VMS

VMS Programming Master Index

Order Number: AA–LA56B–TE

VMS Programming Master Index

Order Number: AA–LA56B–TE

June 1990

This index includes entries for all manuals in the VMS Programming Subkit.

Revision/Update Information: This document supersedes the *VMS Programming Master Index*, Version 5.0.

Software Version: VMS Version 5.4

digital equipment corporation
maynard, massachusetts

June 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

CDA
DDIF
DEC
DECdtm
DECnet
DECUS
DECwindows
DECwriter
DEQNA
Desktop–VMS
DIGITAL
GIGI
HSC
LiveLink
LN03
MASSBUS
MicroVAX
PrintServer 40
Q-bus
ReGIS
ULTRIX
UNIBUS
VAX
VAXcluster
VAX RMS
VAXserver
VAXstation
VMS
VT
XUI
digital™

The following are third-party trademarks:

Adobe, Display PostScript, and PostScript are registered trademarks of Adobe Systems Incorporated.

X Window System, Version 10 and its derivations (X, X10, X Version 10, X Window System) are trademarks of the Massachusetts Institute of Technology.

X Window System, Version 11 and its derivations (X, X11, X Version 11, X Window System) are trademarks of the Massachusetts Institute of Technology.

ZK4647

Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by Digital. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use Digital-supported devices, such as the LN03 laser printer and PostScript printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.

1 Introduction

The *VMS Programming Master Index* is an edited compilation of the individual indexes for books in the Version 5.4 VMS Programming Subkit.

Each main entry in the *VMS Programming Master Index* is followed by an abbreviated book title and a reference to the page where the topic appears in that book. For example, an entry in the master index might appear in the following way:

FAB$B_BKS field • File Applications, 3-24, 4-28
7-19, 7-20; File Def Language, FDL-18;
RMS, 5-3

This entry indicates that you can find information about the FAB$B_BKS field in the following places:

- Pages 3–24, 4–28, 7–19, and 7–20 in the *Guide to VMS File Applications*
- Page FDL–18 in the *VMS File Definition Language Facility Manual*
- Page 5-3 in the *VMS Record Management Services Manual*

The subentries, if any, contain more specific information about the topic. For example, some subentries listed under *FAB$B_BKS field* are as follows:

considerations for calculating
default logic
limitation for RMS
performance considerations

The following table lists the abbreviated names used in the *VMS Programming Master Index* to reference each manual, the volume number of the binder that contains the manual, and the full manual title that corresponds to the abbreviated name:

Index Abbreviation	Volume	Title of Manual
Analyze/RMS_File	6A	*VMS Analyze/RMS_File Utility Manual*
Command Def	2B	*VMS Command Definition Utility Manual*
Convert	6A	*VMS Convert and Convert/Reclaim Utility Manual*
Debugger	2A	*VMS Debugger Manual*
Delta/XDelta	7B	*VMS Delta/XDelta Utility Manual*
Device Support (A)	8A	*VMS Device Support Manual*
Device Support (B)	8B	*VMS Device Support Reference Manual*
File Applications	6A	*Guide to VMS File Applications*
File Def Language	6A	*VMS File Definition Language Facility Manual*
I/O User's I	7A	*VMS I/O User's Reference Manual: Part I*
I/O User's II	7A	*VMS I/O User's Reference Manual: Part II*
Librarian	2B	*VMS Librarian Utility Manual*

Index Abbreviation	Volume	Title of Manual
Linker	2B	*VMS Linker Utility Manual*
MACRO	9	*VAX MACRO and Instruction Set Reference Manual*
Message	2B	*VMS Message Utility Manual*
Modular Procedures	1	*Guide to Creating VMS Modular Procedures*
National Char Set	6A	*VMS National Character Set Utility Manual*
Patch	2B	*VMS Patch Utility Manual*
Programming Resources	1	*Guide to VMS Programming Resources*
RMS	6B	*VMS Record Management Services Manual*
Routines Intro	3	*Introduction to VMS System Routines*
RTL DECtalk	5A	*VMS RTL DECtalk (DTK$) Manual*
RTL General Purpose	5A	*VMS RTL General Purpose (OTS$) Manual*
RTL Intro	5A	*Introduction to the VMS Run-Time Library*
RTL Library	5B	*VMS RTL Library (LIB$) Manual*
RTL Math	5A	*VMS RTL Mathematics (MTH$) Manual*
RTL Parallel Processing	5A	*VMS RTL Parallel Processing (PPL$) Manual*
RTL Screen Management	5C	*VMS RTL Screen Management (SMG$) Manual*
RTL String Manipulation	5C	*VMS RTL String Manipulation (STR$) Manual*
SUMSLP	2B	*VMS SUMSLP Utility Manual*
System Dump Analyzer	7B	*VMS System Dump Analyzer Utility Manual*
System Services Ref	4B	*VMS System Services Reference Manual*
System Services Intro	4A	*Introduction to VMS System Services*
Utility Routines	3	*VMS Utility Routines Manual*
VAXTPU	10	*VAX Text Processing Utility Manual*

Note: **System services, RMS services, and Run-Time Library routines are indexed by facility prefix. All system services and RMS services are now indexed under the prefix "SYS" (for example, the Create service is indexed under SYS$CREATE).**

Run-Time Library routines are organized into the following seven facilities:

DECtalk (DTK$)
General purpose (OTS$)
Library (LIB$)
Math (MTH$)
Parallel processing (PPL$)
Screen management (SMG$)
String manipulation (STR$)

To reference Run-Time Library routines in this index, look under the corresponding facility prefix (for example, the library routine $FIND_FILE is indexed under LIB$FIND_FILE).

Index

A

@ command • *VAXTPU*, 4–32
Abnormal termination of subordinate
  notification of • *RTL Parallel Processing*, 2–3
Abort
  kernel stack not valid • *MACRO*, E–10
  resulting from exceeding virtual address space • *VAXTPU*, 5–1
Abort function • *Debugger*, 2–7, 10–10, CD–38, CD–127, CD–209
  with DECwindows • *Debugger*, 1–21
Aborting an I/O request
  See I/O request
/ABORT qualifier • *Debugger*, CD–182
ABORT statement • *VAXTPU*, 3–26, 3–33, 7–16
Absolute expression • *MACRO*, 3–9
Absolute mode • *MACRO*, 5–14
  assembling relative mode as • *MACRO*, 6–22
/ABSOLUTE qualifier • *Patch*, PAT–27, PAT–30
Absolute queue • *MACRO*, 9–82
  manipulating • *MACRO*, 9–85
Absolute time • *Programming Resources*, 3–23; *System Services Intro*, 9–2
  as input to SYS$BINTIM • *System Services*, SYS–28
  converting to numeric • *System Services*, SYS–366
  in system format • *System Services Intro*, 9–3
Absolute value • *RTL Math*, 1–4
  of complex number • *RTL Math*, MTH–23
/AC
  See /ASCIC qualifier
ACB$V_QUOTA • *Device Support (B)*, 3–7, 3–10
ACB (AST control block) • *Device Support (A)*, 4–20; *Device Support (B)*, 1–38, 1–86, 3–2, 3–4
  contents • *Device Support (B)*, 3–6
ACBB (Add Compare and Branch Byte) instruction • *MACRO*, 9–44
ACBD (Add Compare and Branch D_floating) instruction • *MACRO*, 9–44
ACBF (Add Compare and Branch F_floating) instruction • *MACRO*, 9–44
ACBG (Add Compare and Branch G_floating) instruction • *MACRO*, 9–44
ACBH (Add Compare and Branch H_floating) instruction • *MACRO*, 9–44
ACBL (Add Compare and Branch Long) instruction • *MACRO*, 9–44
ACBW (Add Compare and Branch Word) instruction • *MACRO*, 9–44
Access
  See also Random access
  file • *Routines Intro*, A–5t
  logical I/O • *System Services Intro*, 7–10
  modes • *File Applications*, 1–2; *RMS*, 1–1
  page • *Routines Intro*, A–10t
  physical I/O • *System Services Intro*, 7–8
  random • *File Applications*, 1–2, 3–13
  run-time options • *RMS*, 1–2
  run-time options under VMS RMS • *RMS*, 1–2
  sequential • *File Applications*, 1–2, 3–13
  shared • *File Applications*, 10–30
    in a VAXcluster • *File Applications*, 3–28
  system object • *Routines Intro*, A–11t
  to process-permanent files • *File Applications*, 6–20
ACCESS attribute • *File Def Language*, FDL–2
Access category • *File Applications*, 4–21
Access control list
  See ACL
Access control list (ACL)
  See also ACL Editor routine
  editing • *Utility Routines*, ACL–3
  manipulating • *Utility Routines*, ACL–1
Access control list buffer field
  See XAB$L_ACLBUF field
Access control list buffer size field
  See XAB$W_ACLSIZ field
Access control list context field
  See XAB$L_ACLCTX field
Access Control List Editor routine
  See ACL Editor routine
Access control list error status field
  See XAB$L_ACLSTS field
Access control list length field
  See XAB$W_ACLLEN field
Access entry • *Routines Intro*, 1–9; *System Services Intro*, 1–7
Accessibility of memory
  See Buffer
Access method • *Routines Intro*, 1–9; *System Services Intro*, 1–7
Access mode • *System Services Intro*, 2–2

Access mode (Cont.)
  See also Record access mode
  changing to executive • *System Services*, SYS–64
  changing to kernel • *System Services*, SYS–66
  effect on AST delivery • *System Services Intro*, 5–6
  processor • *Routines Intro*, A–2
  specifying • *System Services Intro*, 2–2
  types of • *System Services Intro*, 2–2
  vector • *MACRO*, 10–20, 10–43, 10–49
  with AST • *System Services Intro*, 5–2
  with logical names • *System Services Intro*, 6–7
ACCESS primary
  secondary attributes • *File Applications*, 7–3
Access rights block
  See ARB
Access specification
  list of mask values • *RMS*, 14–6
Access types • *System Services Intro*, 1–8
Access violation • *System Dump Analyzer*, SDA–16, SDA–19
  See SS$_ACCVIO
access_bit_names data type • *Routines Intro*, A–2
access_mode data type • *Routines Intro*, A–2
Accounting message
  format of • *System Services*, SYS–96
ACE (access control list entry)
  alarm • *System Services Intro*, 3–19
  application • *System Services Intro*, 3–20
  creating • *System Services Intro*, 3–18, 3–25
  default protection • *System Services Intro*, 3–21
  identifier • *System Services Intro*, 3–23
  maintaining • *System Services Intro*, 3–18, 3–25
  translating • *System Services Intro*, 3–18, 3–24
  types of • *System Services Intro*, 3–18
  VMS RMS limitation • *RMS*, 14–2
ACF (configuration control block) • *Device Support (B)*, 1–2 to 1–4
ACL (access control list) • *Programming Resources*, 6–1; *System Services Intro*, 3–2
  as protection basis • *File Applications*, 4–21
  compared with UIC protection • *File Applications*, 1–10
  conversion methods • *RMS*, 14–2
  use with VMS RMS control block • *RMS*, 14–2
ACL (access rights list) • *Device Support (B)*, 1–45
ACLEDIT$EDIT routine • *Utility Routines*, ACL–3
ACL Editor routine
  example of use in BLISS program • *Utility Routines*, ACL–1
  introduction • *Utility Routines*, ACL–1
  options available • *Utility Routines*, ACL–3
ACLEDT$SECTION logical name
  defined • *Utility Routines*, ACL–5
ACP (ancillary control process) • *I/O User's I*, 1–1; *System Dump Analyzer*, SDA–99; *Device Support (B)*, 1–12, 1–39, 1–40, 1–74
  See also XQP
  class • *Device Support (B)*, 1–28
  default • *Device Support (B)*, 1–28
ACP control function • *I/O User's I*, 1–30
  disk quotas • *I/O User's I*, 1–33
  magnetic tape positioning • *I/O User's I*, 1–31
  miscellaneous disk • *I/O User's I*, 1–32
  quota file transfer block • *I/O User's I*, 1–33
ACP function • *I/O User's I*, 1–2
  arguments • *I/O User's I*, 1–2
  attributes • *I/O User's I*, 1–16 to 1–18
  IO$_ACCESS • *I/O User's I*, 1–7, 1–10, 1–14, 1–26
  IO$_ACPCONTROL • *I/O User's I*, 1–7, 1–30
  IO$_CREATE • *I/O User's I*, 1–10, 1–11, 1–14, 1–22
  IO$_DEACCESS • *I/O User's I*, 1–13, 1–14, 1–28
  IO$_DELETE • *I/O User's I*, 1–7, 1–29
  IO$_MODIFY • *I/O User's I*, 1–7, 1–11, 1–13, 1–14, 1–28
  IO$_MOUNT • *I/O User's I*, 1–30
  major • *I/O User's I*, 1–22
ACP-QIO interface • *I/O User's I*, 1–1
  access file function • *I/O User's I*, 1–26
  access subfunction • *I/O User's I*, 1–10
  ACP control function • *I/O User's I*, 1–30
  ANSI standard • *I/O User's I*, 1–2, 1–32
  arguments • *I/O User's I*, 1–2
    disk quota • *I/O User's I*, 1–33
  attribute control block • *I/O User's I*, 1–14
  attributes • *I/O User's I*, 1–16 to 1–18
  attributes statistics block • *I/O User's I*, 1–21
  BLISS-32 programming • *I/O User's I*, 1–2
  create file function • *I/O User's I*, 1–22
    disk • *I/O User's I*, 1–24
    magnetic tape • *I/O User's I*, 1–26
  deaccess file function • *I/O User's I*, 1–28
  delete file function • *I/O User's I*, 1–29
  description • *I/O User's I*, 1–1
  directory entries • *I/O User's I*, 1–9, 1–26
  FIB (file information block) • *I/O User's I*, 1–3
    See also FIB (file information block)
  file characteristics • *I/O User's I*, 1–18
  function codes • *I/O User's I*, A–1
  function modifiers • *I/O User's I*, 1–2

ACP-QIO interface
function modifiers (Cont.)
IO$M_ACCESS • *I/O User's I*, 1–10, 1–23, 1–25, 1–26
IO$M_CREATE • *I/O User's I*, 1–23, 1–24, 1–25, 1–26
IO$M_DELETE • *I/O User's I*, 1–23, 1–24, 1–30
IO$M_DMOUNT • *I/O User's I*, 1–31, 1–32
I/O operations • *I/O User's I*, 1–1
I/O status block • *I/O User's I*, 1–35
record attributes area • *I/O User's I*, 1–19
values • *I/O User's I*, 1–20
serious exception (EOT) • *I/O User's I*, 1–23, 1–27, 1–32
status returns • *I/O User's I*, A–1
VAX MACRO programming • *I/O User's I*, 1–1
XQP (extended QIO processor) • *I/O User's I*, 1–1
ACP queue block
See AQB
ACP subfunction • *I/O User's I*, 1–7
access • *I/O User's I*, 1–10
directory lookup • *I/O User's I*, 1–7
extend • *I/O User's I*, 1–11, 1–35
read/write attributes • *I/O User's I*, 1–14
truncate • *I/O User's I*, 1–13
ACP_MULTIPLE parameter • *Device Support (B)*, 1–28
Action routine
See FDT routine
designating for client messages • *VAXTPU*, 7–357
detached cursor
defining • *VAXTPU*, 7–367
fetching • *VAXTPU*, 7–197
for handling client messages
fetching • *VAXTPU*, 7–197
Action routine bit mask • *Device Support (A)*, 4–12
/ACTIVATING qualifier • *Debugger*, 10–14, CD–17, CD–30, CD–131, CD–188
Activation
predefined tracepoint, multiprocess program • *Debugger*, 10–14
Active area • *VAXTPU*, 7–350
determining location of • *VAXTPU*, 7–196
Active editing point • *VAXTPU*, 2–4
/ACTIVE qualifier • *Debugger*, CD–182
%ACTIVE_TASK • *Debugger*, D–10
Actual offset value
avoiding use of • *RMS*, 2–4
/AD
See /ASCID qualifier
Ada
See VAX Ada
special considerations • *RTL Parallel Processing*, 5–6
Ada data type declaration • *Routines Intro*, A–13
%ADAEXC_NAME • *Debugger*, 9–16, D–10
Ada implementation table • *Routines Intro*, A–13
Adapter
See I/O adapter
Adapter control block
See ADP
Adapter dispatch table • *Device Support (A)*, 14–27, 14–30; *Device Support (B)*, 1–6, 1–7
address • *Device Support (B)*, 1–7
examining • *Device Support (A)*, 13–9
ADAWI (Add Aligned Word Interlocked) instruction • *MACRO*, 9–7
ADDB2 (Add Byte 2 Operand) instruction • *MACRO*, 9–8
ADDB3 (Add Byte 3 Operand) instruction • *MACRO*, 9–8
ADD command • *File Applications*, 10–28; *File Def Language*, FDL–59
ADDD2 (Add D_floating 2 Operand) instruction • *MACRO*, 9–107
ADDD3 (Add D_floating 3 Operand) instruction • *MACRO*, 9–107
ADDF2 (Add F_floating 2 Operand) instruction • *MACRO*, 9–107
ADDF3 (Add F_floating 3 Operand) instruction • *MACRO*, 9–107
ADDG2 (ADD G_floating 2 Operand) instruction • *MACRO*, 9–107
ADDG3 (ADD G_floating 3 Operand) instruction • *MACRO*, 9–107
ADDH2 (ADD H_floating 2 Operand) instruction • *MACRO*, 9–107
ADDH3 (ADD H_floating 3 Operand) instruction • *MACRO*, 9–107
Addition
of decimal strings • *RTL String Manipulation*, STR–3
quadword times • *RTL Library*, LIB–5
two's complement • *RTL Library*, LIB–7
Additional routines
list of • *RTL Math*, 1–4 to 1–9
Addition operator (+) • *System Dump Analyzer*, SDA–12
ADDL2 (Add Long 2 Operand) instruction • *MACRO*, 9–8
ADDL3 (Add Long 3 Operand) instruction • *MACRO*, 9–8

- ADDP4 (Add Packed 4 Operand) instruction • *MACRO*, 9–148
- ADDP6 (Add Packed 6 Operand) instruction • *MACRO*, 9–148
- Address
  - access type • *MACRO*, 8–17
  - definition of • *Routines Intro*, 2–3
  - depositing into • *Debugger*, 4–25
    - with DECwindows • *Debugger*, 1–26
  - examining • *Debugger*, 4–14; *System Dump Analyzer*, SDA–51
    - with DECwindows • *Debugger*, 1–26
  - instructions • *MACRO*, 9–33
  - obtaining • *Debugger*, 3–13, 4–13
    - with DECwindows • *Debugger*, 1–26
  - on VAXBI • *Device Support (A)*, 12–9
  - on XMI • *Device Support (A)*, 12–11
  - specifying breakpoint • *Debugger*, 3–12
  - storage directive (.ADDRESS) • *MACRO*, 6–4
  - symbolizing • *Debugger*, 4–14
    - with DECwindows • *Debugger*, 1–26
  - translation vector • *MACRO*, 10–47
  - virtual • *MACRO*, 8–1
  - virtual memory • *Programming Resources*, 5–10
- address data type • *Routines Intro*, A–2t
- .ADDRESS directive • *MACRO*, 6–4
  - count of, in map • *Linker*, 5–2, 5–5
  - effect on position independence • *Linker*, 4–5
  - effect on shareability • *Linker*, 1–10, 4–4
  - guidelines for use of • *Linker*, 4–5
  - image activator's processing of • *Linker*, 6–20
  - linker's processing of • *Linker*, 6–20
  - relation to fix-up image section • *Linker*, 6–20
- Address expression
  - See also Address
  - code • *Debugger*, 4–20
    - with DECwindows • *Debugger*, 1–23
  - compared to language expression • *Debugger*, 4–8
    - with DECwindows • *Debugger*, 1–23
  - composite, vector • *Debugger*, 11–17
  - current entity • *Debugger*, 4–8, D–5
    - with DECwindows • *Debugger*, 1–8
  - DEPOSIT command • *Debugger*, 4–3, CD–61
  - EVALUATE/ADDRESS command • *Debugger*, 3–13, 4–13, CD–83
  - EXAMINE command • *Debugger*, 4–2, CD–85
  - EXAMINE/SOURCE command • *Debugger*, 6–4
  - logical predecessor • *Debugger*, 4–8, D–5
    - with DECwindows • *Debugger*, 1–8
  - logical successor • *Debugger*, 4–8, D–5
    - with DECwindows • *Debugger*, 1–8

Address expression (Cont.)

