Appendix A
Full List of Errors

ERR.STB Mnemonic	Decimal Value	Octal Value	Full Error Text
BADDIR	1	1	?Bad directory for device
BADNAM	2	2	?Illegal file name
INUSE	3	3	?Account or device in use
NOROOM	4	4	?No room for user on device
NOSUCH	5	5	?Can't find file or account
NODEVC	6	6	?Not a valid device
NOTCLS	7	7	?I/O channel already open
NOTAVL	8	10	?Device not available
NOTOPN	9	11	?I/O channel not open
PRVIOL	10	12	?Protection violation
EOF	11	13	?End of file on device
ABORT	12	14	?Fatal system I/O failure
DATERR	13	15	?Data error on device
HNGDEV	14	16	?Device hung or write locked
HNGTTY	15	17	?Keyboard wait exhausted
FIEXST	16	20	?Name or account now exists
DTOOOF	17	21	?Too many open files on unit
BADFUO	18	22	?Illegal SYS ( ) usage
INTLCK	19	23	?Disk block is interlocked
WRGPAK	20	24	?Pack ids don't match
NOTMNT	21	25	?Disk pack is not mounted
PAKLCK	22	26	?Disk pack is locked out
BADCLU	23	27	?Illegal cluster size
PRIVAT	24	30	?Disk pack is private
INTPAK	25	31	?Disk pack needs 'REBUILDing'
BADPAK	26	32	?Fatal disk pack mount error
DETKEY	27	33	?I/O to detached keyboard
CTRLCE	28	34	?Programmable ^C trap
SATTBD	29	35	?Corrupted file structure
DEVNFS	30	36	?Device not file-structured
BADCNT	31	37	?Illegal byte count for I/O

NOBUFS	32	40	?No buffer space available
B.4	33	41	?Odd address trap
B.10	34	42	?Reserved instruction trap
B.250	35	43	?Memory management violation
B.STAK	36	44	?SP stack overflow
B.SWAP	37	45	?Disk error during swap
B.PRTY	38	46	?Memory parity failure
MAGSEL	39	47	?Magtape select error
MAGRLE	40	50	?Magtape record length error
NRRTS	41	51	?Non-res run-time system
VCSERR	42	52	?Virtual buffer too large
VCAERR	43	53	?Virtual array not on disk
SIZERR	44	54	?Matrix or array too big
VCOERR	45	55	?Virtual array not yet open
BSERR	46	56	?Illegal I/O channel
LINERR	47	57	?Line too long
FLTERR	48	60	%Floating point error
EXPERR	49	61	%Argument too large in exp
FMTERR	50	62	%Data format error
FIXERR	51	63	%Integer error
BDNERR	52	64	%Illegal number
LOGERR	53	65	%Illegal argument in log
SQRERR	54	66	%Imaginary square roots
SUBERR	55	67	?Subscript out of range
MINVER	56	70	?Can't invert matrix
ODD	57	71	?Out of data
ONBAD	58	72	?ON statement out of range
NEDERR	59	73	?Not enough data in record
IOLERR	60	74	?Integer overflow, for loop
DIVBY0	61	75	%Division by 0
NORTS	62	76	?No run-time system
FIELDE	63	77	?Field overflows buffer
NORACS	64	100	?Not a random-access device
NOTMTA	65	101	?Illegal MAGTAPE() usage
ERRERR	66	102	?Missing special feature
BADSWT	67	103	?Illegal switch usage
	68	104	Unused error message
	69	105	Unused error message
	70	106	Unused error message
STMERR	71	107	?Statement not found
EXITTM	72	110	?RETURN without GOSUB
EXITNR	73	111	?FNEND without function call
UNDFNI	74	112	?Undefined function called
COSERR	75	113	?Illegal symbol
TLOPNV	76	114	?Illegal verb
TLNZSP	77	115	?Illegal expression
TLNOIT	78	116	?Illegal mode mixing
TLIFFE	79	117	?Illegal IF statement
TLCONI	80	120	?Illegal conditional clause
TLNOTF	81	121	?Illegal function name
TLQDUM	82	122	?Illegal dummy variable
TLMFND	83	123	?Illegal FN redefinition
TLRNNM	84	124	?Illegal line number(s)
MODERR	85	125	?Modifier error
	86	126	Unused error message
OUTCAS	87	127	?Expression too complicated

