terminal Monitor Program and Service Routines Logic
SY28-0650
Command Processors
OS/VS2 TSO Command Processor Logic Volume I - ACCOUNT
SY28-0651
OS/VS2 TSO Command Processor Logic Volume III - TEST
SY28-0004
OS/VS2 TSO Command Processor Logic Volume III - EDIT
SY33-8548
OS/VS2 TSO Command Processor Logic Volume IV
SY28-0652
GRAPHICS LOGIC
GAM
OS/VS Graphics Access Method Logic
SY27-7240
GPOR
OS/VS Graphics Problem Oriented Routines Logic
SY27-7241
GSP
OS/VS Graphic Subroutine Package (GSP) for FORTRAN IV, COBOL, and PL/I
SY27-7242
SUPPORT PROGRAM LOGIC (ASSEMBLER, LINKAGE EDITOR, LOADER, UTILITIES)
Assembler
OS/VS Assembler Logic
SY33-8041
Linkage Editor and Loader
OS/VS Linkage Editor Logic
SY26-3815
OS/VS Loader Logic
SY26-3814
Utilities
OS/VS Utilities Logic
SY35-0005

Index

For macros, instructions, and control statements,refer to these: BTAM, CRJE, data management, JCL, linkage editor, loader, OS/VS system assembler, supervisor, utilities, instructions-System/370, and TCAM.

OS/VS Programmer's Reference Digest Printed in U. S. A. GC24-5091-2

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)

OS/VS Programmer's Reference Digest
GC24-5091-2

READER'S COMMENT FORM

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 response, however. *For more direct handling of such request, 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.

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