  - selecting from DECwindows window • *Debugger*, 1–23
  - SET BREAK command • *Debugger*, 3–9, CD–130
  - SET TRACE command • *Debugger*, 3–10, CD–187
  - SET WATCH command • *Debugger*, 3–17, CD–200
  - symbolic • *Debugger*, 4–4
    - with DECwindows • *Debugger*, 1–23
  - SYMBOLIZE command • *Debugger*, 4–14, CD–271
  - type of • *Debugger*, 4–4
- Addressing mode • *MACRO*, 5–1
  - absolute • *MACRO*, 5–14, 6–22
  - autodecrement • *MACRO*, 5–7
  - autoincrement • *MACRO*, 5–5
  - autoincrement deferred • *MACRO*, 5–6
  - branch • *MACRO*, 5–18
  - determining • *MACRO*, 6–68
  - displacement • *MACRO*, 5–8
  - displacement deferred • *MACRO*, 5–9
  - forced-immediate • *Patch*, PAT–21
  - general • *MACRO*, 5–15
  - general register • *MACRO*, 5–1
    - summary • *MACRO*, 8–28
  - immediate • *MACRO*, 5–14
    - usage restricted in vector memory instructions • *MACRO*, 10–51, 10–53
  - index • *MACRO*, 5–16
  - literal • *MACRO*, 5–10, 5–15
  - operand specifier formats • *MACRO*, 8–18
  - program counter • *MACRO*, 5–12
    - summary • *MACRO*, 8–29
  - register • *MACRO*, 5–4
  - register deferred • *MACRO*, 5–5
  - relative • *MACRO*, 5–12, 6–19, 6–22
  - relative deferred • *MACRO*, 5–13, 6–19
  - summary • *MACRO*, 5–1, C–10
- Address location
  - changing the value • *Delta/XDelta*, DELTA–18
  - closing current • *Delta/XDelta*, DELTA–22, DELTA–27
  - command strings (XDELTA) • *Delta/XDelta*, DELTA–38
  - displaying contents of current • *Delta/XDelta*, DELTA–17
  - displaying from other processes • *Delta/XDelta*, DELTA–17
  - displaying in ASCII • *Delta/XDelta*, DELTA–25
  - displaying location pointed to by current location • *Delta/XDelta*, DELTA–24
  - displaying next • *Delta/XDelta*, DELTA–22

Address location (Cont.)
  displaying previous • *Delta/XDelta*, DELTA–23
  displaying range of • *Delta/XDelta*, DELTA–17
  listing for executive images • *Delta/XDelta*, DELTA–44
  of command strings in XDELTA • *Delta/XDelta*, DELTA–9
  PCB • *Delta/XDelta*, DELTA–9
  referencing • *Delta/XDelta*, DELTA–10
  using base address and offsets for • *Delta/XDelta*, DELTA–11
/ADDRESS qualifier • *Debugger*, 8–6, CD–47, CD–83, CD–250; *System Dump Analyzer*, SDA–87, SDA–98, SDA–123
Address space • *Programming Resources*, 10–1
  allocating by page • *Programming Resources*, 10–1, 10–3
  allocating in zones • *Programming Resources*, 10–1
  deallocating by page • *Programming Resources*, 10–1, 10–3
  zones • *Programming Resources*, 10–1
Address storage directive (.ADDRESS) • *MACRO*, 6–4
Address symbol
  current • *Delta/XDelta*, DELTA–9
address_range data type • *Routines Intro*, A–2t
ADDW2 (Add Word 2 Operand) instruction • *MACRO*, 9–8
ADDW3 (Add Word 3 Operand) instruction • *MACRO*, 9–8
ADD_KEY_MAP built-in procedure • *VAXTPU*, 7–17 to 7–18
ADJUST_WINDOW built-in procedure • *VAXTPU*, 7–19 to 7–23
ADP$L_AVECTOR • *Device Support (A)*, 16–9
ADP$L_BIMASTER • *Device Support (A)*, 16–10, 16–17
ADP$L_BI_IDR • *Device Support (A)*, 16–10, 16–15
ADP$L_CSR • *Device Support (A)*, 16–9; *Device Support (B)*, 3–82
ADP$L_DPQFL • *Device Support (A)*, E–14; *Device Support (B)*, 3–87
ADP$L_MBASCB • *Device Support (A)*, 16–10; *Device Support (B)*, 1–7
ADP$L_MBASPTE • *Device Support (A)*, 16–10; *Device Support (B)*, 1–8
ADP$L_MR2QFL • *Device Support (A)*, E–14
ADP$L_MRQFL • *Device Support (A)*, E–14
ADP$L_VECTOR • *Device Support (A)*, 14–30
ADP$W_ADPTYPE • *Device Support (A)*, 16–9; *Device Support (B)*, 2–3
ADP$W_BI_VECTOR • *Device Support (A)*, 16–10, 16–15
ADP$W_DPBITMAP • *Device Support (A)*, 14–17; *Device Support (B)*, 3–96
ADP$W_TR • *Device Support (A)*, 16–9, 16–18
ADP$W_XBIA_TR • *Device Support (A)*, 16–17
ADP (adapter control block) • *Device Support (A)*, 1–6, 14–15 to 14–16; *Device Support (B)*, 1–4 to 1–11
  address • *Device Support (A)*, 4–7, 14–17, 14–19, 14–30; *Device Support (B)*, 1–26, 1–36
  alternate map register allocation information • *Device Support (B)*, 1–10
  alternate map register wait queue • *Device Support (B)*, 1–10
  data path allocation information • *Device Support (A)*, 14–17; *Device Support (B)*, 1–9
  data path wait queue • *Device Support (A)*, 14–17; *Device Support (B)*, 1–7
  fields supporting ADPDISP macro • *Device Support (B)*, 2–3
  for generic VAXBI device • *Device Support (A)*, 16–9 to 16–10
  for MBA • *Device Support (A)*, 15–4, 15–7 to 15–8
  for VAXBI adapter • *Device Support (A)*, 16–10
  map register allocation information • *Device Support (B)*, 1–9
  map register wait queue • *Device Support (B)*, 1–8
  size • *Device Support (B)*, 1–4
ADPDISP macro • *Device Support (A)*, 5–5 to 5–6; *Device Support (B)*, 2–2 to 2–4
  examples • *Device Support (B)*, 2–4
ADWC (Add with Carry) instruction • *MACRO*, 9–9
AEN
  See Asynchronous event notification
Affinity
  See Device affinity
/AFTER qualifier • *Debugger*, CD–131, CD–188, CD–200
AGAIN command • *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–22
Aggregate
  DEPOSIT command • *Debugger*, 4–17, 4–19, 11–7, 11–8, CD–61
  EXAMINE command • *Debugger*, 4–17, 4–19, 11–6, 11–7, 11–8, CD–85
  SET WATCH command • *Debugger*, 3–18, 11–3
AID
  See Area identification number
Alarm ACE • *System Services Intro*, 3–19
  See also ACE
  format of • *System Services Intro*, 3–19

Alarm ACE (Cont.)
  purpose of • *System Services Intro,* 3–19
Algorithm • *RTL Math,* 1–3
  for memory allocation • *RTL Library,* 5–7
  for naming buffer change journal file • *VAXTPU,* 1–12
ALIGN command • *Patch,* PAT–18, PAT–38, PAT–39
  with /ABSOLUTE qualifier • *Patch,* PAT–27
.ALIGN directive • *MACRO,* 6–5
Alignment
  data • *Programming Resources,* 8–4
Alignment attribute • *RTL Library,* 5–11
Alignment boundary type field
  See XAB$B_ALN field
Alignment of data transfer • *Device Support (A),* 14–3
Alignment vector • *MACRO,* 10–29, 10–49
ALL keyword
  with EXPAND_NAME • *VAXTPU,* 7–135
  with REMOVE_KEY_MAP • *VAXTPU,* 7–313
  with SET (BELL) • *VAXTPU,* 7–355
  with SET (DEBUG) • *VAXTPU,* 7–364
  with UPDATE • *VAXTPU,* 7–538
Allocation • *File Applications,* 3–23, 4–30, A–1
ALLOCATION attribute • *File Def Language,* FDL–6, FDL–17
Allocation class • *System Services,* SYS–206; *Device Support (B),* 1–28
Allocation control extended address block
  See XABALL block
Allocation options field
  See XAB$B_AOP field
Allocation quantity field
  See FAB$L_ALQ field
Allocation-quantity option • *File Applications,* 4–30
ALLOCATION secondary attribute • *File Applications,* 3–24, 4–30
/ALL qualifier • *Debugger,* CD–162; *System Dump Analyzer,* SDA–51, SDA–108, SDA–111, SDA–115, SDA–126, SDA–143, SDA–157, SDA–161
  CANCEL BREAK command • *Debugger,* CD–17
  CANCEL DISPLAY command • *Debugger,* CD–20
  CANCEL IMAGE command • *Debugger,* CD–22
  CANCEL MODULE command • *Debugger,* CD–24
  CANCEL TRACE command • *Debugger,* CD–30
  CANCEL WATCH command • *Debugger,* CD–34
  CANCEL WINDOW command • *Debugger,* CD–35
  DELETE command • *Debugger,* CD–57
  DELETE/KEY command • *Debugger,* CD–59
  EXTRACT command • *Debugger,* CD–101
  SEARCH command • *Debugger,* CD–121

/ALL qualifier (Cont.)
  SET IMAGE command • *Debugger,* CD–142
  SET MODULE command • *Debugger,* CD–156
  SET TASK command • *Debugger,* CD–182
  SHOW DISPLAY command • *Debugger,* CD–217
  SHOW KEY command • *Debugger,* CD–223
  SHOW PROCESS command • *Debugger,* CD–235
  SHOW TASK command • *Debugger,* CD–253
  SHOW WINDOW command • *Debugger,* CD–262
  with SET MODULE command • *Patch,* PAT–78
Alternate index • *File Applications,* 3–19; *File Def Language,* FDL–29
Alternate index structure • *Analyze/RMS_File,* ARMS–6
Alternate key • *File Applications,* 3–15, 3–16; *Analyze/RMS_File,* ARMS–7; *File Def Language,* FDL–5, FDL–29
Alternate map registers • *Device Support (A),* 14–3, 14–6, 14–23; *Device Support (B),* 1–8, 1–26 to 1–27, 2–3
  See also Map registers
  allocating • *Device Support (A),* 14–19; *Device Support (B),* 3–63 to 3–64
  allocating permanent • *Device Support (A),* 11–2, 14–20; *Device Support (B),* 1–26
  loading • *Device Support (A),* 14–22; *Device Support (B),* 2–44, 3–74 to 3–75
  number of active • *Device Support (B),* 1–10, 1–11
  number of disabled • *Device Support (B),* 1–11
  releasing • *Device Support (A),* 14–26; *Device Support (B),* 2–53, 3–84 to 3–85
  requesting • *Device Support (B),* 2–58, 3–92 to 3–93
Alternate map register wait queue • *Device Support (A),* E–14; *Device Support (B),* 1–10, 3–93
Alternate NCS library, specifying
  See /LIBRARY qualifier
Alternate record • *Analyze/RMS_File,* ARMS–7
Alternate record structure • *File Applications,* 10–22
Alternate start I/O routine • *Device Support (A),* 7–5; *Device Support (B),* 3–17
  address • *Device Support (A),* 6–4; *Device Support (B),* 1–30, 4–2
  context • *Device Support (B),* 4–2
  entry point • *Device Support (B),* 4–2
  exit method • *Device Support (B),* 4–2
  input • *Device Support (B),* 4–2
  register usage • *Device Support (B),* 4–2
  synchronization requirements • *Device Support (B),* 4–2
Alternation
  pattern (|) • *VAXTPU,* 2–16

- ALTMODE key • *I/O User's I*, 8–21
- ALWAYS keyword
  - with GSMATCH option • *Programming Resources*, 5–5
- /ANALYSIS qualifier • *File Def Language*, FDL–42, FDL–47
- Analysis section • *File Applications*, 4–4, 10–1, 10–29
  - FDL • *Analyze/RMS_File*, ARMS–14
- ANALYSIS_OF_AREA attribute • *File Applications*, 10–1, 10–25; *File Def Language*, FDL–2, FDL–3
- ANALYSIS_OF_KEY attribute • *File Applications*, 10–1, 10–25; *File Def Language*, FDL–2, FDL–4
- ANALYZE command • *System Dump Analyzer*, SDA–32
  - /CRASH_DUMP qualifier • *System Dump Analyzer*, SDA–35
  - /RELEASE qualifier • *System Dump Analyzer*, SDA–36
  - /SYMBOL qualifier • *System Dump Analyzer*, SDA–37
  - /SYSTEM qualifier • *System Dump Analyzer*, SDA–38
- ANALYZE/CRASH_DUMP command • *System Dump Analyzer*, SDA–6, SDA–32
- ANALYZE/CRASH_DUMP/RELEASE command • *System Dump Analyzer*, SDA–3
- /ANALYZE qualifier • *File Applications*, 10–29
- ANALYZE/RMS_FILE
  - See Analyze/RMS_File Utility
- ANALYZE/RMS_FILE command • *Programming Resources*, 8–55
- Analyze/RMS_File Utility (ANALYZE/RMS_FILE) • *Programming Resources*, 1–38; *File Applications*, 1–12, 10–1, 10–29; *File Def Language*, FDL–39
  - ANALYSIS_OF_AREA section • *File Def Language*, FDL–3
  - ANALYSIS_OF_KEY section • *File Def Language*, FDL–4
  - analyzing file structure interactively • *Analyze/RMS_File*, ARMS–1
  - creating FDL files • *Analyze/RMS_File*, ARMS–1; *File Def Language*, FDL–39, FDL–40
  - directing output from • *Analyze/RMS_File*, ARMS–10
  - duplicate key values • *File Def Language*, FDL–5
  - error conditions • *Analyze/RMS_File*, ARMS–7
  - examining prolog • *File Applications*, 3–16
  - examples
    - analyzing a file interactively • *Analyze/RMS_File*, ARMS–36

Analyze/RMS_File Utility (ANALYZE/RMS_FILE)
  - examples (Cont.)
    - analyzing a remote file • *Analyze/RMS_File*, ARMS–36
    - creating an FDL file • *Analyze/RMS_File*, ARMS–36
    - creating an FDL file from a remote file • *Analyze/RMS_File*, ARMS–36
  - exiting from • *Analyze/RMS_File*, ARMS–10
  - file optimizing • *File Applications*, 4–4
  - invoking • *Analyze/RMS_File*, ARMS–10
  - list of functions • *Analyze/RMS_File*, ARMS–10
  - output file default name • *Analyze/RMS_File*, ARMS–16
  - restrictions • *Analyze/RMS_File*, ARMS–11
  - user response to errors • *Analyze/RMS_File*, ARMS–8
  - using to obtain information about VAX RMS Journaling • *Analyze/RMS_File*, ARMS–1
  - with DECnet–VAX • *Analyze/RMS_File*, ARMS–7
  - with FDL files • *File Applications*, 4–2
- ANALYZE/SYSTEM command • *System Dump Analyzer*, SDA–2, SDA–32
- Analyzing
  - crash dump
    - See also Crash dump
    - See also System failure
    - privileges required • *System Dump Analyzer*, SDA–32
    - requirements • *System Dump Analyzer*, SDA–6
  - running system • *System Dump Analyzer*, SDA–38
    - See also System
    - privileges required • *System Dump Analyzer*, SDA–8, SDA–32
- Anchored search • *VAXTPU*, 7–24
- ANCHOR keyword • *VAXTPU*, 7–24 to 7–25
  - with SEARCH • *VAXTPU*, 7–327, 7–328
  - with SEARCH_QUIETLY • *VAXTPU*, 7–332
- AND operator • *System Dump Analyzer*, SDA–12; *MACRO*, 3–16; *VAXTPU*, 3–7
- ANL file type • *File Applications*, 10–5; *Analyze/RMS_File*, ARMS–16
- ANSI escape sequence • *I/O User's I*, B–9
- "Ansi_crt" string constant parameter to GET_INFO • *VAXTPU*, 7–196
- Ansynchronous events • *RTL Screen Management*, 4–1
- ANY built-in procedure • *VAXTPU*, 7–26 to 7–27
- ANY_CYLINDER option • *File Applications*, 4–31
- AOBLEQ (Add One and Branch Less Than or Equal) instruction • *MACRO*, 9–46

AOBLSS (Add One and Branch Less Than) instruction • *MACRO*, 9–47
%AP • *Debugger*, 4–23, D–3
AP (argument pointer) • *System Dump Analyzer*, SDA–13
AP (argument pointer) symbol • *System Dump Analyzer*, SDA–13
APL
  See VAX APL
APL data type declaration • *Routines Intro*, A–15
APL Implementation table • *Routines Intro*, A–15
Apostrophe (')
  ASCII string delimiter • *Debugger*, 4–17
  instruction delimiter • *Debugger*, 4–22
/APPEND qualifier • *Debugger*, CD–101; *Convert*, CONV–1, CONV–7
APPEND_LINE built-in procedure • *VAXTPU*, 7–28 to 7–29
Application
  characteristics of parallel • *RTL Parallel Processing*, 1–3
  creating • *RTL Parallel Processing*, 2–1
  deleting • *RTL Parallel Processing*, 2–2
  items to consider when developing • *RTL Parallel Processing*, 5–1
  naming • *RTL Parallel Processing*, 2–4
  use of DECwindows VAXTPU built-in procedures in • *VAXTPU*, B–1 to B–33
Application ACE • *System Services Intro*, 3–20
  format of • *System Services Intro*, 3–20
  purpose of • *System Services Intro*, 3–20
Application design • *File Applications*, 2–1, 2–24
  shared access consideration • *File Applications*, 3–3
  space consideration • *File Applications*, 3–2
  speed consideration • *File Applications*, 3–1
Application programs
  connecting to LAT ports • *I/O User's I*, 8–48
Approximate key match • *File Applications*, 8–11
AQB (ACP queue block) • *System Dump Analyzer*, SDA–99
ARB (access rights block) • *Device Support (A)*, 4–10; *Device Support (B)*, 1–42
ARB built-in procedure • *VAXTPU*, 7–30 to 7–31
Arc cosine
  in degrees • *RTL Math*, MTH–6, MTH–70
  in radians • *RTL Math*, MTH–3, MTH–68
Arc sine
  in degrees • *RTL Math*, MTH–11, MTH–74
  in radians • *RTL Math*, MTH–9, MTH–72
Arc tangent
  hyperbolic • *RTL Math*, MTH–21, MTH–84
Arc tangent (Cont.)
  in degrees • *RTL Math*, MTH–15, MTH–19, MTH–78, MTH–82
  in radians • *RTL Math*, MTH–13, MTH–17, MTH–76, MTH–80
Area • *File Applications*, 3–23; *File Def Language*, FDL–28
  multiple • *File Applications*, 3–6, 3–23, 3–25
    defining in an FDL file • *File Applications*, 3–24
    on a volume set • *File Applications*, 3–23
Area allocation quantity field
  See XAB$L_ALQ field
AREA attribute • *File Def Language*, FDL–2, FDL–6, FDL–27, FDL–28, FDL–40
Area default extension quantity field
  See XAB$W_DEQ field
AREA DESCRIPTOR structure • *File Applications*, 10–19
Area extension size • *RTL Library*, 5–9
Area identification number (AID)
  program example • *RMS*, 4–8
Area identification number field
  See XAB$B_AID field
AREA primary attribute • *File Applications*, 3–23
  BEST_TRY_CONTIGUOUS secondary attribute • *File Applications*, 4–31
  EXACT_POSITIONING secondary attribute • *File Applications*, 4–31
  POSITION secondary attribute • *File Applications*, 4–31
  VOLUME secondary attribute • *File Applications*, 4–32
Areas option • *File Applications*, 4–30
Argument
  access mechanism • *Modular Procedures*, B–8
  actual • *MACRO*, 4–1
  adding new • *Modular Procedures*, 6–3
  characteristics of • *Modular Procedures*, B–1; *System Services Intro*, 2–3; *RTL Intro*, 3–3, 3–6
    passing mechanism • *System Services Intro*, 1–7
  delimiters • *RMS*, 3–10
  device- or function-dependent • *I/O User's I*, 1–2
  explicit • *Modular Procedures*, 2–3
  implicit • *Modular Procedures*, 2–3
  in a macro • *MACRO*, 4–1
  initialization and control block store macros • *RMS*, 3–8
  length • *MACRO*, 6–64