FUNERR	88	130	?Arguments don't match
TLTMAF	89	131	?Too many arguments
TLINCD	90	132	%Inconsistent function usage
CPNSDF	91	133	?Illegal DEF nesting
CPUPFR	92	134	?FOR without NEXT
CPUFNX	93	135	?NEXT without FOR
CPUPDF	94	136	?DEF without FNEND
CPUPED	95	137	?FNEND without DEF
TLJNKY	96	140	?Literal string needed
TLNOFN	97	141	?Too few arguments
SASYNE	98	142	?Syntax error
SAFNOS	99	143	?String is needed
SASNOI	100	144	?Number is needed
TLURTP	101	145	?Data type error
TLXDIM	102	146	?1 or 2 dimensions only
FUCORE	103	147	?Program lost-sorry
RESERR	104	150	?RESUME and no error
DIMED2	105	151	?Redimensioned array
TLIDIM	106	152	?Inconsistent subscript use
NOGOTO	107	153	?ON statement needs GOTO
EOSERR	108	154	?End of statement not seen
TLCNTD	109	155	?What?
TLPRNM	110	156	?Bad line number pair
EDBMCE	111	157	?Not enough available memory
EDEXON	112	160	?Execute only file
NRNERR	113	161	?Please use the RUN command
EDCONE	114	162	?Can't continue
EDARSV	115	163	?File exists–RENAME/REPLACE
PRERRS	116	164	?PRINT–USING format error
BADSWT	117	165	?Illegal switch usage
PRNER1	118	166	?Bad number in PRINT–USING
NONOIM	119	167	?Illegal in immediate mode
PRNER2	120	170	?PRINT–USING buffer overflow
BADERR	121	171	?Illegal statement
DISERR	122	172	?Illegal field variable
STPERR	123	173	Stop
DIMERR	124	174	?Matrix dimension error
NOMATH	125	175	?Wrong math package
XCDCOR	126	176	?Maximum memory exceeded
SCAERR	127	177	%SCALE factor interlock

Appendix B
Device Information

This appendix summarizes MODE and RECORD values and other useful information for:

- Disks
- Flexible diskettes
- Magnetic tape
- Line printers
- Terminals
- Pseudo keyboards

For your convenience, values are given in both decimal and octal, and FIRQB and XRB offsets are listed. All decimal values have a decimal point; values without a decimal point are in octal.

This appendix is a "quick reference." See the *RSTS/E Programming Manual* if you need more detail on any topic, except where otherwise noted.

B.1 Disks

This section summarizes disk MODE values and lists the device cluster size and total size for each disk that RSTS/E supports.

B.1.1 MODE Values

Tables B–1 and B–2 summarize disk MODE values for file-structured and non-file-structured access.

Table B–1: MODE Values for File-Structured Disk Access (FIRQB + FQMODE)

Decimal	Octal	Function
0.	0	Normal read/write
1.	1	Update
2.	2	Append
4.	4	Guarded update (4 + 1)
8.	10	Special extend (RSTS/E updates file's size and retrieval pointers during extend operations)
16.	20	Create contiguous file
32.	40	Create tentative file
64.	100	Create contiguous file conditionally
128.	200	No supersede
256.	400	Random data caching
512.	1000	Create file and place at beginning of directory (with 2000)
1024.	2000	Create file and place at end of directory
2048.	4000	Sequential data caching (with 400)
4096.	10000	Read normally regardless (privileged)
8192.	20000	Open file read-only
16384.	40000	Write UFD (privileged)

Table B–2: MODE Values for Non-File-Structured Disk Access (FIRQB + FQMODE)

Decimal	Octal	Function
0.	0	Access device clusters.
128.	200	Access disk blocks.
512.	1000	Read beyond last writable portion of disk; suppress error logging. (RSTS/E uses this mode in its online DSKINT program; it is not recommended for general use.)