Argument (Cont.)
list • *I/O User's I*, A–1 to A–9; *I/O User's II*, A–1 to A–6
LPA11-K subroutine • *I/O User's I*, 4–16
mechanism array • *System Services Intro*, 10–10
number of • *MACRO*, 6–63
optional • *Modular Procedures*, 2–11, A–3
order • *Modular Procedures*, 2–11, A–2
passing • *RMS*, 1–2
passing mechanism • *Modular Procedures*, B–8; *RTL Intro*, 2–21
separator • *RMS*, 3–6
separator in VMS RMS coding • *RMS*, 3–6
signal array • *System Services Intro*, 10–10
specifying • *System Services Intro*, 2–7
specifying as run-time values • *RMS*, 3–9
to FAB • *RMS*, 1–2
to RAB • *RMS*, 1–4
VMS data types • *Modular Procedures*, B–6
VMS Usage • *Modular Procedures*, B–1; *System Services Intro*, 1–6; *RTL Intro*, 2–6
Argument blocks • *Modular Procedures*, 6–4
Argument data type • *Routines Intro*, 2–15; *System Services Intro*, 1–7
Argument keyword
delimiting for VMS RMS service • *RMS*, 3–10
Argument list • *Routines Intro*, 2–4; *System Services Intro*, 2–3
count field • *RMS*, 2–5
creating • *System Services Intro*, 2–8
definition of • *Routines Intro*, 2–3
description • *RMS*, 2–4
error routine address field • *RMS*, 2–5
evaluation • *Routines Intro*, 2–6
for AST service routine • *System Services Intro*, 5–4
for condition handler • *System Services Intro*, 10–8
format • *Routines Intro*, 2–4
for system services • *System Services Intro*, 2–3
interpreting • *Routines Intro*, 2–4
new FAB address field • *RMS*, 2–5
passing to service • *RMS*, 3–10
passing to VMS RMS service • *RMS*, 3–10
success routine address field • *RMS*, 2–5
using macros • *System Services Intro*, 2–6
Argument-passing mechanism • *System Services Intro*, 1–8
Argument pointer
See AP
Arguments heading • *Routines Intro*, 1–7; *System Services Intro*, 1–6
Argument substitution • *RTL Screen Management*, 5–15
arg_list data type • *Routines Intro*, A–2t
Arithmetic
See also Condition handler
using system routines • *Programming Resources*, 1–24
Arithmetic expression • *VAXTPU*, 3–9
evaluating • *Patch*, PAT–59
special operators for • *Patch*, PAT–23
Arithmetic instruction
decimal string • *MACRO*, 9–144
floating-point • *MACRO*, 9–101
integer • *MACRO*, 9–5
Arithmetic operations • *RTL Screen Management*, 5–16
Arithmetic operators • *Delta/XDelta*, DELTA–10; *System Dump Analyzer*, SDA–12
Arithmetic shift • *Delta/XDelta*, DELTA–10
Arithmetic shifting operator (@) • *System Dump Analyzer*, SDA–13
Arithmetic shift operator • *MACRO*, 3–16
Array
conversion of • *RTL Math*, MTH–63
mechanism • *System Services Intro*, 10–10
signal • *System Services Intro*, 10–10
virtual address • *System Services Intro*, 11–5
ARRAY data type • *VAXTPU*, 2–2 to 2–3
See also CREATE_ARRAY built-in procedure
Array descriptor • *Routines Intro*, 2–25
Array type • *Debugger*, 4–17
vector register • *Debugger*, 11–7
ASB (asynchronous save block) • *System Dump Analyzer*, SDA–76
.ASCIC directive • *MACRO*, 6–8
/ASCIC qualifier • *Debugger*, CD–61, CD–85
.ASCID directive • *MACRO*, 6–9
effect on position independence • *Linker*, 4–5
effect on shareability • *Linker*, 1–10, 4–4
/ASCID qualifier • *Debugger*, CD–62, CD–85
ASCII
character set • *MACRO*, A–1
depositing string • *Delta/XDelta*, DELTA–37
displaying contents in • *Delta/XDelta*, DELTA–25
operator • *MACRO*, 3–12
ASCII (8-bit) code • *I/O User's I*, 2–8
/ASCII-/NOASCII qualifier
with DELETE command • *Patch*, PAT–53
with DEPOSIT command • *Patch*, PAT–56, PAT–57
with EVALUATE command • *Patch*, PAT–60
with EXAMINE command • *Patch*, PAT–63

Index

/ASCII-/NOASCII qualifier (Cont.)
  with REPLACE command • *Patch*, PAT–72
  with SET MODE command • *Patch*, PAT–76
  with VERIFY command • *Patch*, PAT–91
ASCII built-in procedure • *VAXTPU*, 7–32 to 7–34
ASCII character
  delimiting in control block fields • *RMS*, 3–6, 3–7
ASCII character set
  See DEC Multinational Character Set
.ASCII directive • *MACRO*, 6–10
ASCII-NOASCII mode • *Patch*, PAT–16
ASCII pad character • *Convert*, CONV–18
/ASCII qualifier • *Debugger*, CD–62, CD–86
ASCII space character
  conversion function • *Convert*, CONV–3
  using as pad character • *National Char Set*, NCS–10
ASCII string
  converting to binary • *System Services*, SYS–27
  entering • *Patch*, PAT–20
ASCII string storage directive • *MACRO*, 6–7
  counted (.ASCIC) • *MACRO*, 6–8
  string (.ASCII) • *MACRO*, 6–10
  string-descriptor (.ASCID) • *MACRO*, 6–9
  zero-terminated (.ASCIZ) • *MACRO*, 6–11
ASCII string type • *Debugger*, 4–17, 4–28, CD–61, CD–85, CD–195
ASCII time • *System Services Intro*, 9–7
/ASCIW qualifier • *Debugger*, CD–62, CD–86
.ASCIZ directive • *MACRO*, 6–11
/ASCIZ qualifier • *Debugger*, CD–62, CD–86
$ASCTIM
  RTL jacket routine • *RTL Library*, LIB–401
ASHL (Arithmetic Shift Long) instruction • *MACRO*, 9–10
ASHP (Arithmetic Shift and Round Packed) instruction • *MACRO*, 9–150
ASHQ (Arithmetic Shift Quad) instruction • *MACRO*, 9–10
Assembler • *Programming Resources*, 1–9
Assembler directives,
  summary • *MACRO*, C–1
Assembler notation • *MACRO*, 10–17
Assembly termination • *MACRO*, 6–25
Assembly termination directive (.END) • *MACRO*, 6–25
ASSIGN command • *Linker*, LINK–21; *System Services Intro*, 6–2; *File Applications*, 4–14
  /TRANSLATION_ATTRIBUTES qualifier • *File Applications*, 5–7
Assignment statement • *MACRO*, 1–1, 3–17; *VAXTPU*, 3–21

AST (asynchronous system trap) • *Programming Resources*, 4–7; *Debugger*, 9–17; *RTL Library*, 2–22; *Device Support (B)*, 3–6 to 3–7
  See also Attention AST
  See also Synchronization
  access mode • *System Services Intro*, 5–2
  blocking • *System Services Intro*, 12–9, 12–15
  CALL command • *Debugger*, 9–17, CD–10
  condition handling at AST level • *Modular Procedures*, 3–26
  control • *Device Support (B)*, 1–86
  declaring • *System Services Intro*, 5–3; *System Services*, SYS–121
  definition • *Modular Procedures*, 3–19
  delivering • *Programming Resources*, 4–8; *System Services Intro*, 5–5; *Device Support (A)*, 3–4; *Device Support (B)*, 3–2, 3–11
  disabling • *Debugger*, CD–68; *System Services*, SYS–400
  disabling interrupts • *Modular Procedures*, 3–24
  displaying AST handling conditions • *Debugger*, CD–210
  enabling • *Debugger*, CD–80; *System Services*, SYS–400
  example • *System Services Intro*, 5–6
  execution • *Programming Resources*, 4–7
  for aborted I/O request • *Device Support (B)*, 3–11
  handler • *Modular Procedures*, 3–19, 3–21
  I/O at AST level • *Modular Procedures*, 3–25, A–5
  interrupt • *Modular Procedures*, 3–19
  out of band • *Device Support (A)*, 11–8; *Device Support (B)*, 1–86
  parameter • *System Services Intro*, 5–4
  process-requested • *Device Support (A)*, 4–20; *Device Support (B)*, 3–7, 3–10, 3–73
  process wait state • *System Services Intro*, 5–3
  queuing • *Device Support (A)*, 3–4; *Device Support (B)*, 3–73
  quota • *System Services Intro*, 7–4; *I/O User's I*, 3–24, 4–14, 6–13, 7–5, 8–43
  reentrancy • *Modular Procedures*, 3–19, 3–20, A–5
  service routine • *Modular Procedures*, 3–19; *System Services Intro*, 5–4
  setting for power recovery • *System Services*, SYS–409
  setting timer for • *System Services*, SYS–406
  SHOW CALLS command • *Debugger*, 9–17
  special kernel-mode • *Device Support (A)*, 3–4, 3–5, 4–20, 7–8, 7–8; *Device Support (B)*, 1–12
  system service • *System Services Intro*, 5–1
  thread • *Modular Procedures*, 3–19

AST (asynchronous system trap) (Cont.)
  user specified • *Device Support (B)*, 1–39
  writing • *Programming Resources*, 4–7
  writing AST-reentrant procedures • *Modular Procedures*, 3–20
AST control block
  See ACB
AST-driven program
  debugging • *Debugger*, 9–17
Asterisk (*)
  HELP command • *Debugger*, CD–107
  multiplication operator • *Debugger*, D–7
ASTLM (AST limit) quota
  effect of canceling wakeup on • *System Services*, SYS–45
ASTLVL (AST level) processor register • *Device Support (A)*, 3–4
  displaying • *System Dump Analyzer*, SDA–90
AST procedure (for connect to interrupt facility) • *Device Support (A)*, 19–19
/AST qualifier • *Debugger*, 9–17, CD–11
AST reentrant • *RTL Screen Management*, 4–1
AST routines
  global symbols • *System Dump Analyzer*, SDA–60
AST service routine (for connect to interrupt facility) • *Device Support (A)*, 19–9, 19–11, 19–12
ast_procedure data type • *Routines Intro*, A–2t
ASYNCHRONOUS attribute • *File Def Language*, FDL–9
Asynchronous DDCMP driver • *I/O User's II*, 5–1
  AST service routine address • *I/O User's II*, 5–10
  attention AST • *I/O User's II*, 5–10
  capabilities • *I/O User's II*, 5–1
  characteristics • *I/O User's II*, 5–7 to 5–8
    controller • *I/O User's II*, 5–7, 5–10
    device • *I/O User's II*, 5–2
    extended • *I/O User's II*, 5–8
    modifying • *I/O User's II*, 5–7
    tributary • *I/O User's II*, 5–10
  controller
    mode • *I/O User's II*, 5–8
    starting • *I/O User's II*, 5–6
  controller counter parameter IDs • *I/O User's II*, 5–11
  device characteristics • *I/O User's II*, 5–2
  duplex modes • *I/O User's II*, 5–7
  enable attention AST • *I/O User's II*, 5–9
  enable modem • *I/O User's II*, 5–7
  errors • *I/O User's II*, 5–3
  error summary bits • *I/O User's II*, 5–3
  extended characteristics • *I/O User's II*, 5–8
  full-duplex mode • *I/O User's II*, 5–1

Asynchronous DDCMP driver (Cont.)
  function codes • *I/O User's II*, 5–4, A–4
  function modifiers • *I/O User's II*, 5–5, 5–6, 5–8 to 5–10
  I/O functions • *I/O User's II*, 5–5, 5–6, 5–10
  I/O status block • *I/O User's II*, 5–14
  message size • *I/O User's II*, 5–2, 5–5, 5–6
  modem
    disabling line • *I/O User's II*, 5–9
  modifying characteristics • *I/O User's II*, 5–7
  parameter ID • *I/O User's II*, 5–7
  point-to-point
    configuration • *I/O User's II*, 5–1
  privilege • *I/O User's II*, 5–5
  protocol • *I/O User's II*, 5–7
    starting • *I/O User's II*, 5–8
    stopping • *I/O User's II*, 5–9
  quotas • *I/O User's II*, 5–1
  read function • *I/O User's II*, 5–5
  read internal counters • *I/O User's II*, 5–10
  sense mode function • *I/O User's II*, 5–10
  set controller mode • *I/O User's II*, 5–6
    characteristics • *I/O User's II*, 5–7 to 5–8
    message size • *I/O User's II*, 5–8
    P2 buffer • *I/O User's II*, 5–7
    parameter ID • *I/O User's II*, 5–7
  set mode function • *I/O User's II*, 5–6
  set tributary mode • *I/O User's II*, 5–8
    extended characteristics • *I/O User's II*, 5–8
    P2 buffer • *I/O User's II*, 5–8
  shutdown controller mode • *I/O User's II*, 5–9
  shutdown tributary mode • *I/O User's II*, 5–9
  starting
    controller • *I/O User's II*, 5–7
    protocol • *I/O User's II*, 5–8
    tributary • *I/O User's II*, 5–8
  status returns • *I/O User's II*, A–5
  stopping
    controller • *I/O User's II*, 5–9
    modem line • *I/O User's II*, 5–9
    protocol • *I/O User's II*, 5–9
    tributary • *I/O User's II*, 5–9
  supported device • *I/O User's II*, 5–1
  SYS$GETDVI • *I/O User's II*, 5–2
  tributary
    starting • *I/O User's II*, 5–8
    stopping • *I/O User's II*, 5–9
  tributary counter parameter IDs • *I/O User's II*, 5–13
  unit and line status • *I/O User's II*, 5–3
  write function • *I/O User's II*, 5–5

- Asynchronous event notification • *Device Support (A),* 17–2, 17–28 to 17–30; *Device Support (B),* 2–70, 2–73 to 2–90
  - example • *Device Support (A),* 17–29 to 17–30
- Asynchronous I/O option
  - See FAB$V_ASY option
  - See RAB$V_ASY option
- Asynchronous input/output • *Programming Resources,* 7–47
- Asynchronous memory management exception handling • *MACRO,* 10–19, 10–30
- Asynchronous operation • *File Applications,* 8–17, 8–18
  - contrasted with synchronous operation • *RMS,* 2–7
  - performance • *File Applications,* 9–9
  - using R0 • *RMS,* 2–5
- Asynchronous save block
  - See ASB
- Asynchronous SCSI data transfer mode
  - enabling • *I/O User's I,* 11–7, 11–13; *Device Support (A),* 17–13; *Device Support (B),* 2–88
- Asynchronous system service • *System Services Intro,* 2–12
- Asynchronous system trap
  - See AST
- Asynchronous system trap (AST)
  - disabling • *RTL Parallel Processing,* 5–6
  - enabling an event • *RTL Parallel Processing,* 4–6
- ASY option • *File Def Language,* FDL–9
- AT$_GENBI • *Device Support (B),* 1–33
- AT$_MBA • *Device Support (B),* 1–33
- AT$_UBA • *Device Support (B),* 1–33
- Atomic data type • *Routines Intro,* 2–15
- At sign (@)
  - contents-of operator • *Debugger,* D–7
  - execute-procedure command • *Debugger,* 8–1, CD–7
  - SET ATSIGN command • *Debugger,* CD–129
  - SHOW ATSIGN command • *Debugger,* CD–211
- ATTACH built-in procedure • *VAXTPU,* 7–35 to 7–36
- ATTACH command • *Debugger,* 3–5, CD–9; *System Dump Analyzer,* SDA–41
- Attached processor
  - See Secondary processor
- Attention AST
  - See also AST
  - asynchronous DDCMP driver • *I/O User's II,* 5–9
  - blocking • *Device Support (B),* 1–82, 1–83
  - delivering • *Device Support (B),* 3–2
  - disabling • *Device Support (B),* 3–6 to 3–7
- Attention AST (Cont.)
  - DMC11/DMR11 driver • *I/O User's II,* 1–7
  - DMP11/DMF32 driver • *I/O User's II,* 2–19
  - DR11–W/DRV11–WA driver • *I/O User's II,* 3–14
  - enabling • *Device Support (B),* 3–6 to 3–7
  - Ethernet/802 drivers • *I/O User's II,* 6–36
  - flushing • *Device Support (B),* 3–4
  - mailbox • *I/O User's I,* 7–9
  - terminal • *I/O User's I,* 8–42
- Attention condition • *Device Support (A),* 15–9 to 15–10
  - See also MASSBUS
  - See also MBA
  - See also MBA$L_AS
- Attention summary register
  - See MBA$L_AS
- Attributes • *File Applications,* 4–2, 4–9; *File Def Language,* FDL–1, FDL–46
  - buffer • *VAXTPU,* 7–60
  - display • *Debugger,* 7–3, 7–6, 7–9, 7–20, CD–123, CD–244
  - for TPU
    - setting records • *VAXTPU,* 7–448
  - program section
    - absolute • *Linker,* 6–4
    - concatenated • *Linker,* 1–12, 6–4
    - executable • *Linker,* 6–5
    - global • *Linker,* 1–13, 6–5, 6–12
    - in image section generation • *Linker,* 6–15
    - in shareable images • *Linker,* 4–3
    - local • *Linker,* 1–13, 6–5
    - modification of • *Linker,* 6–3
    - nonexecutable • *Linker,* 6–5
    - nonposition-independent • *Linker,* 1–13, 6–6
    - nonshareable • *Linker,* 1–13, 6–6
    - nonvector • *Linker,* 1–13, 6–6
    - nonwritability • *Linker,* 6–6
    - nonwritable • *Linker,* 1–13
    - overlaid • *Linker,* 1–12, 6–4
    - position-independent • *Linker,* 1–13, 6–6
    - relocatable • *Linker,* 6–4
    - shareable • *Linker,* 1–13, 6–6
    - vector • *Linker,* 1–13, 6–6
    - writability • *Linker,* 6–6
    - writable • *Linker,* 1–13
  - window • *VAXTPU,* 7–78
    - with DECwindows • *Debugger,* 1–10
- Audit trail
  - changing the value of • *SUMSLP,* SUM–12
- Autoconfiguration
  - See also System Generation Utility

Autoconfiguration (Cont.)
- driver control of • *Device Support (A),* 12–21
- of SCSI device • *I/O User's I,* 11–9; *Device Support (A),* 17–30