B.1.2 Disk Device Sizes

Table B–3 lists the device cluster size and device size (in 512–byte blocks) for each disk that RSTS/E supports. All values are in decimal.

Table B–3: Disk Device Sizes

Disk	Device Cluster Size	Device Size
RX50	1	800
RF11	1	1024 times number of platters
RS03	1	1024
RS04	1	2048
RK05	1	4800
RK05F	1	4800 per unit; 2 units for each drive
RL01	1	10220
RL02	1	20460
RD51	1	21600
RK06	1	27104
RK07	1	53768
RC25	1	50902 per unit; 2 units per spindle
RP02	2	40000
RP03	2	80000
RM02	4	131648
RM03	4	131648
RP04	4	171796
RP05	4	171796
RA80	4	237208
RM80	4	242575
RP06	8	340664
RA60	8	400175
RM05	8	500352
RA81	16	888012

B.2 Flexible Diskettes

Tables B–4 and B–5 summarize MODE and RECORD values for flexible diskettes.

Table B–4: Flexible Diskette MODE Values (FIRQB+FQMODE)

Decimal	Octal	Function
0.	0	Block mode
16384.	40000	Sector mode

Table B–5: Flexible Diskette RECORD Values (XRB+XRBLK)

Decimal	Octal	Function
8192.	20000	Access logical record 0
16384.	40000	Write Deleted Data Mark
32767.+1.	100000	Perform this I/O operation in block mode

B.3 Magnetic Tape

This section summarizes MODE and CLUSTERSIZE values for magnetic tape.

B.3.1 File-Structured Processing

Tables B–6 and B–7 summarize MODE and CLUSTERSIZE values for file-structured magnetic tape.

Table B–6: MODE Values for File-Structured Magnetic Tape (FIRQB + FQMODE)

Decimal	Octal	Function
0.	0	Read file label at current tape position
2.	2	Do not rewind tape when searching for file
16.	20	Write over existing file
32.	40	Rewind tape before searching for file
64.	100	Rewind on CLOSE
128.	200	Open for append
512.	1000	Write new file label without searching
16384.	40000	Search for DOS–formatted file label
24576.	60000	Search for ANSI–formatted file label

Table B–7: CLUSTERSIZE Values for ANSI Magnetic Tape Files

Label Field Name	CLUSTERSIZE (FIRQB+FQCLUS)		Label Result
	Decimal	Octal	
Record format	0.	0	U = undefined.*
	16384.	40000	F = fixed-length.
	32767.+1.	100000	D = variable length.
	–16384.	140000	S = spanned.**
Record length (in bytes)	Betw 0. and 4095.	Betw 0 and 7777	For U, always 0. For F, fixed record length. For D, maximum record length. For S, not used.**
System Dependent (File characteristics)	0.	0	M = carriage control embedded.
	4096.	10000	A = FORTRAN carriage control.
	8192.	20000	(space) = implied carriage control. When printed, line feed precedes and carriage return follows each record.

* RSTS/E undefined record formats cannot be processed directly by other operating systems.

** RSTS/E does not support ANSI format S records.

B.3.2 Non-File-Structured Processing

MODE values in non-file-structured magnetic tape processing have the form:

MODE(FIRQB+FQMODE) = D + P + S

where:

D (density) = 12.(14) for 800 BPI
256.(400) for 1600 BPI, phase-encoded

P (parity) = 0.(0) for odd
1.(1) for even

S (stay) = 0.(0) to clear MODE value after CLOSE
8192.(20000) to retain MODE value after CLOSE

Note that DIGITAL recommends the use of odd parity. When you use even parity, you cannot write binary data. In addition, many operating systems and tape drives do not support even parity.

For information on magnetic tape special functions, the magnetic tape status word, and the file characteristics word, see the description of .SPEC for magnetic tape (Section 3.23.3).