Autodecrement mode • *MACRO,* 5–7
- operand specifier format • *MACRO,* 8–21

Autoincrement deferred mode • *MACRO,* 5–6
- operand specifier format • *MACRO,* 8–20

Autoincrement mode • *MACRO,* 5–5
- operand specifier format • *MACRO,* 8–19

Automatic initialization • *RTL Parallel Processing,* 2–1
AUTO window, DECwindows • *Debugger,* 1–10
AUTO_REPEAT keyword • *VAXTPU,* 7–353
"Auto_repeat" string constant parameter to GET_INFO • *VAXTPU,* 7–196
/AW
- See /ASCIW qualifier

/AZ
- See /ASCIZ qualifier

B

BACK command • *Analyze/RMS_File,* ARMS–23
Backplane interconnect • *Device Support (A),* 1–11, 1–16, 14–2
- See also CMI
- See also Q22 bus
- See also SBI
- See also VAXBI

Backplane interconnect interface chip
- See BIIC

Backslash (\ )
- current value • *Debugger,* 4–6
- global-symbol specifier • *Debugger,* 5–10, CD–170, D–7
- path name delimiter • *Debugger,* 5–9, 6–4, D–7
  - with DECwindows • *Debugger,* 1–10, 1–28

BACKUP attribute • *File Def Language,* FDL–15
Backup date and time field
- See XAB$Q_BDT field

Backup Utility (BACKUP) • *File Applications,* 10–2
- copying system dump file • *System Dump Analyzer,* SDA–4
- eliminating extents • *File Applications,* 9–8
- making archive copies • *File Applications,* 10–31

Backward indexing • *RTL Math,* 2–6
BADDALRQSZ bugcheck • *Device Support (B),* 3–3, 3–19
Bad page list
- displaying • *System Dump Analyzer,* SDA–115

/BAD qualifier • *System Dump Analyzer,* SDA–115
Balance set
- swapping • *System Services Intro,* 11–7

Barrier
- adjusting a quorum for • *RTL Parallel Processing,* 4–4
- creating • *RTL Parallel Processing,* 4–2
- definition of • *RTL Parallel Processing,* 4–2
- deleting • *RTL Parallel Processing,* 4–3
- reading • *RTL Parallel Processing,* 4–3
- setting a quorum for • *RTL Parallel Processing,* 4–4
- waiting at • *RTL Parallel Processing,* 4–3

Barrier synchronization
- See also Parallel processing
- advantages and disadvantages • *RTL Parallel Processing,* 5–7
- PPL$ routines for • *RTL Parallel Processing,* 4–2 to 4–4

Base
- of numeric constant
  - specifying • *VAXTPU,* 3–37

Base address
- cluster • *Linker,* 6–15
- defaults for images • *Linker,* 1–7, 3–5
- image section in map • *Linker,* 5–5
- specification of • *Linker,* 3–6
- system image • *Linker,* 1–7, 3–5, LINK–19

Based image
- creation of • *Linker,* 1–7, 3–5
- memory allocation for • *Linker,* 1–7, 3–5, 4–4
- rules for upward compatibility • *Linker,* 1–11, 4–9

Base message number directive (.BASE)
- in message source file • *Message,* MSG–16

Base operand specifier • *MACRO,* 8–26
Base register
- loading • *Delta/XDelta,* DELTA–40
- symbol for • *Delta/XDelta,* DELTA–9

BASIC
- See VAX BASIC

BASIC data type declaration • *Routines Intro,* A–18
BASIC implementation table • *Routines Intro,* A–18
BATCH clause
- for QUALIFIER clause • *Command Def,* CDU–25, CDU–33

Batch job • *VAXTPU,* 5–5
Batch job command procedure
- using a card reader • *I/O User's I,* 2–2

Batch-like editing • *VAXTPU,* 5–3
Batch queue
- default • *File Def Language,* FDL–24

Baud rate
  terminal • *I/O User's I*, 8–40
BBC (Branch on Bit Clear) instruction • *MACRO*, 9–50
BBCC (Branch on Bit Clear and Clear) instruction • *MACRO*, 9–51
BBCCI (Branch on Bit Clear and Clear Interlocked) instruction • *MACRO*, 9–52
BBCS (Branch on Bit Clear and Set) instruction • *MACRO*, 9–51
BBS (Branch on Bit Set) instruction • *MACRO*, 9–50
BBSC (Branch on Bit Set and Clear) instruction • *MACRO*, 9–51
BBSS (Branch on Bit Set and Set) instruction • *MACRO*, 9–51
BBSSI (Branch on Bit Set and Set Interlocked) instruction • *MACRO*, 9–52
BCC (Branch on Carry Clear) instruction • *MACRO*, 9–48
;B command • *Delta/XDelta*, DELTA–28
BCS (Branch on Carry Set) instruction • *MACRO*, 9–48
BDB (buffer descriptor block) • *System Dump Analyzer*, SDA–76
BDB summary page (BDBSUM) • *System Dump Analyzer*, SDA–76
/BEFORE qualifier • *Librarian*, LIB–14; *National Char Set*, NCS–23
BEGINNING_OF built-in procedure • *VAXTPU*, 7–37 to 7–38
BELL keyword • *VAXTPU*, 7–355
  with SET (MESSAGE_ACTION_TYPE) • *VAXTPU*, 7–426
"Bell" string constant parameter to GET_INFO • *VAXTPU*, 7–205
BEQL (Branch on Equal) instruction • *MACRO*, 9–48
BEQLU (Branch on Equal Unsigned) instruction • *MACRO*, 9–48
BEST_TRY_CONTIGUOUS attribute • *File Def Language*, FDL–6, FDL–18
BEST_TRY_CONTIGUOUS secondary attribute • *File Applications*, 3–23, 4–31
"Beyond_eob" string constant parameter to GET_INFO • *VAXTPU*, 7–185
"Beyond_eol" string constant parameter to GET_INFO • *VAXTPU*, 7–185, 7–220
BGEQ (Branch on Greater Than or Equal) instruction • *MACRO*, 9–48
BGEQU (Branch on Greater Than or Equal Unsigned) instruction • *MACRO*, 9–48
BGTR (Branch on Greater Than) instruction • *MACRO*, 9–48
BGTRU (Branch on Greater Than Unsigned) instruction • *MACRO*, 9–48
BI
  See VAXBI bus
BICB2 (Bit Clear Byte 2 Operand) instruction • *MACRO*, 9–11
BICB3 (Bit Clear Byte 3 Operand) instruction • *MACRO*, 9–11
BICL2 (Bit Clear Long 2 Operand) instruction • *MACRO*, 9–11
BICL3 (Bit Clear Long 3 Operand) instruction • *MACRO*, 9–11
BICPSW (Bit Clear PSW) instruction • *MACRO*, 9–71
BICW2 (Bit Clear Word 2 Operand) instruction • *MACRO*, 9–11
BICW3 (Bit Clear Word 3 Operand) instruction • *MACRO*, 9–11
BID (block identifier) field • *RMS*, 2–1
BIIC$L_BCICR • *Device Support (A)*, 16–16, 16–28
BIIC$L_BER • *Device Support (A)*, 16–7, 16–15, 16–16, 16–26
BIIC$L_BICSR • *Device Support (A)*, 16–13, 16–24 to 16–26
BIIC$L_DTREG • *Device Support (A)*, 16–7, 16–24
BIIC$L_EAR • *Device Support (A)*, 16–28
BIIC$L_EICR • *Device Support (A)*, 16–11, 16–15, 16–26 to 16–27
BIIC$L_GPR0 • *Device Support (A)*, 16–30
BIIC$L_GPR1 • *Device Support (A)*, 16–30
BIIC$L_GPR2 • *Device Support (A)*, 16–30
BIIC$L_GPR3 • *Device Support (A)*, 16–30
BIIC$L_IDR • *Device Support (A)*, 16–15, 16–27
BIIC$L_IPIDR • *Device Support (A)*, 16–27
BIIC$L_IPIMR • *Device Support (A)*, 16–27
BIIC$L_IPISR • *Device Support (A)*, 16–27
BIIC$L_IPISTPF • *Device Support (A)*, 16–29
BIIC$L_SAR • *Device Support (A)*, 16–27
BIIC$L_UICR • *Device Support (A)*, 16–11, 16–15, 16–29 to 16–30
BIIC$L_WSR • *Device Support (A)*, 16–28 to 16–29
BIIC$V_ARBCNTRL • *Device Support (A)*, 16–14
BIIC$V_BROKE • *Device Support (A)*, 16–13
BIIC$V_SST • *Device Support (A)*, 16–13, 16–14
BIIC$V_STS • *Device Support (A)*, 16–13, 16–14
BIIC (backplane interconnect interface chip) • *Device Support (A)*, 16–5
  clearing error register • *Device Support (A)*, 16–14, 16–15
  CSR space • *Device Support (A)*, 16–5
  enabling error interrupts • *Device Support (A)*, 16–16, 16–26
  enabling options • *Device Support (A)*, 16–16
  initializing • *Device Support (A)*, 11–2

BIIC (backplane interconnect interface chip) (Cont.)
  self-test • *Device Support (A)*, 16–13 to 16–14; *Device Support (B)*, 2–5
  setting interrupt vectors • *Device Support (A)*, 16–15
$BIICDEF macro • *Device Support (A)*, 16–5, 16–23
BIIC registers
  accessing • *Device Support (A)*, 16–5
  symbolic names • *Device Support (A)*, 16–23 to 16–30
%BIN • *Debugger*, 4–12, D–5
BIN2 value • *File Def Language*, FDL–30
BIN4 value • *File Def Language*, FDL–30
BIN8 value • *File Def Language*, FDL–30
Binary data
  compression of • *Utility Routines*, DCX–1
Binary operator • *Message*, MSG–7; *System Dump Analyzer*, SDA–12 to SDA–13; *MACRO*, 3–15
  summary • *MACRO*, C–8
/BINARY qualifier • *Debugger*, 4–12, CD–81, CD–83, CD–86
Binary semaphore • *Programming Resources*, 4–17; *RTL Parallel Processing*, 4–10
  operations on • *RTL Parallel Processing*, 4–10
Binary value
  converting to ASCII string • *System Services*, SYS–165
BIOCNT (buffered I/O count) • *Convert*, CONV–24; *Device Support (A)*, 2–3
BIOLM (buffered I/O limit) quota • *System Services Intro*, 7–3
  adjusting • *Device Support (A)*, 4–20
  charging • *Device Support (A)*, 4–9, 4–12
  checking • *Device Support (A)*, 4–9
  for mailbox • *Device Support (B)*, 1–73
BIO option • *File Def Language*, FDL–2, FDL–9
BIRQ level • *Device Support (A)*, 14–33, 14–34
BISB2 (Bit Set Byte 2 Operand) instruction • *MACRO*, 9–12
BISB3 (Bit Set Byte 3 Operand) instruction • *MACRO*, 9–12
BISL2 (Bit Set Long 2 Operand) instruction • *MACRO*, 9–12
BISL3 (Bit Set Long 3 Operand) instruction • *MACRO*, 9–12
BISPSW (Bit Set PSW) instruction • *MACRO*, 9–72
BISW2 (Bit Set Word 2 Operand) instruction • *MACRO*, 9–12
BISW3 (Bit Set Word 3 Operand) instruction • *MACRO*, 9–12
BITB (Bit Test Byte) instruction • *MACRO*, 9–13
4-bit field • *File Def Language*, FDL–31
Bit field
  replace field • *RTL Library*, LIB–253
  return sign extended to longword • *RTL Library*, LIB–142
Bit field operator (<p,s,e>) • *Debugger*, D–7
BITL (Bit Test Long) instruction • *MACRO*, 9–13
Bits per inch
  See bpi
BITW (Bit Test Word) instruction • *MACRO*, 9–13
Bitwise AND operator • *RTL Math*, 1–5
Bitwise complement operator • *RTL Math*, 1–8
Bitwise exclusive OR operator • *RTL Math*, 1–5
Bitwise inclusive OR operator • *RTL Math*, 1–6
Bitwise shift • *RTL Math*, 1–9
BI_NODE_RESET macro • *Device Support (A)*, 16–13; *Device Support (B)*, 2–5
Black box testing • *Modular Procedures*, 4–2
BLANK_TABS keyword • *VAXTPU*, 7–483
BLAS
  definition of • *RTL Math*, 2–1
BLAS Level 1
  BLAS1$VIxAMAX • *RTL Math*, MTH–149
  BLAS1$VxASUM • *RTL Math*, MTH–152
  BLAS1$VxAXPY • *RTL Math*, MTH–155
  BLAS1$VxCOPY • *RTL Math*, MTH–160
  BLAS1$VxDOT • *RTL Math*, MTH–165
  BLAS1$VxNRM2 • *RTL Math*, MTH–170
  BLAS1$VxROT • *RTL Math*, MTH–173
  BLAS1$VxROTG • *RTL Math*, MTH–178
  BLAS1$VxSCAL • *RTL Math*, MTH–183
  BLAS1$VxSWAP • *RTL Math*, MTH–187
BLB (buffer lock block) • *System Dump Analyzer*, SDA–76
BLBC (Branch on Low Bit Clear) instruction • *MACRO*, 9–53
BLBS (Branch on Low Bit Set) instruction • *MACRO*, 9–53
BLEQ (Branch on Less Than or Equal) instruction • *MACRO*, 9–48
BLEQU (Branch on Less Than or Equal Unsigned) instruction • *MACRO*, 9–48
BLINK keyword
  with MARK • *VAXTPU*, 7–261
  with SELECT • *VAXTPU*, 7–337
  with SET (PROMPT_AREA) • *VAXTPU*, 7–446
  with SET (STATUS_LINE) • *VAXTPU*, 7–476
  with SET (VIDEO) • *VAXTPU*, 7–492
"Blink_status" string constant parameter to GET_INFO • *VAXTPU*, 7–221
"Blink_video" string constant parameter to GET_INFO • *VAXTPU*, 7–221

BLISS
- example in • *RTL Parallel Processing*, 6–4

BLISS-32
- See VAX BLISS-32

BLISS data type declaration • *Routines Intro*, A–22

BLISS implementation table • *Routines Intro*, A–22

BLK option • *File Def Language*, FDL–33

BLN field
- See Block length field

Block • *File Applications*, 1–4, 3–6
- I/O • *File Applications*, 8–13 to 8–14

Block boundary option
- See FAB$V_BLK option

Block code field
- See XAB$B_COD field

Blocked
- definition of • *RTL Parallel Processing*, 1–2

Block I/O
- additional services that use • *RMS*, 4–23
- applicable services • *RMS*, 4–23
- description • *RMS*, 4–23
- how implemented by VMS RMS services • *RMS*, 4–23
- how to execute • *RMS*, 4–24
- how to specify for relative and indexed files • *RMS*, 4–24
- program example • *RMS*, 4–25
- requirements for mixing with record I/O • *RMS*, 4–23
- restrictions to • *RMS*, 4–23
- services • *RMS*, 3–5
- specifying • *RMS*, 4–23
- use of NBP for sequential files • *RMS*, 4–25
- with multiple record streams • *RMS*, 4–25
- with record I/O processing • *RMS*, 4–25

Block I/O execution
- contrasted with record I/O execution • *RMS*, 4–24

Block I/O option
- See FAB$V_BIO option
- See RAB$V_BIO option

Block identifier field
- See BID field
- See FAB$B_BID field
- See NAM$B_BID field
- See RAB$B_BID field

Blocking AST
- description • *System Services Intro*, 12–9
- using • *System Services Intro*, 12–15

Block length (BLN) field
- See also NAM$B_BLN field

Block length field in allocation XAB
- See XAB$B_BLN field

Block length field in date and time XAB
- See XAB$B_BLN field

Block length field in file access block
- See FAB$B_BLN field

Block length field in file header characteristics XAB
- See XAB$B_BLN field

Block length field in item list XAB
- See XAB$B_BLN field

Block length field in key XAB
- See XAB$B_BLN field

Block length field in protection XAB
- See XAB$B_BLN field

Block length field in record access block
- See RAB$B_BLN field

Block length field in revision date and time XAB
- See XAB$B_BLN field

Block length field in summary XAB
- See XAB$B_BLN field

Block length field in terminal XAB
- See XAB$B_BLN field

Block or record I/O option
- See FAB$V_BRO option

Block size • *RTL Library*, 5–10

Block-size option • *File Applications*, 4–28

Block spanning option • *File Applications*, 3–10

Block storage allocation directives (.BLKx) • *MACRO*, 6–12

BLOCK_COUNT attribute • *File Def Language*, FDL–32

BLOCK_IO attribute • *File Def Language*, FDL–2, FDL–9

BLOCK_IO secondary attribute • *File Applications*, 7–3

BLOCK_SPAN attribute • *File Applications*, 3–10; *File Def Language*, FDL–33

BLOCK_SPAN secondary attribute • *File Applications*, 4–29

BLSS (Branch on Less Than) instruction • *MACRO*, 9–48

BLSSU (Branch on Less Than Unsigned) instruction • *MACRO*, 9–48

BMB summary page (BLBSUM) • *System Dump Analyzer*, SDA–76

BNEQ (Branch on Not Equal) instruction • *MACRO*, 9–48

BNEQU (Branch on Not Equal Unsigned) instruction • *MACRO*, 9–48

BOLD keyword
- with MARK • *VAXTPU*, 7–261

BOLD keyword (Cont.)
  with SELECT • *VAXTPU*, 7–337
  with SET (PROMPT_AREA) • *VAXTPU*, 7–446
  with SET (STATUS_LINE) • *VAXTPU*, 7–476
  with SET (VIDEO) • *VAXTPU*, 7–492
"Bold_status" string constant parameter to GET_INFO • *VAXTPU*, 7–221
"Bold_video" string constant parameter to GET_INFO • *VAXTPU*, 7–221
boolean data type • *Routines Intro*, A–2t
Boolean expression • *VAXTPU*, 3–11
Boolean value flag • *Routines Intro*, A–2t
BOOTED processor state • *Device Support (B)*, 1–16
Booting with XDELTA • *Device Support (A)*, 13–1 to 13–5
Boot stack • *Device Support (B)*, 1–15
Bootstrap procedures for XDELTA • *Delta/XDelta*, DELTA–2 to DELTA–8
BOOT_REJECTED processor state • *Device Support (B)*, 1–16
Border
  virtual display • *Programming Resources*, 7–10
BOT (beginning-of-tape)
  See Magnetic tape, BOT marker
/BOTTOM qualifier • *Debugger*, CD–118
Boundary tag • *RTL Library*, 5–8
Bound marker • *VAXTPU*, 2–9 to 2–10
Bound procedure value • *Modular Procedures*, 3–12
"Bound" string constant parameter to GET_INFO • *VAXTPU*, 7–171, 7–185, 7–221
bpi (bits per inch) • *File Applications*, 1–8
BPT (Breakpoint) instruction • *Device Support (A)*, 13–6
BPT (Breakpoint Fault) instruction • *MACRO*, 9–73
Branch access type • *MACRO*, 8–17
Branch instruction
  calculating the location for • *Patch*, PAT–70
  calculating the relative displacement for • *Patch*, PAT–70
Branch mode • *MACRO*, 5–18
  operand specifier format • *MACRO*, 8–29
/BRANCH qualifier • *Debugger*, CD–17, CD–30, CD–131, CD–188, CD–265
BRB (Branch Byte Displacement) instruction • *MACRO*, 9–54
BREAK built-in procedure • *VAXTPU*, 7–39
Breakpoint • *Delta/XDelta*, DELTA–28 to DELTA–31
  canceling • *Debugger*, 3–16, CD–17
  clearing • *Delta/XDelta*, DELTA–28, DELTA–29; *Device Support (A)*, 13–18
  complex • *Delta/XDelta*, DELTA–30; *Device Support (A)*, 13–18
Breakpoint (Cont.)
  defined • *Debugger*, 3–9
  delayed triggering of • *Debugger*, 3–14, CD–131
  displaying • *Debugger*, CD–212
  displaying XDELTA breakpoint list • *Device Support (A)*, 13–18
  DO clause • *Debugger*, 3–14
  exception • *Debugger*, 9–10, CD–130
  initial, in XDELTA • *Delta/XDelta*, DELTA–7
  initial, in XDELTA multiprocessing environment • *Delta/XDelta*, DELTA–8
  in multiprocessing environment • *Delta/XDelta*, DELTA–13, DELTA–35
  on activation (multiprocess program) • *Debugger*, 10–14
  on termination (image exit) • *Debugger*, 10–14
  on vector instruction • *Debugger*, 11–3
  predefined • *Debugger*, 9–10
  proceeding from • *Delta/XDelta*, DELTA–32; *Device Support (A)*, 13–5, 13–18
  proceeding from XDELTA initial • *Delta/XDelta*, DELTA–8
  range for DELTA • *Delta/XDelta*, DELTA–28
  range for XDELTA • *Delta/XDelta*, DELTA–28
  setting • *Debugger*, 3–9, CD–130; *Delta/XDelta*, DELTA–28, DELTA–29
  setting in driver code • *Device Support (A)*, 13–6, 13–10, 13–17
  showing • *Delta/XDelta*, DELTA–28
  simple • *Delta/XDelta*, DELTA–28
  source display at • *Debugger*, 6–7
  WHEN clause • *Debugger*, 3–14
  with DECwindows • *Debugger*, 1–24
  XDELTA restriction on breakpoint 1 • *Delta/XDelta*, DELTA–7
Breakpoint command • *Delta/XDelta*, DELTA–28
BREAKPOINTS parameter • *Device Support (A)*, 13–1, 13–5
"Breakpoint" string constant parameter to GET_INFO • *VAXTPU*, 7–179
Brief image map • *Linker*, 1–12
Brief map • *Linker*, 5–1, LINK–3
  module information in • *Linker*, 5–2, 5–3
  sections in • *Linker*, 5–2
BRIEF prompt • *File Def Language*, FDL–55
/BRIEF qualifier • *Debugger*, CD–223, CD–236; *Linker*, LINK–3
BR level • *Device Support (A)*, 14–33
  relation to SCB vectors • *Device Support (B)*, 1–9
BROADCAST keyword
  with SET (BELL) • *VAXTPU*, 7–355

- Broadcast message • *Programming Resources*, 7–43; *I/O User's I*, 8–18, 8–21, 8–23, 8–46
  - alternate handler • *Programming Resources*, 7–44
  - default handler • *Programming Resources*, 7–43
- BRO option • *File Def Language*, FDL–3
- BRW (Branch Word Displacement) instruction • *MACRO*, 9–54
- BSBB (Branch to Subroutine Byte Displacement) instruction • *MACRO*, 9–55
- BSBW (Branch to Subroutine Word Displacement) instruction • *MACRO*, 9–55
- Bucket • *File Applications*, 3–6, 3–17; *Analyze/RMS_File*, ARMS–2; *File Def Language*, FDL–5, FDL–27
  - defined • *File Applications*, 2–1
  - examining • *Analyze/RMS_File*, ARMS–6
  - fill • *File Def Language*, FDL–28
  - fill percentage • *Convert*, CONV–14
  - list of free • *Convert*, CONV–4
  - reclaiming • *File Applications*, 3–17, 10–30; *Convert*, CONV–1
  - size • *File Applications*, A–1
    - considering performance • *File Applications*, 3–25
    - for indexed files • *File Applications*, 7–20
    - for relative files • *File Applications*, 7–19
    - option • *File Applications*, 4–28
    - relative to index depth • *File Applications*, 3–24
    - with multiple areas • *File Applications*, 3–23
  - split • *Analyze/RMS_File*, ARMS–6
- Bucket boundary • *File Applications*, 3–19; *File Def Language*, FDL–35
  - file organization considerations • *RMS*, 5–4
- Bucket code field
  - See RAB$L_BKT field
- Buckets
  - reclaiming with CONV$RECLAIM routine • *Utility Routines*, CONV–18
  - reclamation statistics • *Utility Routines*, CONV–18
- Bucket size • *File Applications*, A–1
- Bucket size field
  - See FAB$B_BKS field
- Bucket size field in allocation XAB
  - See XAB$B_BKZ field
- Bucket size field in file header characteristics XAB
  - See XAB$B_BKZ field
- Bucket split • *File Applications*, 3–6, 3–22, 9–13, 10–31
  - minimizing • *File Applications*, 3–26; *RMS*, 13–4
- BUCKET_IO attribute • *File Def Language*, FDL–9
- BUCKET_SIZE attribute • *File Def Language*, FDL–6, FDL–18
- BUCKET_SIZE secondary attribute • *File Applications*, 4–28, 7–19, 7–20
- Buffer
  - See also Global buffer
  - allocating • *Device Support (A)*, 1–23, 2–3, 7–6 to 7–7, E–5; *Device Support (B)*, 3–12 to 3–13, 3–14, 3–15, 3–22 to 3–23
  - allocating a physically contiguous • *Device Support (B)*, 3–16
  - attributes • *VAXTPU*, 7–60
  - controlling modification indicator • *VAXTPU*, 7–431
  - converting contents of to string format using STR • *VAXTPU*, 7–520
  - converting name to journal file name • *VAXTPU*, 7–172
  - current • *VAXTPU*, 7–59
  - data area • *Device Support (A)*, 7–7
  - deallocating • *Device Support (A)*, 2–7, 4–20, 7–8; *Device Support (B)*, 3–3, 3–19
  - deleting • *VAXTPU*, 7–107
  - determining if unmodifiable records are present in • *VAXTPU*, 7–175
  - direction
    - current • *VAXTPU*, 7–85
    - setting • *VAXTPU*, 7–379
  - erasing • *VAXTPU*, 2–4, 7–117
  - erasing unmodifiable records from
    - preventing or allowing • *VAXTPU*, 7–375
  - format • *Device Support (A)*, 7–7
  - getting file name of journal • *VAXTPU*, 7–172
  - header area • *Device Support (A)*, 7–7, 7–8
  - I/O • *File Applications*, 7–16
    - size • *File Applications*, 3–2
  - journal file • *VAXTPU*, 1–11
  - key • *File Applications*, 9–13, 9–15, 9–18
  - local • *File Applications*, 3–9, 3–27, 7–20
  - locking • *Device Support (A)*, 1–23, 6–7; *Device Support (B)*, 1–42, 1–43, 3–31 to 3–33, 3–34 to 3–36, 3–40 to 3–42, 3–45 to 3–47, 3–54 to 3–55, 3–58 to 3–60
  - locking multiple areas • *Device Support (B)*, 3–34, 3–45, 3–58
  - margin action settings • *VAXTPU*, 7–414, 7–456
  - margin settings • *VAXTPU*, 7–412, 7–419, 7–454
  - moving data to from system to user • *Device Support (B)*, 3–80 to 3–81
  - moving data to from user to system • *Device Support (B)*, 3–79
  - multiple • *File Applications*, 3–7; *VAXTPU*, 2–4, 7–59

- Buffer (Cont.)
  - number of • *File Applications*, 3–11, 3–26, 3–27
  - record header • *File Applications*, 9–17, 9–18, 9–20
  - recovering contents of • *VAXTPU*, 7–307
  - selecting for optimum performance • *File Applications*, 7–17 to 7–18
  - sensing safe journaling • *VAXTPU*, 7–175
  - sensing unmodifiable records erasable state • *VAXTPU*, 7–169
  - size • *Device Support (A)*, 7–6
  - storing address of • *Device Support (A)*, 7–7
  - tab stops • *VAXTPU*, 7–481
  - testing accessibility of • *Device Support (A)*, 7–6; *Device Support (B)*, 2–39 to 2–40, 3–31 to 3–33, 3–34 to 3–36, 3–40 to 3–42, 3–43 to 3–44, 3–45 to 3–47, 3–54 to 3–55, 3–56 to 3–57, 3–58 to 3–60
  - unlocking • *Device Support (B)*, 3–109
  - user • *File Applications*, 9–17
  - variables • *VAXTPU*, 2–4
  - visible • *VAXTPU*, 7–59
  - VMS RMS space allocation • *File Applications*, 7–17
- Buffer address register • *Device Support (A)*, 14–23
- Buffer area
  - requirement for Get service • *File Applications*, 8–2
- Buffer cache • *File Applications*, 7–5, 7–18
  - for storing index levels • *File Applications*, 7–20
  - types • *File Applications*, 7–20
  - using with multistreaming • *File Applications*, 7–4
- Buffer change journaling • *VAXTPU*, 1–11
  - and keystroke journaling • *VAXTPU*, 7–307
  - converting buffer to journal file name • *VAXTPU*, 7–172
  - default file naming • *VAXTPU*, 1–12
  - enabling • *VAXTPU*, 7–405
  - getting file name of journal • *VAXTPU*, 7–172
  - getting information on journal file • *VAXTPU*, 7–203
  - recovery • *VAXTPU*, 7–307
  - sensing safe state • *VAXTPU*, 7–175
  - sensing the enable • *VAXTPU*, 1–12, 5–10
  - specifying file name • *VAXTPU*, 7–405
- BUFFER command
  - for message buffer • *VAXTPU*, 4–18
- BUFFER data type • *VAXTPU*, 2–3 to 2–4
- Buffer descriptor block
  - See BDB
- Buffered data path • *Device Support (A)*, 14–8; *Device Support (B)*, 1–8
  - See also Data path
  - See BDP
- Buffered data path (Cont.)
  - allocating permanent • *Device Support (A)*, 11–2, 14–18, E–12; *Device Support (B)*, 1–26
  - flow of read operation using • *Device Support (A)*, 14–12 to 14–13
  - flow of write operation using • *Device Support (A)*, 14–12
  - functions • *Device Support (A)*, 14–11
  - odd transfer • *Device Support (B)*, 1–8
  - purging • *Device Support (A)*, 14–14, 14–19, 14–24 to 14–25; *Device Support (B)*, 3–82 to 3–83
  - releasing • *Device Support (A)*, 10–2, 14–19, 14–25; *Device Support (B)*, 2–55, 3–87
  - requesting • *Device Support (A)*, 14–11, 14–17 to 14–18; *Device Support (B)*, 2–60, 3–96 to 3–97
  - rules for using • *Device Support (A)*, 14–11, 14–15
  - speed • *Device Support (A)*, 14–15
- Buffered data path wait queue
  - See Data path wait queue
- Buffered function bit mask • *Device Support (A)*, 4–11, 6–7
- Buffered I/O • *Device Support (A)*, 1–22, 1–23, 2–3, 4–11, 11–7, 16–19; *Device Support (B)*, 1–40, 1–41, 1–79
  - chained • *Device Support (B)*, 1–40
  - complex • *Device Support (B)*, 1–40
  - FDT routines for • *Device Support (A)*, 7–6 to 7–8
  - functions • *Device Support (A)*, 6–4
  - postprocessing • *Device Support (A)*, 7–8; *Device Support (B)*, 3–72
  - reasons for using • *Device Support (A)*, 1–22 to 1–23, 6–7, 6–8
- Buffered I/O byte count quota
  - See BYTLM quota
- Buffered I/O count
  - See BIOCNT
- Buffered I/O limit quota
  - See BIOLM quota
- Buffered I/O operation • *Programming Resources*, 3–20
- Buffered I/O quota • *I/O User's I*, 3–24, 6–13, 7–5
- Buffered read function bit
  - See IRP$V_FUNC
- Buffering mode • *RTL Screen Management*, 2–17
- Buffering technique • *File Applications*, 7–16 to 7–22
- Buffer lock block
  - See BLB
- Buffer names • *VAXTPU*, 2–4
- Buffer overrun
  - with LPA11-K • *I/O User's I*, 4–12

- "Buffer" string constant parameter to GET_INFO • *VAXTPU*, 7–185, 7–193, 7–222
- BUFFER_BEGIN keyword • *VAXTPU*, 7–69, 7–273
  - with POSITION • *VAXTPU*, 7–287
  - with SEARCH • *VAXTPU*, 7–327
  - with SEARCH_QUIETLY • *VAXTPU*, 7–332
- /BUFFER_COUNT qualifier • *File Applications*, 7–19, 7–20
- BUFFER_END keyword • *VAXTPU*, 7–69, 7–273
  - with POSITION • *VAXTPU*, 7–287
  - with SEARCH • *VAXTPU*, 7–327
  - with SEARCH_QUIETLY • *VAXTPU*, 7–332
- Bugcheck • *Device Support (A)*, 13–21
  - BADDALRQSZ • *Device Support (B)*, 3–3, 3–19
  - code • *System Dump Analyzer*, SDA–15
  - examining information regarding • *Device Support (A)*, 13–5
  - fatal conditions • *System Dump Analyzer*, SDA–16 to SDA–20
  - halt/restart • *System Dump Analyzer*, SDA–7
  - handling routines
    - global symbols • *System Dump Analyzer*, SDA–60
  - identifying • *System Dump Analyzer*, SDA–21
  - ILLQBUSCFG • *Device Support (B)*, 1–22
  - INCONSTATE • *Device Support (B)*, 3–88, 3–97
  - information • *Delta/XDelta*, DELTA–8
  - reason • *System Dump Analyzer*, SDA–94
  - SPLACQERR • *Device Support (A)*, 13–28, 13–30, E–18; *Device Support (B)*, 3–111
  - SPLIPLHIGH • *Device Support (A)*, 13–28, E–18; *Device Support (B)*, 3–111, 3–113
  - SPLIPLLOW • *Device Support (A)*, 13–28, E–18; *Device Support (B)*, 3–114, 3–115, 3–116, 3–117
  - SPLRELERR • *Device Support (A)*, 13–29, 13–30, E–18; *Device Support (B)*, 3–114, 3–115
  - SPLRSTERR • *Device Support (A)*, 13–29, 13–30, E–18; *Device Support (B)*, 3–116, 3–117
  - UBMAPEXCED • *Device Support (B)*, 3–74, 3–78
  - UNSUPRTCPU • *Device Support (B)*, 2–10
- BUGL (Bugcheck Longword Message Identifier) instruction • *MACRO*, 9–197
- BUGREBOOT parameter • *Device Support (A)*, 13–2, 13–5, 13–22
- BUGW (Bugcheck Word Message Identifier) instruction • *MACRO*, 9–197
- Building applications on EVE • *VAXTPU*, G–1 to G–12
- Built-in definition
  - function of • *National Char Set*, NCS–7
- Built-in definition (Cont.)
  - _IDENTITY conversion function • *National Char Set*, NCS–8
  - _NATIVE collating sequence • *National Char Set*, NCS–7
- Built-in procedure
  - descriptions • *VAXTPU*, 7–15 to 7–548
  - functions listed • *VAXTPU*, 7–1 to 7–15
  - name of as reserved word • *VAXTPU*, 3–12
  - occluded • *VAXTPU*, 3–12
- Built-in symbol • *Debugger*, C–6, D–2
- Built-in value type • *Command Def*, CDU–6, CDU–24
- Bus
  - device assignments • *Device Support (A)*, 12–10
- Bus grant • *Device Support (A)*, 14–33, 14–34
- Bus request
  - See BR level, BIRQ level
- Busy bit
  - See UCB$V_BSY
- Busy wait • *Modular Procedures*, 3–21
- BVC (Branch on Overflow Clear) instruction • *MACRO*, 9–48
- BVS (Branch on Overflow Set) instruction • *MACRO*, 9–48
- BYPASS privilege • *System Services Intro*, 7–6
- BYTCNT (byte count) quota • *Device Support (A)*, 3–13
  - checking • *Device Support (A)*, E–5
  - crediting • *Device Support (A)*, E–5; *Device Support (B)*, 3–18
  - debiting • *Device Support (A)*, E–5; *Device Support (B)*, 3–12, 3–20 to 3–21, 3–22 to 3–23
  - system maximum • *Device Support (B)*, 3–20, 3–22
  - verifying • *Device Support (B)*, 3–20 to 3–21, 3–22 to 3–23
- Byte • *File Applications*, 1–1
- Byte count quota
  - See BYTCNT
- Byte count register
  - See MBA$L_BCR
- Byte data type • *MACRO*, 8–1
- .BYTE directive • *MACRO*, 6–14
- Byte limit
  - See BYTLM
- BYTE mode • *Patch*, PAT–16
- Byte offset register • *Device Support (A)*, 14–13
- /BYTE qualifier • *Debugger*, CD–62, CD–86
- /BYTE qualifier
  - with ALIGN command • *Patch*, PAT–38

- /BYTE qualifier (Cont.)
  - with DELETE command • *Patch*, PAT–52
  - with DEPOSIT command • *Patch*, PAT–55, PAT–57
  - with EVALUATE command • *Patch*, PAT–59
  - with EXAMINE command • *Patch*, PAT–62
  - with REPLACE command • *Patch*, PAT–71
  - with SET MODE command • *Patch*, PAT–76
  - with VERIFY command • *Patch*, PAT–90
- Byte storage directive (.BYTE) • *MACRO*, 6–14
- byte_signed data type • *Routines Intro*, A–2t
- BYTLM (buffered I/O byte count) quota • *System Services Intro*, 7–3; *File Applications*, 9–8
  - limiting size of user's ACL buffer • *RMS*, 14–3
- BYTLM (byte limit) quota • *Device Support (A)*, 3–13
  - checking • *Device Support (A)*, E–5
  - crediting • *Device Support (A)*, E–5; *Device Support (B)*, 3–18
  - debiting • *Device Support (A)*, E–5; *Device Support (B)*, 3–12, 3–20 to 3–21, 3–22 to 3–23

C

- C
  - See VAX C
  - example in • *RTL Parallel Processing*, 6–14
- Cache
  - buffer • *File Applications*, 7–4
    - for file sharing • *File Applications*, 9–6
    - global • *File Applications*, 7–21
    - specifying as read-only • *File Applications*, 7–22
    - with multiple buffers • *File Applications*, 9–9
  - memory • *File Applications*, 3–12, 3–15, 3–26
    - for file sharing • *File Applications*, 3–14
    - for random processing • *File Applications*, 3–14
    - for storing index • *File Applications*, 3–25
    - process local • *File Applications*, 3–9
    - relative to bucket size • *File Applications*, 3–25
  - tape • *I/O User's I*, 6–8
    - write-back volatile • *I/O User's I*, 6–8
- Cache control block • *Device Support (B)*, 1–83
- Caching • *System Services Intro*, 12–14; *Device Support (B)*, 1–75
- Call
  - testing for successful completion of • *System Services Intro*, 2–16
- Callable interface • *VAXTPU*, 4–1, 7–41
- /CALLABLE_EDT qualifier • *Debugger*, CD–139
- /CALLABLE_LSEDIT qualifier • *Debugger*, CD–139
- /CALLABLE_TPU qualifier • *Debugger*, CD–139
- Callback data structure
  - of widget
    - using in VAXTPU • *VAXTPU*, 7–496
- Callback routines
  - levels of • *VAXTPU*, 4–9
- Callbacks • *VAXTPU*, 4–8 to 4–10
  - handling in EVE • *VAXTPU*, 4–11
- CALL command • *Debugger*, 8–11, CD–10
  - and ASTs • *Debugger*, 9–17, CD–10
  - multiprocess program • *Debugger*, 10–6
  - vectorized program • *Debugger*, 11–23
  - with DECwindows • *Debugger*, 1–8
- Caller access mode • *RMS*, 5–5
- %CALLER_TASK • *Debugger*, D–10
- Call frame • *MACRO*, 9–64
  - condition handler • *Programming Resources*, 9–13
  - displaying in SDA • *System Dump Analyzer*, SDA–79
  - field and buttons in main window
    - with DECwindows • *Debugger*, 1–8, 1–21, 1–22, 1–28
  - following a chain • *System Dump Analyzer*, SDA–79
  - removing from stack • *System Services*, SYS–530
- CALLG (Call Procedure with General Argument List) instruction • *MACRO*, 9–65
  - example • *System Services Intro*, 2–11
  - RTL routine to access • *RTL Library*, LIB–23
  - using MACRO • *System Services Intro*, 2–10
- Calling convention • *RTL Math*, 1–2
- Calling sequence • *Routines Intro*, 2–4; *RMS*, 2–4
- Calling services • *RMS*, 1–1
- Calling standard • *Routines Intro*, 2–1; *RTL Intro*, 1–1, 3–1
- Call-in-progress count • *Modular Procedures*, 3–24
- /CALL qualifier • *Debugger*, CD–17, CD–30, CD–131, CD–188, CD–265
- CALLS (Call Procedure with Stack Argument List) instruction • *MACRO*, 9–67
  - argument • *System Services Intro*, 2–6
  - example • *System Services Intro*, 2–10
  - using MACRO • *System Services Intro*, 2–10
- /CALLS qualifier • *Debugger*, CD–156, CD–253
- Call stack
  - See also Scope
  - displaying • *Debugger*, 2–14, 9–13, CD–214, CD–248
    - with DECwindows • *Debugger*, 1–24
  - removing frame from • *System Services*, SYS–530
  - unwinding • *System Services Intro*, 10–14