B.4 Line Printers

This section summarizes line printer MODE and RECORD values.

Table B–8: Line Printer MODE Values (FIRQB+FQMODE)

Decimal	Octal	Function
1. – 127.	1 – 177	Sets form length to number of lines per page for software formatting (512., 1000) and automatic page skip (2048., 4000). (This is the QUE system program's LPFORM option.)
128.	200	Changes the character 0 (zero) to O (letter O).
256.	400	Truncates lines that are longer than unit was configured for instead of printing the rest of the line on the next physical line on the page.
512.	1000	Enables software formatting. Forms control characters are $\geq 200_8$.
1024.	2000	Translates lowercase characters to uppercase characters. Applies only to uppercase and lowercase line printers.
2048.	4000	Skips six lines (that is, skips over perforation line) at the bottom of each form.
4096.	10000	Moves paper to top of hardware form.
8192.	20000	Suppresses form feed on CLOSE.

Table B–9: Line Printer RECORD Values (XRB+XRMOD)

Decimal	Octal	Function
2.	2	Print over perforation (disables MODE 4000 for this output step).
4.	4	Do not return control to program until output is complete or an error occurs.
8.	10	Clear pending output buffers before buffering characters for the request.
8192.	20000	Return control to program if output stall is to occur. (See .WRITE directive for more information.)

B.5 Terminals

This section summarizes MODE and RECORD values for terminals. It also includes information about echo control mode and VT100 ANSI–compatible escape sequences.

B.5.1 Terminal MODE and RECORD Values

Tables B–10 through B–12 summarize terminal MODE values and RECORD values for terminal input and output.

Table B–10: Terminal MODE Values (FIRQB+FQMODE)

Decimal	Octal	Function
1.	1	Enable binary input from a terminal
2.	2	Reserved for TECO
4.	4	Suppress automatic CR/LF at right margin
8.	10	Enable echo control (turns off other modes and automatically enables MODE 4)
16.	20	Guard program against CTRL/C interruption and dial-up line hibernation
32.	40	Enable incoming XON/XOFF processing
64.	100	Reserved
128.	200	Enable use of RUBOUT as a delimiter on video terminals
256.	400	Set escape sequence mode

Table B–11: RECORD Values for Terminal Input (XRB+XRMOD)

Decimal and Octal Values	Function
8192. 20000	Perform conditional input (execute input request without waiting for data to be available).
32767.+1.+K 100000+K	Perform multiterminal service input from assigned keyboard number K.
32767.+1.+16384.+S 140000+S	Perform multiterminal service input from any assigned keyboard: S = 0 — Wait until input is available from any terminal. The error ?Data error on device may occur due to a race condition with CTRL/C. 1. < S < 255. 1 < S < 377 — Wait up to S seconds for input from any terminal and then return ?Data error on device if no input is available. S = 8192. S = 20000 — Request input immediately; return ?Data error on device if no input is pending.

Table B–12: RECORD Values for Terminal Output (XRB+XRMOD)

Decimal	Octal	Function
256.	400	Declare echo control field (use with MODE 10).
4096.	10000	Output binary data to terminal.
8192.	20000	Return control to program if output stall is to occur. (See .WRITE directive for more information.)
32767.+1.+K	100000+K	Perform multiterminal service output to assigned keyboard K.

B.5.2 Echo Control Mode

In echo control mode, the system strips the parity bit from all characters. All characters returned to your program have ASCII values in the range 1 to 177 (octal). The system does not pass synchronization or editing characters to your program. Delimiters are passed to your program but are never echoed. Table B–13 summarizes the echo control mode character set.