Call stack (Cont.)
  used to control instruction display • *Debugger,* 7–10, CD–170
    with DECwindows • *Debugger,* 1–8, 1–22
  used to control source display • *Debugger,* 7–6, CD–170
    with DECwindows • *Debugger,* 1–8, 1–21
  used to control symbol search • *Debugger,* 5–10, CD–170
    with DECwindows • *Debugger,* 1–8, 1–28
CALL_USER built-in procedure • *VAXTPU,* 7–40 to 7–43
CAN$C_CANCEL • *Device Support (A),* 11–8
CAN$C_DASSGN • *Device Support (A),* 11–8
CANCEL ALL command • *Debugger,* CD–15
CANCEL BREAK command • *Debugger,* 3–16, CD–17
Cancel Ctrl/O option
  See RAB$V_CCO option
CANCEL DISPLAY command • *Debugger,* 7–12, CD–20
Cancel I/O bit
  See UCB$V_CANCEL
Cancel I/O routine • *System Dump Analyzer,* SDA–99; *Device Support (A),* 1–4, 9–8, 11–6 to 11–9; *Device Support (B),* 1–30
  address • *Device Support (A),* 6–4, 11–1; *Device Support (B),* 4–4
  context • *Device Support (A),* 11–7 to 11–8; *Device Support (B),* 4–4
  device dependent • *Device Support (A),* 11–9
  device independent • *Device Support (A),* 11–8 to 11–9
  entry point • *Device Support (B),* 4–4
  exit method • *Device Support (B),* 4–5
  flushing ASTs in • *Device Support (B),* 3–4
  for connect to interrupt facility • *Device Support (A),* 19–8, 19–10, 19–18 to 19–19
  input • *Device Support (B),* 4–5
  of CONINTERR.EXE • *Device Support (A),* 19–12, 19–18
  of SCSI third-party class driver • *Device Support (A),* 17–28
  register usage • *Device Support (B),* 4–4
  synchronization requirements • *Device Support (B),* 4–4
  when unneeded • *Device Support (A),* 11–8
CANCEL IMAGE command • *Debugger,* 5–14, CD–22
CANCEL MODE command • *Debugger,* CD–23; *Patch,* PAT–40
CANCEL MODULE command • *Debugger,* 5–7, CD–24; *Patch,* PAT–41
CANCEL PATCH_AREA command • *Patch,* PAT–19, PAT–43
CANCEL RADIX command • *Debugger,* 4–12, CD–26
CANCEL SCOPE command • *Debugger,* 5–12, CD–27; *Patch,* PAT–44
CANCEL SOURCE command • *Debugger,* 6–3, CD–28
CANCEL TRACE command • *Debugger,* 3–16, CD–30
CANCEL TYPE/OVERRIDE command • *Debugger,* 4–26, CD–33
CANCEL WATCH command • *Debugger,* 3–17, CD–34
CANCEL WINDOW command • *Debugger,* 7–15, CD–35
$CANDEF macro • *Device Support (A),* 11–8
Capability field • *RTL Screen Management,* 5–3
  Boolean • *RTL Screen Management,* 5–4
  characters with normal ASCII value • *RTL Screen Management,* 5–15
  creating • *RTL Screen Management,* 5–17
  delimiters • *RTL Screen Management,* 5–3
  nonprinting characters • *RTL Screen Management,* 5–14
  numeric • *RTL Screen Management,* 5–6
  padding • *RTL Screen Management,* 5–15
  string • *RTL Screen Management,* 5–7
  used by SMG • *RTL Screen Management,* 5–22
  user-defined renditions • *RTL Screen Management,* 5–13
Card reader • *Device Support (B),* 1–76
  card punch combinations • *I/O User's I,* 2–1
  026 card reader code • *I/O User's I,* 2–2, 2–8
  029 card reader code • *I/O User's I,* 2–2, 2–8
  code • *I/O User's I,* 2–8
  device characteristics • *I/O User's I,* 2–5
  device driver • *Device Support (A),* 9–6 to 9–8
  driver • *I/O User's I,* 2–1
  end-of-file status • *I/O User's I,* 2–2
  error recovery • *I/O User's I,* 2–3
  failure categories • *I/O User's I,* 2–4
  features • *I/O User's I,* 2–1
  for batch job command procedures • *I/O User's I,* 2–2
  function codes • *I/O User's I,* 2–5, A–2
  function modifiers
    IO$M_BINARY • *I/O User's I,* 2–1, 2–6
    IO$M_PACKED • *I/O User's I,* 2–1, 2–6
  I/O functions
    IO$_READLBLK • *I/O User's I,* 2–6
    IO$_READPBLK • *I/O User's I,* 2–6
    IO$_READVBLK • *I/O User's I,* 2–6

- Card reader
  - I/O functions (Cont.)
    - IO$_SENSEMODE • *I/O User's I*, 2–7
    - IO$_SETCHAR • *I/O User's I*, 2–10
    - IO$_SETMODE • *I/O User's I*, 2–8
  - I/O status block • *I/O User's I*, 2–11
  - read function • *I/O User's I*, 2–6
  - read modes • *I/O User's I*, 2–1
  - sense mode function • *I/O User's I*, 2–7
  - set mode function • *I/O User's I*, 2–7
  - set translation mode • *I/O User's I*, 2–2
  - status returns • *I/O User's I*, A–2
  - supported device • *I/O User's I*, 2–1
  - SYS$GETDVI returns • *I/O User's I*, 2–5
- Carriage control • *Convert*, CONV–2; *Device Support (B)*, 1–74
  - converting formats • *Convert*, CONV–2
  - effect of CARRIAGE_RETURN keyword • *File Def Language*, FDL–33
  - formats listed • *Convert*, CONV–2
  - line printer • *I/O User's I*, 5–6
  - terminal • *I/O User's I*, 8–36
- Carriage control device • *File Def Language*, FDL–33
- Carriage return option
  - See FAB$V_CR option
- CARRIAGE_CONTROL attribute • *File Def Language*, FDL–33
- CARRIAGE_CONTROL secondary attribute • *File Applications*, 4–29
- CARRIAGE_RETURN keyword • *File Def Language*, FDL–33
- Carry condition code (C) • *MACRO*, 8–15
- Case
  - using upper and lower • *Modular Procedures*, A–6
- CASEB (Case Byte) instruction • *MACRO*, 9–56
- CASEL (Case Long) instruction • *MACRO*, 9–56
- CASE macro • *Device Support (B)*, 2–6
  - example • *Device Support (B)*, 2–6
- Case sensitivity • *Debugger*, 9–9
  - of widget names • *VAXTPU*, 7–74
- CASE statement • *VAXTPU*, 3–23 to 3–25
- Case-style error handler • *VAXTPU*, 3–28 to 3–31
- CASEW (Case Word) instruction • *MACRO*, 9–56
- Catchall handler • *Programming Resources*, 9–5, 9–13; *Debugger*, 9–13
- CBT option • *File Def Language*, FDL–6, FDL–18
- CCB$B_AMOD • *Device Support (B)*, 3–103
- CCB$L_UCB • *Device Support (A)*, 4–5
- CCB (channel control block) • *Device Support (A)*, 1–6, 4–5; *Device Support (B)*, 1–11 to 1–12
  - address • *Device Support (B)*, 3–103
- CCB (channel control block) (Cont.)
  - displaying in SDA • *System Dump Analyzer*, SDA–76
- CCO option • *File Def Language*, FDL–14
- C data type declaration • *Routines Intro*, A–25
- CDDB (class driver data block) • *System Dump Analyzer*, SDA–99
- CDROM
  - See Disk
- CDRP (class driver request packet) • *System Dump Analyzer*, SDA–87, SDA–148
- CDT (connection descriptor table) • *System Dump Analyzer*, SDA–87, SDA–148
- CDT argument • *RMS*, B–16
- CDU
  - See Command Definition Utility
- Cell • *Analyze/RMS_File*, ARMS–2; *File Def Language*, FDL–35
  - fixed-length • *File Applications*, 3–12
- CELL AND RECORD structure • *File Applications*, 10–16
- CF keyword
  - description • *National Char Set*, NCS–15
- Chaining • *RTL Library*, 2–5
  - vector instructions • *MACRO*, 10–22
- Change mode handler • *System Services Intro*, 10–6
  - declaring • *System Services*, SYS–123
- CHANGES attribute • *File Def Language*, FDL–26
- CHANGE_CASE built-in procedure • *VAXTPU*, 7–44 to 7–46
- Channel • *RTL Library*, 2–23; *Device Support (A)*, 1–6
  - See also Process I/O channel
  - assigning I/O • *System Services Intro*, 7–14; *System Services*, SYS–23
  - canceling I/O • *System Services*, SYS–39
  - deassigning • *System Services Intro*, 7–20
  - input/output • *Programming Resources*, 7–45
- Channel access mode protection option • *RMS*, 5–5
- Channel access mode subfield
  - See FAB$V_CHAN_MODE option
- Channel control block
  - See CCB
- channel data type • *Routines Intro*, A–2t
- Channel index number • *Device Support (A)*, 4–5, 11–8; *Device Support (B)*, 3–68, 3–103, 4–5
- /CHANNEL qualifier • *System Dump Analyzer*, SDA–131
- Channel request block
  - See CRB
- Channel wait queue
  - See Device controller data channel wait queue

- Character
  - formatting on line printer • *I/O User's I,* 5–2
  - pad • *Convert,* CONV–18
  - terminal terminator • *I/O User's I,* 8–28
- Character case • *Librarian,* LIB–2
- Character-cell measuring system
  - converting to coordinate system • *VAXTPU,* 7–50
- Characteristic
  - See also Device characteristics
  - getting information about
    - asynchronously • *System Services,* SYS–257
    - synchronously • *System Services,* SYS–297
- Character-oriented output • *RTL Screen Management,* 2–8
- Character set • *VAXTPU,* 3–1
  - See DEC Multinational Character Set
  - in source statement • *MACRO,* 3–1
  - special characters • *MACRO,* C–6
  - table • *MACRO,* A–1
  - terminal lowercase • *I/O User's I,* 8–21
- Character string • *Routines Intro,* A–2t
  - See also String
  - data type • *MACRO,* 8–7
  - instructions • *MACRO,* 9–126
  - length • *MACRO,* 6–64
- "Character" string constant parameter to GET_INFO • *VAXTPU,* 7–171
- Character string routine • *RTL Library,* 2–14
  - LIB$CHAR • *RTL Library,* LIB–25
- Character string translation routine • *RTL Library,* 2–14
- Character_cell display • *VAXTPU,* 5–8
- char_string data type • *Routines Intro,* A–2t
- CHECK ECO command • *Patch,* PAT–45, PAT–46
- CHECK NOT ECO command • *Patch,* PAT–47
- /CHECK qualifier • *File Applications,* 10–1; *Analyze/RMS_File,* ARMS–13
  - limitation • *Analyze/RMS_File,* ARMS–14, ARMS–20
  - using with /OUTPUT qualifier • *Analyze/RMS_File,* ARMS–16
  - with wildcard characters • *Analyze/RMS_File,* ARMS–10
- Check report • *File Applications,* 10–1, 10–5
- CHG (change) option
  - in XAB$B_FLG field • *RMS,* B–21
- Child
  - of widget
    - fetching in VAXTPU • *VAXTPU,* 7–210
- Children
  - of widget
    - fetching in VAXTPU • *VAXTPU,* 7–210
- "children" string constant parameter to GET_INFO • *VAXTPU,* 7–210
- CHME (Change Mode to Executive) instruction • *MACRO,* 9–190
- CHMK (Change Mode to Kernel) instruction • *Device Support (A),* 4–1; *MACRO,* 9–190
- CHMS (Change Mode to Supervisor) instruction • *MACRO,* 9–190
- CHMU (Change Mode to User) instruction • *MACRO,* 9–190
- CIF option • *File Def Language,* FDL–19
- C implementation table • *Routines Intro,* A–25
- $CINDEF macro • *Device Support (A),* 19–10
- Circumflex (^) • *Debugger,* 4–8, D–5
- Class
  - of widget
    - fetching in VAXTPU • *VAXTPU,* 7–214
  - of widget resource
    - fetching in VAXTPU • *VAXTPU,* 7–215
- Class driver • *Device Support (A),* 17–4
  - See Terminal class driver
  - SCSI template • *Device Support (A),* 17–9
- Class driver data block
  - See CDBB
- Class driver entry vector table • *Device Support (B),* 1–34
- Class driver request packet
  - See CDRP
- Class driver vector table • *Device Support (A),* 18–5 to 18–6; *Device Support (B),* 1–89
  - address • *Device Support (A),* 18–9; *Device Support (B),* 2–8
  - relocating • *Device Support (B),* 2–7
- "class" string constant parameter to GET_INFO • *VAXTPU,* 7–214
- CLASS_CTRL_INIT macro • *Device Support (A),* 18–12; *Device Support (B),* 1–89, 2–7
- CLASS_DDT vector table entry • *Device Support (A),* 18–19
- CLASS_DISCONNECT service routine • *Device Support (A),* 18–19
- CLASS_DS_TRANS service routine • *Device Support (A),* 18–13, 18–20
- CLASS_FORK service routine • *Device Support (A),* 18–14, 18–20
- CLASS_GETNXT service routine • *Device Support (A),* 18–20, 18–21; *Device Support (B),* 1–89, 2–8
  - address • *Device Support (A),* 18–9

CLASS_POWERFAIL service routine • *Device Support (A)*, 18–13, 18–22
CLASS_PUTNXT service routine • *Device Support (A)*, 18–18, 18–21; *Device Support (B)*, 1–89, 2–8
  address • *Device Support (A)*, 18–9
CLASS_READERROR service routine • *Device Support (A)*, 18–18, 18–22
CLASS_SETUP_UCB service routine • *Device Support (A)*, 18–12, 18–22
CLASS_SET_LINE service routine • *Device Support (A)*, 18–13
CLASS_UNIT_INIT macro • *Device Support (A)*, 18–9, 18–12, 18–19; *Device Support (B)*, 2–8
Clauses
  summary of • *Command Def*, CDU–19 to CDU–22
/CLEAR qualifier • *Debugger*, CD–70
CLI$DCL_PARSE • *Command Def*, CDU–17, CDU–46
CLI$DCL_PARSE routine • *Utility Routines*, CLI–6
CLI$DISPATCH • *Command Def*, CDU–17, CDU–46
CLI$DISPATCH routine • *Utility Routines*, CLI–9
CLI$GET_VALUE • *Command Def*, CDU–17, CDU–45, CDU–46
CLI$GET_VALUE routine • *Utility Routines*, CLI–10
CLI$PRESENT • *Command Def*, CDU–17, CDU–45, CDU–46
CLI$PRESENT routine • *Utility Routines*, CLI–13
CLI (command language interpreter) • *Command Def*, CDU–1; *RTL Library*, 2–2
CLI access routine • *RTL Library*, 2–2
Client message
  designating routine to handle • *VAXTPU*, 7–357
  fetching action routine for handling • *VAXTPU*, 7–197
  finding out type of • *VAXTPU*, 7–197
  sending from VAXTPU • *VAXTPU*, 7–344
CLIENT_MESSAGE
  keyword parameter to SET built-in procedure • *VAXTPU*, 7–357
"client_message" string constant parameter to GET_INFO • *VAXTPU*, 7–197
"client_message_routine" string constant parameter to GET_INFO • *VAXTPU*, 7–197
Clipboard
  fetching data from • *VAXTPU*, 7–149
  overview of • *VAXTPU*, 7–149
  reading data from • *VAXTPU*, 7–295
  writing data to • *VAXTPU*, 7–540
CLI routines • *Command Def*, CDU–1
  See also Command string
CLI routines (Cont.)
  example of use in FORTRAN program • *Utility Routines*, CLI–2
  introduction • *Utility Routines*, CLI–1
  list of • *Utility Routines*, CLI–1
  types of • *Command Def*, CDU–17
  use of • *Command Def*, CDU–45, CDU–46
  when to use • *Utility Routines*, CLI–1
CLI symbol • *RTL Library*, LIB–343
  deleting • *RTL Library*, LIB–116
  getting value of • *RTL Library*, LIB–219
  RTL routines • *RTL Library*, LIB–116, LIB–219
Clock
  See also Interval clock
  setting system • *System Services Intro*, 9–8
Clock rate
  with LPA11-K • *I/O User's I*, 4–10
Cloned UCB routine • *Device Support (A)*, 11–12 to 11–13; *Device Support (B)*, 1–78
  address • *Device Support (A)*, 6–4; *Device Support (B)*, 1–31, 4–6
  context • *Device Support (B)*, 4–6
  exit method • *Device Support (A)*, 11–13; *Device Support (B)*, 4–7
  input • *Device Support (A)*, 11–12; *Device Support (B)*, 4–6
  register usage • *Device Support (A)*, 11–12; *Device Support (B)*, 4–6
  synchronization requirements • *Device Support (B)*, 4–6
Close Current Location, Open Next command • *Delta/XDelta*, DELTA–22
Close service
  condition values • *RMS*, RMS–5
    See also Completion status code
  contrasted with Disconnect service • *RMS*, 4–5
  control block input fields • *RMS*, RMS–4
  control block output fields • *RMS*, RMS–4
  function • *RMS*, 4–1
  introduction • *RMS*, 4–1
  limitations with XABs • *RMS*, RMS–4
  use restrictions • *RMS*, RMS–4
Closures • *VAXTPU*, 4–11
CLRB (Clear Byte) instruction • *MACRO*, 9–14
CLRD (Clear D_floating) instruction • *MACRO*, 9–108
CLRF (Clear F_floating) instruction • *MACRO*, 9–108
CLRG (Clear G_floating) instruction • *MACRO*, 9–108
CLRH (Clear H_floating) instruction • *MACRO*, 9–108
CLRL (Clear Long) instruction • *MACRO*, 9–14
CLRO (Clear Octa) instruction • *MACRO*, 9–14
CLRQ (Clear Quad) instruction • *MACRO*, 9–14
CLRW (Clear Word) instruction • *MACRO*, 9–14