Table B–13: Echo Control Mode Character Set

Type of Character	ASCII Code (Octal)	Code Returned to User	Comments
Ignored	0		Used as filler for timing.
Delimiters	Private	?	Private delimiter.
	3	3	^C (CTRL/C).
	4	4	^D (CTRL/D).
	12	12	Line feed.
	14	14	Form feed.
	15	15,12	Carriage return (with line feed appended).
	32	32	^Z (CTRL/Z); generates error 11_{10}.
	33	33	If "NO ESC SEQ" is in effect and escape character is received, 33 is returned to user and is treated as a delimiter.
			If "ESC SEQ" is in effect, escape character triggers an escape sequence. The escape sequence is returned to user and the whole sequence is considered the delimiter.
	175	33 or 175	If "NO ESC" is in effect, 175 is translated to escape (33).
			If "ESC" is in effect, 175 is data.
	176	33 or 176	If "NO ESC" is in effect, 176 is translated to escape (33).
			If "ESC" is in effect, 176 is data.

Table B–13: Echo Control Mode Character Set (Cont.)

Type of Character	ASCII Code (Octal)	Code Returned to User	Comments
Editing	177	–	RUBOUT (DEL character); on video terminals, generates a backspace followed by the paint character and another backspace; on hard-copy terminals, echoes deleted characters between backslashes (\\).
	25	–	^U (CTRL/U); repeatedly simulates RUBOUT until no characters remain in field.
Data	40–137	40–137	Normal 64–character graphic set.
	140–176	100–136	If "NO LC INPUT," lowercase letters are translated to uppercase.
	140–176	140–176	If "LC INPUT," lowercase letters are returned to user.
Synchronization	21	–	XON (CTRL/Q). Resume suspended output (if the STALL characteristic is set).
	23	–	XOFF (CTRL/S). Suspend output (if the STALL characteristic is set).
Other	1,2,5,6, 7,10,11, 13,16–20, 22,24, 26–31, 34–37,	–	Echoed as BEL (code 7) but otherwise ignored.
	21,23	–	If the terminal is set "NO STALL," synchronization characters are also echoed as BEL (code 7) and ignored.

Declaring a Field in Echo Control Mode

Use .WRITE to declare a field. The .WRITE must include the value 256.(400) at XRB+XRMOD and the value N at XRB+XRBC. N, which must be between 1 and the size of the buffer, describes how many bytes in the buffer represent the field declaration:

N = 1 The byte contains field size and overflow handling information. The field size must be in the range 1. to 127.(1 – 177). Adding 128.(200) to the field size specifies keypunch overflow handling instead of normal overflow handling.

N = 2 The first byte contains field size and overflow handling as described for N = 1. The second byte contains the ASCII value of the paint character. If this byte is 0 or N = 1, a space is the paint character.

N > 2 The first (N–2) bytes contain a prompt to display on the terminal before the field. Byte (N–1) is the field size declaration as described for N = 1. The last byte is the paint character as described for N = 2.

B.5.3 Escape Sequences

Table B–14 summarizes the VT100 ANSI–compatible escape sequences that move the cursor, erase all or part of the screen, and control line size and VT100 character attributes (bold, underscore, blink, and reverse video). The table uses the symbols Pl, Pc, and Pn, where:

Pl Means line number.

Pc Means column number.

Pn Is a decimal parameter expressed as a string of ASCII digits. The parameter's meaning for each escape sequence is explained in the table. Separate multiple parameters with a semicolon (;). If you omit a parameter or specify 0, the terminal uses the default parameter value for that escape sequence.

Be sure to include the left square bracket ([) in the escape sequence prefix where shown in the table. Escape sequences cannot contain embedded spaces. Refer to the *VT100 User Guide* for a complete description of VT100 escape sequences.

Table B–14: VT100 ANSI–compatible Escape Sequences for Screen Control

Escape Sequence	Description
Cursor Movement	
ESC[PnA	Moves the cursor up n lines without affecting the column position. The parameter Pn specifies the number of lines. The default value is one line.
ESC[PnB	Moves the cursor down n lines without affecting the column position. The parameter Pn specifies the number of lines. The default value is one line.
ESC[PnC	Moves the cursor forward (right) n columns without affecting the line position. The parameter Pn specifies the number of columns. The default value is one column.
ESC[PnD	Moves the cursor backward (left) n columns without affecting the line position. The parameter Pn specifies the number of columns. The default value is one column.