CLUB (cluster block) • *System Dump Analyzer,* SDA–83
CLUDCB (cluster quorum disk control block) • *System Dump Analyzer,* SDA–83
CLUFCB (cluster failover control block) • *System Dump Analyzer,* SDA–83
Cluster
  See also VAXcluster
  creation of • *Linker,* 1–7, 1–8, 3–6, 6–8, 6–11
  current • *Linker,* 6–12
  default • *Linker,* 6–9
  empty • *Linker,* 6–11
  for transfer vector • *Linker,* 4–8
  in a based image • *Linker,* 1–7, 3–5
  memory allocation for • *Linker,* 6–15
  order of processing • *Linker,* 6–9, 6–12
  protection of • *Linker,* 1–8, 3–10
  shareable image • *Linker,* 6–7
Cluster-based shareable image • *Linker,* 6–15
Cluster-based user • *Linker,* 6–15
Cluster block
  See CLUB
Cluster failover control block
  See CLUFCB
Clustering algorithm • *Linker,* 6–8
Cluster management code
  global symbols • *System Dump Analyzer,* SDA–60
CLUSTER option • *Programming Resources,* 5–6
  See also Linker Utility
Cluster quorum disk control block
  See CLUDCB
Cluster system block
  See CSB
Cluster system identification number
  See CSID
CLUSTER_SIZE attribute • *File Def Language,* FDL–18
CLUSTRLOA.STB • *System Dump Analyzer,* SDA–60
CLUSTRLOA symbol • *System Dump Analyzer,* SDA–13
CMEXEC privilege
  for analyzing VAX RMS Journaling files • *Analyze/RMS_File,* ARMS–11
CMI (CPU-to-memory interconnect) • *Device Support (A),* 1–11
CMPB (Compare Byte) instruction • *MACRO,* 9–15
CMPC3 (Compare Characters 3 Operand) instruction • *MACRO,* 9–128
CMPC5 (Compare Characters 5 Operand) instruction • *MACRO,* 9–128
CMPD (Compare D_floating) instruction • *MACRO,* 9–109
CMPF (Compare F_floating) instruction • *MACRO,* 9–109
CMPG (Compare G_floating) instruction • *MACRO,* 9–109
CMPH (Compare H_floating) instruction • *MACRO,* 9–109
CMPL (Compare Long) instruction • *MACRO,* 9–15
CMPP3 (Compare Packed 3 Operand) instruction • *MACRO,* 9–152
CMPP4 (Compare Packed 4 Operand) instruction • *MACRO,* 9–152
CMPV (Compare Field) instruction • *MACRO,* 9–38
CMPW (Compare Word) instruction • *MACRO,* 9–15
CMPZV (Compare Zero Extended Field) instruction • *MACRO,* 9–38
CMS (Code Management System)
  See VAX DEC/CMS
Coarse granularity • *RTL Parallel Processing,* 5–1
COBOL
  See VAX COBOL
COBOL data type declaration • *Routines Intro,* A–28
COBOL implementation table • *Routines Intro,* A–28
COBOL intermediate temporary data type • *Routines Intro,* 2–20
Code
  See also Instruction
  AST-reentrant • *Modular Procedures,* 3–19
  fully reentrant • *Modular Procedures,* 3–19
  maintaining readability • *Modular Procedures,* 3–7
  position-independent • *Modular Procedures,* 3–1
  writing AST-reentrant procedures • *Modular Procedures,* 3–20
Code address expression
  selecting from window
   with DECwindows • *Debugger,* 1–23
Code Management System (CMS)
  See VAX DEC/CMS
Coding conventions
  See Device driver
Coding guidelines • *Modular Procedures,* 3–1
Collating key data type • *RMS,* 13–6
Collating sequence
  creating
   limitation • *National Char Set,* NCS–9
   using appended • *National Char Set,* NCS–9
   using modified • *National Char Set,* NCS–9
   using name of existing • *National Char Set,* NCS–8
   using reordered • *National Char Set,* NCS–10

Collating sequence
  creating (Cont.)
    using reversed • *National Char Set*, NCS–10
    using series of expressions • *National Char Set*, NCS–8
  expression forms listed • *National Char Set*, NCS–8
  MODIFICATIONS keyword clause formats listed • *National Char Set*, NCS–17
Collating sequence name field
  See XAB$L_COLNAM field
Collating sequence size field
  See XAB$L_COLSIZ field
Collating sequence table field
  See XAB$L_COLTBL field
COLLATING_SEQUENCE attribute • *File Def Language*, FDL–27
Colon (:)
  in label field • *MACRO*, 2–2
  range delimiter • *Debugger*, 4–18, 11–4, 11–6, 11–7, CD–85
COLUMN_MOVE_VERTICAL keyword • *VAXTPU*, 7–359
"Column_move_vertical" string constant parameter to GET_INFO • *VAXTPU*, 7–206
COM$DELATTNAST • *Device Support (B)*, 3–2
COM$DRVDEALMEM • *Device Support (A)*, 16–21; *Device Support (B)*, 3–3
COM$FLUSHATTNS • *Device Support (B)*, 3–4, 3–6
COM$POST • *Device Support (A)*, 7–5; *Device Support (B)*, 3–5, 4–2
COM$POST_NOCNT • *Device Support (B)*, 3–5
COM$SETATTNAST • *Device Support (B)*, 3–6 to 3–7
[command • *Delta/XDelta*, DELTA–16
/ command • *Delta/XDelta*, DELTA–17, DELTA–25
! command • *Delta/XDelta*, DELTA–20
' command • *Delta/XDelta*, DELTA–37
= command • *Delta/XDelta*, DELTA–42
Command • *System Dump Analyzer*, SDA–10 to SDA–14
  See SCSI command
  for Analyze/RMS_File Utility • *File Applications*, 10–11
  for EDIT/FDL • *File Applications*, 4–3
  interactive • *Analyze/RMS_File*, ARMS–21
  list of commands • *Delta/XDelta*, DELTA–15
Command address register
  See MBA$L_CAR
Command chaining • *I/O User's II*, 4–2
Command definition file • *Command Def*, CDU–4
Command definition file (Cont.)
  changing syntax • *Command Def*, CDU–5 to CDU–6
  creating • *Command Def*, CDU–4 to CDU–14
  defining verbs in • *Command Def*, CDU–8 to CDU–9
  for sample program • *Command Def*, CDU–45, CDU–46
  processing • *Command Def*, CDU–14 to CDU–16
  statements in • *Command Def*, CDU–19 to CDU–37
Command Definition Language statements • *Command Def*, CDU–5
Command Definition Utility (CDU) • *Command Def*, CDU–1
  CDU command • *Programming Resources*, 1–16
  creating command table • *Programming Resources*, 1–17
  defining commands • *Programming Resources*, 1–16
  directing output from • *Command Def*, CDU–18
  exiting • *Command Def*, CDU–18
  format • *Command Def*, CDU–18
  invoking • *Command Def*, CDU–18
  modifying command table • *Programming Resources*, 1–16
  overview • *Command Def*, CDU–18
  parsing commands • *Programming Resources*, 1–17
Command descriptions • *Patch*, PAT–38 to PAT–91
Command file • *VAXTPU*, 4–29 to 4–31
  debugging • *VAXTPU*, 4–34
  default • *VAXTPU*, 4–21
  definition • *VAXTPU*, 1–10
  running SUMSLP from a • *SUMSLP*, SUM–12
  sample • *VAXTPU*, 4–30
Command format
  debugger • *Debugger*, CD–3
Command interface
  COMMAND box, DECwindows • *Debugger*, 1–20, 1–28
  debugger • *Debugger*, 2–1
    with DECwindows • *Debugger*, 1–28, 1–35
Command language interpreter
  See CLI
Command language routines
  See CLI routines
Command line
  DCL
    determining whether /RECOVER specified on • *VAXTPU*, 7–408
  fetching values from • *VAXTPU*, 7–176, 7–177

Command line (Cont.)
/JOURNAL command qualifier • *VAXTPU,* 1–11, 1–12
/NOJOURNAL command qualifier • *VAXTPU,* 1–12
/RECOVER command qualifier • *VAXTPU,* 1–11, 7–307
Command packet • *I/O User's II,* 4–4
Command procedure
See also Initialization file, debugger
creating
using CREATE command • *Patch,* PAT–4, PAT–48
creating using text editor • *Patch,* PAT–5
debugger • *Debugger,* 8–1
default directory for • *Debugger,* CD–129, CD–211
displaying commands in • *Debugger,* CD–159
exiting • *Debugger,* CD–7, CD–94, CD–112
file specification • *Patch,* PAT–48
invoking • *Debugger,* CD–7
log file as • *Debugger,* 8–5
passing parameters to • *Debugger,* 8–2, CD–44
processing selected patches in • *Patch,* PAT–33 to PAT–34
recreating displays with • *Debugger,* 7–22, CD–101
using DEFINE command in • *Patch,* PAT–5
using symbolic references in • *Patch,* PAT–4 to PAT–6
using user-defined symbols in • *Patch,* PAT–5
with DECwindows • *Debugger,* 1–29
Command processing • *Linker,* 6–8
See also DCL
/COMMAND qualifier • *Debugger,* 8–6, CD–47; *VAXTPU,* 4–25, 5–3 to 5–4, 5–6 to 5–7
Command string • *Command Def,* CDU–1 to CDU–2
See also CLI routines
action routine • *Utility Routines,* CLI–9
checking for presence of command string entities • *Utility Routines,* CLI–13
dispatching to action routine • *Utility Routines,* CLI–9
keyword path • *Utility Routines,* CLI–13
labels
list of label names • *Utility Routines,* CLI–12
obtaining values of command string entities • *Utility Routines,* CLI–10
parsing a DCL command string • *Utility Routines,* CLI–6
positional qualifiers • *Utility Routines,* CLI–14
processing with CLI routines • *Utility Routines,* CLI–1
prompting for input • *Utility Routines,* CLI–7
Command string (Cont.)
symbol substitution • *Utility Routines,* CLI–6
"Command" string constant parameter to GET_INFO • *VAXTPU,* 7–176
Command synonyms • *VAXTPU,* G–5 to G–7
Command table
adding commands to • *Command Def,* CDU–15, CDU–43
creating a new • *Command Def,* CDU–16
creating an object module for • *Command Def,* CDU–4
deleting commands from • *Command Def,* CDU–15, CDU–39
input • *Command Def,* CDU–44
listing file for • *Command Def,* CDU–40
object module for • *Command Def,* CDU–16, CDU–41
output file • *Command Def,* CDU–42
process • *Command Def,* CDU–2
system • *Command Def,* CDU–2
with CLI routines • *Utility Routines,* CLI–1, CLI–7
Command verb
See DEFINE VERB statement
Command window
in EVE editor • *VAXTPU,* 4–16
"Command_file" string constant parameter to GET_INFO • *VAXTPU,* 7–176
Comment
block • *Modular Procedures,* 3–9, A–6
character • *File Def Language,* FDL–40
delimiters • *Modular Procedures,* 3–9
entering a • *Patch,* PAT–23
format • *Debugger,* CD–4
in FDL files • *File Def Language,* FDL–40
Comment character • *VAXTPU,* 1–5
COMMENT keyword
with LOOK_UP_KEY • *VAXTPU,* 7–254
Comment lines
in help files • *Librarian,* LIB–6
Comment separator • *RMS,* 3–6
use in VMS RMS coding • *RMS,* 3–6
Common block • *Programming Resources,* 3–6
aligning • *Programming Resources,* 8–4
installing as a shared image • *Programming Resources,* 5–13
interprocess • *Programming Resources,* 5–13
modifying • *Programming Resources,* 3–6
per-process • *Programming Resources,* 3–6
Common Data Dictionary • *Programming Resources,* 1–8, 1–9, 1–10
Common event flag cluster • *System Services Intro,* 4–4

- Common event flag cluster (Cont.)
  - permanent • *Programming Resources*, 4–5
  - temporary • *Programming Resources*, 4–4
- Common source files • *Modular Procedures*, 3–7, A–6
  - declarations • *Modular Procedures*, 3–7
- Communication
  - intersystem • *Programming Resources*, 3–26
- Compact Disc Read-Only Memory (CDROM)
  - See Disk
- Compatibility mode handler • *System Services Intro*, 10–6
  - declaring • *System Services*, SYS–123
- Compilation
  - conditional • *VAXTPU*, 3–36
- COMPILE built-in procedure • *VAXTPU*, 4–19, 7–47 to 7–49
- Compiler
  - compiler generated type • *Debugger*, 4–4
  - /DEBUG qualifier • *Debugger*, 5–2, 6–1
    - with DECwindows • *Debugger*, 1–3
  - /LIST qualifier • *Debugger*, 6–1
  - /NOOPTIMIZE qualifier • *Debugger*, 5–2, 9–1
    - with DECwindows • *Debugger*, 1–3
- Compiler limits • *VAXTPU*, 7–47
- Compilers • *Programming Resources*, 1–5 to 1–11
- Compiling
  - in a VAXTPU buffer • *VAXTPU*, 4–19
  - in EVE editor • *VAXTPU*, 4–19
  - programs • *VAXTPU*, 4–18 to 4–19
  - to create section file • *VAXTPU*, 4–24
- Complement operator • *MACRO*, 3–14
- Completion routine
  - condition for AST execution • *RMS*, 3–11
  - service macro arguments • *RMS*, 3–11
- Completion status code
  - description • *RMS*, 2–5, A–9 to A–20
  - errors for inaccessible control block condition • *RMS*, 2–6
  - handling • *RMS*, 3–12
  - hexadecimal values • *RMS*, A–2 to A–9
  - listing conditions when not returned • *RMS*, A–2
  - severity codes • *RMS*, 2–6
  - testing • *RMS*, 2–5
- Completion status code field
  - use with debugger • *RMS*, A–2
- Completion status code field in FAB
  - See FAB$L_STS field
- Completion status code field in RAB
  - See RAB$L_STS field
- Completion status code value field
  - use with debugger • *RMS*, A–2
- Completion status field
  - as alternative to use of R0 • *RMS*, 2–4
  - for signaling errors • *RMS*, 2–6
- Completion status value field • *File Applications*, 5–12
  - as alternative to use of R0 • *RMS*, 2–4
  - for signaling errors • *RMS*, 2–6
- Completion status value field in FAB
  - See FAB$L_STV field
- Completion status value field in RAB
  - See RAB$L_STV field
- Complex breakpoint • *Delta/XDelta*, DELTA–30
- Complex number • *RTL Math*, 1–4, MTH–57, MTH–59, MTH–110, MTH–120
  - absolute value of • *RTL Math*, MTH–23
  - complex exponential of • *RTL Math*, MTH–31, MTH–33
  - conjugate of • *RTL Math*, MTH–44, MTH–45
  - cosine of • *RTL Math*, MTH–26, MTH–28
  - division of • *RTL General Purpose*, OTS–40
  - made from floating-point • *RTL Math*, MTH–40, MTH–42
  - multiplication of • *RTL General Purpose*, OTS–53
  - natural logarithm of • *RTL Math*, MTH–35, MTH–37
  - sine of • *RTL Math*, MTH–53, MTH–54
- complex_number data type • *Routines Intro*, A–3t
- Component • *Routines Intro*, A–8t
- Composed input
  - See also Key table
  - terminating • *Programming Resources*, 7–28
- Composition operations • *RTL Screen Management*, 2–1
- Compression • *File Def Language*, FDL–5, FDL–28
  - negative values • *File Def Language*, FDL–4
  - of data record • *File Def Language*, FDL–27
  - within data record • *File Def Language*, FDL–4
  - within primary key • *File Def Language*, FDL–4, FDL–27
- /COMPRESS qualifier • *Librarian*, LIB–15; *National Char Set*, NCS–24
  - See also /DATA qualifier
  - See also /SQUEEZE qualifier
  - using with /OUTPUT • *Librarian*, LIB–36
- CONCATENATE clause
  - for VALUE clause • *Command Def*, CDU–24, CDU–33
- Concatenating input files • *Convert*, CONV–5
- Concatenation
  - pattern (+) • *VAXTPU*, 2–15
  - string • *VAXTPU*, 3–4
- Concealed logical name • *File Applications*, 5–7

- Conditional assembly block directive
  - .ENDC • *MACRO*, 6–26
  - (.IF) • *MACRO*, 6–40
  - listing unsatisfied code • *MACRO*, 6–89
- Conditional compilation • *VAXTPU*, 3–36
- Conditional statements • *VAXTPU*, 3–22 to 3–23
- Condition code • *Programming Resources*, 9–1; *MACRO*, 8–14, 9–4
  - carry (C) • *MACRO*, 8–15
  - chaining • *Programming Resources*, 9–23
  - defining • *Programming Resources*, 9–7
  - modifying • *Programming Resources*, 9–20
  - negative (N) • *MACRO*, 8–15
  - overflow (V) • *MACRO*, 8–15
  - signaling • *Programming Resources*, 9–5
  - SS$_EXQUOTA • *Programming Resources*, 9–3
  - SS$_NOPRIV • *Programming Resources*, 9–3
  - zero (Z) • *MACRO*, 8–15
- Condition code and message • *Programming Resources*, 9–1
- Condition handler • *Routines Intro*, 1–12, 2–45; *RTL Library*, 4–12
  - See also Signal argument vector
  - argument list • *System Services Intro*, 10–8
  - arithmetic • *Programming Resources*, 9–26
  - call frame • *Programming Resources*, 9–13
  - catchall • *Programming Resources*, 9–13; *RTL Library*, 4–14
  - condition code • *Programming Resources*, 9–16
  - continuing execution of • *RTL Library*, 4–21
  - course of action • *System Services Intro*, 10–12
  - debugging • *Programming Resources*, 9–20; *Debugger*, 9–10
  - default • *Routines Intro*, 2–51; *RTL Library*, 4–13
  - deleting • *Routines Intro*, 2–47
  - establishing • *Programming Resources*, 9–14; *Routines Intro*, 2–46
  - establishment of • *RTL Library*, 4–20, LIB–140
  - example • *System Services Intro*, 10–13
  - exceptions • *Routines Intro*, 1–12, 2–45
  - exit • *Routines Intro*, A–5t
  - exiting • *Programming Resources*, 9–17
  - interaction between default and user-supplied handlers • *RTL Library*, 4–15
  - last-chance • *RTL Library*, 4–14
  - last-chance exception vector • *Programming Resources*, 9–13
  - mechanism array • *Programming Resources*, 9–15
  - memory
    - use of • *Routines Intro*, 2–51
  - multiple active signals • *Routines Intro*, 2–54
  - operations involving • *Routines Intro*, 2–46
- Condition handler (Cont.)
  - options • *Routines Intro*, 2–45
  - parameters and invocation • *Routines Intro*, 2–49
  - primary exception vector • *Programming Resources*, 9–13
  - properties of • *Routines Intro*, 2–49
  - register values • *Routines Intro*, 2–53
  - request to unwind • *Routines Intro*, 2–52
  - resignaling • *RTL Library*, 4–21
  - returning from • *Routines Intro*, 2–52
  - searching for • *Programming Resources*, 9–12
  - secondary exception vector • *Programming Resources*, 9–13
  - signal array • *Programming Resources*, 9–14
  - software supplied • *RTL Library*, 4–13
  - specifying • *System Services Intro*, 10–6
  - stack usage • *Routines Intro*, 2–46
  - traceback • *Programming Resources*, 9–13; *RTL Library*, 4–13
  - unwinding • *RTL Library*, 4–22
  - use of • *Programming Resources*, 9–13, 9–20
  - user-supplied • *RTL Library*, 4–13
  - writing • *Programming Resources*, 9–14
  - writing of • *RTL Library*, 4–20
- Condition handling • *RTL Math*, 1–3; *RTL Library*, 4–2
  - See also Condition handler
  - See also Condition Handling Facility
  - See also Condition value
  - See also Exception
  - See also Exception condition
  - See also Message Utility
  - at AST level • *Modular Procedures*, 3–26
  - continuing • *RTL Library*, 4–14
  - default • *Programming Resources*, 9–5
  - displaying messages • *RTL Library*, 4–16
  - logging error messages • *RTL Library*, 4–4
  - logging error messages to a file • *RTL Library*, 4–27
  - resignaling • *Programming Resources*, 9–18; *RTL Library*, 4–14
  - return status • *Programming Resources*, 9–3
  - signal • *Programming Resources*, 9–5
  - stack traceback • *RTL Library*, 4–3
  - stack unwind • *RTL Library*, 4–4, 4–14
  - unwinding • *Programming Resources*, 9–18
  - user-defined messages • *RTL Library*, 4–4
  - vector processor • *Routines Intro*, 2–51
- Condition Handling Facility • *RTL Library*, 4–19
  - defined • *RTL Library*, 4–1
  - function of • *RTL Library*, 4–2