(continued on next page)

Table B–14: VT100 ANSI–compatible Escape Sequences for Screen Control (Cont.)

Escape Sequence	Description
Cursor Movement	
ESC[Pl;PcH	Direct cursor address – moves the cursor to the specified line and column position. If you do not specify a line or column position, the cursor moves to the home position, which is the top left corner of the screen.
ESCD	Index – moves the cursor to the current column position on the next line.
ESCM	Reverse index – moves the cursor to the current column position on the preceding line.
ESCE	Moves the cursor to the first column position on the next line.
Erasing	
ESC[K or ESC[0K	Erases from the current cursor position to the end of the line.
ESC[1K	Erases from the beginning of the current line to the cursor.
ESC[2K	Erases the entire line containing the cursor.
ESC[J or ESC[0J	Erases from the current cursor position to the end of the screen.
ESC[1J	Erases from the beginning of the screen to the current cursor position.
ESC[2J	Erases the entire screen.
Line Size (Double Height and Double Width)	
ESC#3	Changes the current line to the top half of a double-height double-width line.
ESC#4	Changes the current line to the bottom half of a double-height double-width line.
ESC#6	Changes the current line to a double-width single-height line.
To display double-height characters, use the ESC#3 and ESC#4 sequences as a pair on adjacent lines and send the same characters to both lines. The use of double-width characters reduces the number of characters per line by half.	
Character Attributes (require Advanced Video Option)	
ESC[Pn;Pn;Pn;...;m	Turns bold, underscore, blink, and reverse video attributes on and off. Pn can have the following values: 0 or none All attributes off 1 Bold on 4 Underscore on 5 Blink on 7 Reverse video on The terminal executes the parameters in order and ignores any other parameter values. Unlike line size commands, which affect only the current line, the character attributes affect the entire screen. Remember to turn them off before ending your program.

B.6 Pseudo Keyboards

This section summarizes MODE and RECORD values for pseudo keyboards. It also lists errors your program can receive on a pseudo keyboard output request.

Table B–15: Pseudo Keyboard MODE Values (FIRQB+FQMODE)

Decimal	Octal	Function
0.	0	System kills controlled job when the pseudo keyboard is closed.
1.	1	System detaches controlled job when the pseudo keyboard is closed.

Table B–16: RECORD Option Bit Values for Pseudo Keyboard Output (XRB+XRMOD)

Bit	Value		Result
	Decimal	Octal	
0	1.	1	If set, the system does not check job status before sending data to the pseudo keyboard.
1	2.	2	If set, the system tests whether pseudo keyboard is waiting for a system command (CTRL/C state) or is waiting for program input (KB wait state).
2	4.	4	If set, the system does not send data to the pseudo keyboard but instead returns control to the controlling job.
3	8.	10	If set, and there are no small buffers for keyboard input, the system waits until small buffers are available. However, your program receives the NOROOM error if the output buffer chain is full.
4	16.	20	If set, the system kills the job currently running at the pseudo keyboard.

Table B–17: Possible Errors on Pseudo Keyboard Output Request

Error	Meaning
INUSE	Job at pseudo keyboard is not ready for input
NOROOM	No buffer space is available
NOSUCH	No controlled job exists at pseudo keyboard
CTRLCE	Job at pseudo keyboard is not in CTRL/C state

Appendix C
Supplementary RSX Directives for Resident Libraries

The RSX emulator directives that deal with resident libraries (ATRG$, DTRG$, CRAW$, ELAW$, MAP$, and UMAP$) use 8–word areas to pass and receive data to and from the emulator. These areas, called the resident library definition block (RDB) and window definition block (WDB), can be defined or defined and filled using supplementary directives included in the RSXMAC.SML file.

Note that the expansions for these directives are the same as in the RSX–11M environment, where their use is more extensive. Only the arguments relevant to RSTS/E are described here.

C.1 RDB Directives

Two directives are available for use with resident library definition blocks (RDBs): RDBDF$ and RDBBK$.

The RDBDF$ directive simply assigns literal values to the offsets and status bit mnemonics shown in Chapter 5 for the RDB areas for the ATRG$ and DTRG$ directives. You can use these mnemonics to reference offsets and bit values in an RDB you have allocated space for in your program.

The RDBBK$ directive defines these offsets and, in addition, generates code to allocate space for the RDB and fills it with values you specify in the call.

The form for the relevant arguments in the ATRG$ call is:

RDBBK$,,*libnam*,,<RS.WRT> (for read/write access)

or

RDBBK$,,*libnam*,,<RS.RED> (for read-only access)

where *libnam* is the name of the resident library to be attached.

For example, the RDBBK$ call below expands to the instructions that follow.

```
RDBBK$   ,,DATLIB,,<RS.WRT>
```

Expansion

```
.WORD    0
.WORD    0
.RAD50   /DATLIB/
.WORD    0
.WORD    0
.WORD    RS.WRT
.WORD    0
```

(RS.WRT is assigned to a literal value of 2, so bit 1 is set in the seventh word of the RDB, requesting read/write access.)

In addition, all the offsets necessary to reference the data returned in the RDB by ATRG$ or DTRG$ are generated. For example, you can use the mnemonic RS.UNM to test bit 14 of the word at offset R.GSTS in the RDB after execution of a DTRG$ directive. RS.UNM is assigned to a literal value of 40000_8 by the RDBBK$ directive.

C.2 WDB Directives

Two directives are available for use with window definition blocks (WDBs): WDBDF$ and WDBBK$.

The WDBDF$ directive simply assigns literal values to the offsets and status bit mnemonics shown in Chapter 5 for the WDB areas for the CRAW$, ELAW$, MAP$, and UMAP$ directives. You can use these mnemonics to reference offsets and bit values in a WDB you have allocated space for in your program.

The WDBBK$ directive defines these offsets and, in addition, generates code to allocate space for the WDB and fills it with values you specify in the call.

The form for the relevant arguments in a CRAW$ call is:

WDBBK$ *apr,siz,rid,off,len,<bit1![bit2!bit3]>*

apr = The base apr.

siz = The size of the window.

rid = The resident library ID.

off = The offset into the library, in 32–word blocks.

len = The length to be mapped.

bit... = Mnemonic values for bit settings, separated by exclamation points. Relevant mnemonics for CRAW$ are:

WS.MAP = Window is to be mapped.
WS.WRT = Map with read/write access.
WS.RED = Map with read-only access.

For example, the WDBBK$ call below expands to the instructions that follow.

```
WDBBK$   7,128.,0,0,0,<WS.MAP!WS.RED>
```

Expansion

```
.BYTE   0,7
.WORD   0
.WORD   128.
.WORD   0
.WORD   0
.WORD   0
.WORD   WS.MAP!WS.RED
```

This WDB, when used in a CRAW$ directive, would create a window 4K words long (128_{10} 32–word blocks) in APR 7. The call also specifies an offset and map length of zero and read-only mapping. Note that to use this WDB, you have to supply the resident library ID, which you get from the RDB returned by an ATRG$ call. To fill in the library ID, move the word at offset R.GID in the RDB to offset W.NRID in the WDB.

The WDBBK$ directive also defines the offsets and bit settings referred to in the discussion of ELAW$, MAP$, and UMAP$.

Index

C

E

F

G

H

I

J

K

L

M

N

Q

R

S

T

U

V

W

X

Z

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA and Puerto Rico call **800–258–1710**

In Canada call **800–267–6146**

In New Hampshire, Alaska or Hawaii call **603–884–6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

*Any prepaid order from Puerto Rico must be placed with the Local Digital Subsidiary:
809–754–7575

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. _____

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number. _____

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

Do Not Tear - Fold Here and Tape

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN: Commercial Engineering Publications MK01–2/E06
RSTS/E Documentation
DIGITAL EQUIPMENT CORPORATION
CONTINENTAL BOULEVARD
MERRIMACK, N.H. 03054

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line