Condition-handling routines
  global symbols • *System Dump Analyzer*, SDA–60
Condition-handling services • *System Services Intro*, 1–2, 10–1
Condition Handling Standard • *Routines Intro*, 2–44
Condition value • *Modular Procedures*, 3–3; *Routines Intro*, A–4t; *System Services Intro*, 1–5, 1–9, 2–15; *RTL Intro*, 3–6, 3–15; *RTL Library*, 4–5 to 4–7, 4–24, LIB–272
  See also Completion status code
  definition of • *Routines Intro*, 2–3
  description of • *Routines Intro*, 2–8
  evaluating • *System Dump Analyzer*, SDA–48
  examining • *System Dump Analyzer*, SDA–51
  field
    cntrl • *Routines Intro*, 2–9
    condition identification • *Routines Intro*, 2–8
    facility • *Routines Intro*, 2–9
    message number • *Routines Intro*, 2–9
    severity code • *Routines Intro*, 2–9
  high-level language • *System Services Intro*, 2–18
  information provided by • *System Services Intro*, 2–16
  interpreting severity codes • *Routines Intro*, 2–10
  registers
    use of • *Routines Intro*, 2–12
  returned • *Routines Intro*, 1–14
    in I/O status block • *Routines Intro*, 1–14
    in mailbox • *Routines Intro*, 1–14
    in R0 • *Routines Intro*, 1–5
    signaled in register • *Routines Intro*, 1–7, 1–15
  severity • *RTL Library*, 4–6
  signaled • *Routines Intro*, 1–7, 1–15
  symbols for • *Routines Intro*, 2–9
  testing • *System Services Intro*, 2–16
  use of • *Routines Intro*, 2–11
Condition values returned heading • *Routines Intro*, 1–12
/CONDITION_VALUE qualifier • *Debugger*, CD–81, CD–86; *System Dump Analyzer*, SDA–48
cond_value data type • *Routines Intro*, A–4t
Configuration register
  See CSR
  See MBA$L_CSR
CONFREGL array • *Device Support (A)*, 16–7
CONINTERR.EXE • *Device Support (A)*, 19–8, 19–13
  cancel I/O routine of • *Device Support (A)*, 19–12
  connecting to • *Device Support (A)*, 19–9
Conjugate of complex number • *RTL Math*, MTH–44, MTH–45
CONNECT attribute • *File Def Language*, FDL–2, FDL–8
CONNECT command • *Debugger*, 10–4, 10–16, CD–36; *I/O User's I*, 8–17
  See System Generation Utility
Connection • *Device Support (A)*, 17–5, 17–9
  breaking • *Device Support (B)*, 2–73
  displaying SDA information • *System Dump Analyzer*, SDA–87, SDA–123, SDA–148
  obtaining characteristics of • *Device Support (B)*, 2–75 to 2–76
  requesting • *Device Support (A)*, 17–26; *Device Support (B)*, 2–70 to 2–71
  setting characteristics of • *Device Support (B)*, 2–88 to 2–89
Connection characteristics buffer • *Device Support (B)*, 2–88
Connection descriptor table
  See CDT
Connection manager
  displaying SDA information • *System Dump Analyzer*, SDA–82
/CONNECTION qualifier • *System Dump Analyzer*, SDA–148
CONNECT primary attribute
  ASYNCHRONOUS secondary attribute • *File Applications*, 9–9, 9–15, 9–18, 9–19, 9–20
  DELETE_ON_CLOSE secondary attribute • *File Applications*, 9–12
  END_OF_FILE secondary attribute • *File Applications*, 9–10
  FAST_DELETE secondary attribute • *File Applications*, 9–9, 9–12, 9–20
  FILL_BUCKETS secondary attribute • *File Applications*, 9–13, 9–18
  GLOBAL_BUFFER_COUNT secondary attribute • *File Applications*, 9–9
  KEY_GREATER_EQUAL attribute • *File Applications*, 8–9
  KEY_GREATER_EQUAL secondary attribute • *File Applications*, 9–12, 9–15
  KEY_GREATER_THAN attribute • *File Applications*, 8–9, 8–10
  KEY_GREATER_THAN secondary attribute • *File Applications*, 9–13, 9–15
  KEY_LIMIT secondary attribute • *File Applications*, 9–13, 9–16
  KEY_OF_REFERENCE secondary attribute • *File Applications*, 9–13, 9–15
  LOCATE_MODE secondary attribute • *File Applications*, 9–9, 9–16
  LOCK_ON_READ secondary attribute • *File Applications*, 7–11, 9–16
  LOCK_ON_WRITE secondary attribute • *File Applications*, 7–11, 9–16, 9–18

CONNECT primary attribute (Cont.)
MANUAL_LOCKING secondary attribute • *File Applications*, 9–16
MANUAL_UNLOCKING secondary attribute • *File Applications*, 7–15
MULTIBLOCK_COUNT secondary attribute • *File Applications*, 3–11, 7–18, 9–9
MULTIBUFFER_COUNT secondary attribute • *File Applications*, 3–11, 3–13, 3–26, 7–17, 7–18, 7–19, 7–20, 9–9
NOLOCK secondary attribute • *File Applications*, 7–11, 9–15
NONEXISTENT_RECORD attribute • *File Applications*, 8–9
NONEXISTENT_RECORD secondary attribute • *File Applications*, 7–15, 9–16
READ_AHEAD secondary attribute • *File Applications*, 9–9, 9–16
READ_REGARDLESS secondary attribute • *File Applications*, 7–12, 9–16
TIMEOUT_PERIOD secondary attribute • *File Applications*, 7–12, 9–17, 9–19
TRUNCATE_ON_PUT secondary attribute • *File Applications*, 9–11, 9–19
UPDATE_IF attribute • *File Applications*, 8–8
UPDATE_IF secondary attribute • *File Applications*, 9–11, 9–19
WAIT_FOR_RECORD secondary attribute • *File Applications*, 7–12, 9–17
WRITE_BEHIND secondary attribute • *File Applications*, 9–10, 9–19
Connect service • *File Applications*, 8–5; *RMS*, RMS–6
and asynchronous operations • *File Applications*, 8–18
and next record • *File Applications*, 8–15, 8–16
comparing positioning for various file organizations • *RMS*, RMS–7
condition values • *RMS*, RMS–9
connecting record stream • *RMS*, 4–4
control block input fields • *RMS*, RMS–7
control block output fields • *RMS*, RMS–8
effect on next-record position • *File Applications*, 8–16
program example • *RMS*, 4–12
use with multiple keys • *RMS*, 4–12
Connect to interrupt driver
See CONINTERR.EXE
Connect to interrupt facility
cancel I/O routine • *Device Support (A)*, 19–18 to 19–19
condition values returned • *Device Support (A)*, 19–11
Connect to interrupt facility (Cont.)
CONNECT command • *Device Support (A)*, 19–9
example of A/D converter using • *Device Support (A)*, 19–19, 19–21 to 19–23
example of time sampling using • *Device Support (A)*, 19–19, 19–23 to 19–25
example of watchdog timer using • *Device Support (A)*, 19–19, 19–20 to 19–21
interrupt service routine • *Device Support (A)*, 19–16 to 19–18
mapping I/O address space • *Device Support (A)*, 19–8
privileges required • *Device Support (A)*, 19–12
programming language requirements • *Device Support (A)*, 19–14
start I/O routine • *Device Support (A)*, 19–15 to 19–16
SYSGEN requirements • *Device Support (A)*, 19–9
unit initialization routine • *Device Support (A)*, 19–15
user-specified routines • *Device Support (A)*, 19–9, 19–13 to 19–19
Considerations when developing a parallel processing application • *RTL Parallel Processing*, 5–1
Console disk
See RX01 console disk
Console terminal • *I/O User's I*, 8–1
Constant
specifying radix of • *VAXTPU*, 3–37
TPU$K_DISJOINT • *VAXTPU*, 7–198, 7–368
TPU$K_INVISIBLE • *VAXTPU*, 7–198, 7–368
TPU$K_OFF_LEFT • *VAXTPU*, 7–198, 7–368
TPU$K_OFF_RIGHT • *VAXTPU*, 7–198, 7–368
TPU$K_UNMAPPED • *VAXTPU*, 7–198, 7–368
CONSTANT declaration • *VAXTPU*, 3–35
Constants • *VAXTPU*, 3–5 to 3–6
local • *VAXTPU*, 3–20
predefined • *VAXTPU*, 3–13
Contents-of operator • *Debugger*, 4–7, 4–20, D–7
Context
SDA CPU • *System Dump Analyzer*, SDA–10
SDA process • *System Dump Analyzer*, SDA–9
CONTEXT attribute • *File Def Language*, FDL–10, FDL–18
context data type • *Routines Intro*, A–5t
Context modes • *Patch*, PAT–15
See also Entry and display modes
Context switch
scalar • *MACRO*, 10–19, 10–20, 10–43
vector • *MACRO*, 10–32

Context variable
use with DCX routines • *Utility Routines,* DCX–16
Contiguity • *File Applications,* 10–29
CONTIGUOUS attribute • *File Def Language,* FDL–7, FDL–18
Contiguous-best-try option • *File Applications,* 4–30
See also FAB$V_CBT option
Contiguous option • *File Applications,* 4–30
See also FAB$V_CTG option
/CONTIGUOUS qualifier • *Linker,* LINK–4
CONTIGUOUS secondary attribute • *File Applications,* 3–23, 4–30
Continuation character (-)
in listing file • *MACRO,* 3–9
in source statement • *MACRO,* 2–1
use in VMS RMS coding • *RMS,* 3–6
Control action
inhibiting • *Programming Resources,* 7–42
Control and status register
See CSR
Control block • *File Def Language,* FDL–2
See also Data structure
See also VMS RMS
See Data structure
dual purpose • *RMS,* 1–4
field name conventions • *RMS,* 2–2
for extended attributes • *RMS,* 1–3
for file name operations • *RMS,* 1–3
for file services • *RMS,* 1–2
formatting • *System Dump Analyzer,* SDA–56
for record services • *RMS,* 1–4
macro names • *RMS,* 3–2
requirements for valid default values • *RMS,* 1–4
symbolic bit offset • *RMS,* 2–4
symbolic constant (keyword) value • *RMS,* 2–4
symbolic naming exceptions • *RMS,* 2–3
symbolic offsets • *RMS,* 2–2
types of macros • *RMS,* 3–1
use restrictions • *RMS,* 2–1
use with VAX languages • *RMS,* 2–1
Control block store macro
description • *RMS,* 3–1
example • *RMS,* 3–9
placement guidelines • *RMS,* 3–8
requirement for number sign • *RMS,* 3–8
use of R0 • *RMS,* 3–8
Control character
entering • *VAXTPU,* 3–2
list • *I/O User's I,* B–1
terminal • *I/O User's I,* 8–4 to 8–6, 8–9
Control character
translation
example • *VAXTPU,* A–2
Control code
function key • *VAXTPU,* 7–241
Control connection routines • *I/O User's I,* C–1
PTD$CANCEL • *I/O User's I,* C–2
PTD$CREATE • *I/O User's I,* C–3
PTD$DELETE • *I/O User's I,* C–6
PTD$READ • *I/O User's I,* C–7
PTD$SET_EVENT_NOTIFICATION • *I/O User's I,* C–9
PTD$WRITE • *I/O User's I,* C–12
Control instructions • *MACRO,* 9–42
Controller
See Device controller
Controller initialization routine • *Device Support (A),* 1–3, 11–1 to 11–6, 12–4, 12–8
address • *Device Support (A),* 4–6, 6–3, 11–1, 14–30; *Device Support (B),* 1–25, 2–26, 4–8
allocating controller data channel in • *Device Support (A),* 8–4
context • *Device Support (A),* 11–1; *Device Support (B),* 4–8
entry point • *Device Support (B),* 4–8
exit method • *Device Support (B),* 4–8
for generic VAXBI device • *Device Support (A),* 16–12 to 16–18
forking • *Device Support (B),* 1–21
forking in • *Device Support (A),* 3–24, 11–6
for terminal port driver • *Device Support (A),* 18–12; *Device Support (B),* 2–7
functions • *Device Support (A),* 11–1; *Device Support (B),* 4–9
input • *Device Support (A),* 11–2; *Device Support (B),* 4–8
register usage • *Device Support (B),* 4–8
synchronization requirements • *Device Support (A),* E–11 to E–12; *Device Support (B),* 4–8
Control mask
See Device activation bit mask
Control region • *System Services Intro,* 11–2; *System Dump Analyzer,* SDA–14
adding page to • *System Services,* SYS–163
base register • *System Dump Analyzer,* SDA–14
examining • *System Dump Analyzer,* SDA–52
length register • *System Dump Analyzer,* SDA–14
Control region operator (H) • *System Dump Analyzer,* SDA–12
Control region page table
displaying • *System Dump Analyzer,* SDA–127
Control region space prefix symbol • *Delta/XDelta,* DELTA–9

Control register
- See CSR
- See MBA$L_CR

Control routine • *RMS*, 4–27

Control sequence
- function key • *VAXTPU*, 7–241
- terminal • *I/O User's I*, 8–8

CONTROL_FIELD_SIZE attribute • *File Def Language*, FDL–34, FDL–35

CONTROL_FIELD_SIZE secondary attribute • *File Applications*, 4–29

CONV$CONVERT routine • *Utility Routines*, CONV–8

CONV$PASS_FILES routine • *Utility Routines*, CONV–11

CONV$PASS_OPTIONS routine • *Utility Routines*, CONV–14

CONV$RECLAIM routine • *Utility Routines*, CONV–18; *Convert*, CONV–4

Convention
- for calling system services • *System Services Intro*, 2–1

Conversion • *Convert*, CONV–3
- binary text to unsigned integer • *RTL General Purpose*, OTS–18
- floating-point to character string • *RTL General Purpose*, OTS–4
- hexadecimal text to unsigned integer • *RTL General Purpose*, OTS–37
- integer to binary text • *RTL General Purpose*, OTS–6
- integer to FORTRAN L format • *RTL General Purpose*, OTS–9
- integer to hexadecimal • *RTL General Purpose*, OTS–16
- numeric text to binary • *RTL Library*, LIB–76
- numeric text to floating-point • *RTL General Purpose*, OTS–31, OTS–35
- of VFC records • *Convert*, CONV–15
- unsigned decimal to integer • *RTL General Purpose*, OTS–28
- unsigned octal to signed integer • *RTL General Purpose*, OTS–25

Conversion function
- creating
  - using inverted conversion function • *National Char Set*, NCS–12
  - using modified conversion function • *National Char Set*, NCS–11
  - using name of existing conversion function • *National Char Set*, NCS–11
  - using reordered conversion function • *National Char Set*, NCS–12

Conversion function
- creating (Cont.)
  - using series of conversion functions • *National Char Set*, NCS–11
- expression forms listed • *National Char Set*, NCS–11
- MODIFICATIONS keyword clause format • *National Char Set*, NCS–16
- processing order for multiple • *National Char Set*, NCS–11
- using to create collating sequence • *National Char Set*, NCS–9

Conversion of double to single floating-point value • *RTL Math*, 1–9

Conversion to greatest floating-point integer • *RTL Math*, 1–6

CONVERT
- See Convert Utility

CONVERT built-in procedure • *VAXTPU*, 7–50
- example of use • *VAXTPU*, B–1 to B–4

CONVERT command • *RMS*, 4–9
- list of qualifiers • *Utility Routines*, CONV–14
- passing options • *Utility Routines*, CONV–14
- passing options in an array • *Utility Routines*, CONV–16
- setting qualifiers • *Utility Routines*, CONV–14

CONVERT/FDL command • *Programming Resources*, 8–58

Convert option
- See RAB$V_CVT option

CONVERT/RECLAIM
- See Convert/Reclaim Utility

Convert/Reclaim Utility (CONVERT/RECLAIM) • *Programming Resources*, 1–39; *File Applications*, 1–14, 3–16; *Convert*, CONV–1, CONV–3
- DCL qualifier • *Convert*, CONV–24
- directing output from • *Convert*, CONV–5
- example
  - reclaiming buckets • *Convert*, CONV–29
- exiting • *Convert*, CONV–5
- invoking • *Convert*, CONV–5
- restrictions • *Convert*, CONV–5
- with DECnet–VAX • *Convert*, CONV–3
- with Prolog 3 files • *File Applications*, 3–17, 10–30

Convert routines
- See CONV routines

Convert Utility (CONVERT) • *Programming Resources*, 1–39; *File Applications*, 1–13, 9–8; *Convert*, CONV–1; *File Def Language*, FDL–3
- appending a remote file • *Convert*, CONV–30

- Convert Utility (CONVERT) (Cont.)
  - converting a carriage control to stream • *Convert*, CONV–30
  - converting a remote file • *Convert*, CONV–29
  - converting carriage control formats • *Convert*, CONV–2
  - creating data files • *File Applications*, 4–17, 4–18
  - creating data files with • *File Def Language*, FDL–41
  - creating output files • *Convert*, CONV–1
  - DCL qualifiers • *Convert*, CONV–5 to CONV–28
  - directing output from • *Convert*, CONV–5
  - establishing RFAs • *Convert*, CONV–4
  - examples • *Convert*, CONV–28 to CONV–30
    - converting a carriage control file to variable length • *Convert*, CONV–30
    - converting fixed format to variable length • *Convert*, CONV–30
    - converting record formats • *Convert*, CONV–29
    - improving a file's performance • *Convert*, CONV–29
    - reorganizing a remote file • *Convert*, CONV–29
  - exception conditions • *Convert*, CONV–3
  - exiting • *Convert*, CONV–5
  - FDL output data file • *File Def Language*, FDL–41
  - invoking • *Convert*, CONV–5
  - library routine • *File Def Language*, FDL–41
  - loading output files • *Convert*, CONV–1
  - making a file contiguous • *File Applications*, 10–30
  - optimizing data files • *File Applications*, 10–29
  - populating a file • *File Applications*, 4–22
  - reorganizing files • *File Applications*, 10–31
  - reorganizing noncontiguous files • *File Applications*, 3–26, 10–30
  - restrictions • *Convert*, CONV–5
  - with corrupted files • *File Applications*, 10–1, 10–2
  - with DECnet–VAX • *Convert*, CONV–3
  - with FDL files • *File Applications*, 4–2
  - with Prolog 1 and 2 files • *File Applications*, 3–16
  - with Prolog 3 files • *File Applications*, 3–17
- CONV routines
  - examples • *Utility Routines*, CONV–1 to CONV–7
  - introduction • *Utility Routines*, CONV–1
  - list of • *Utility Routines*, CONV–1
  - using wildcard characters • *Utility Routines*, CONV–12
- Coordinate measuring system
  - converting to character-cell system • *VAXTPU*, 7–50
- COPY command • *System Dump Analyzer*, SDA–3, SDA–4, SDA–42
  - /CONTIGUOUS qualifier • *File Applications*, 9–8, 10–29
- Copying
  - vector • *RTL Math*, MTH–160
- Copy string • *RTL General Purpose*, OTS–90
- COPY_TEXT built-in procedure • *VAXTPU*, 7–53 to 7–54
- Coroutine • *Device Support (B)*, 3–35, 3–46, 3–59, 3–109
- Corrupted file • *Analyze/RMS_File*, ARMS–14
- Corruption
  - detecting • *Device Support (A)*, 13–23 to 13–27
- Cosine
  - hyperbolic • *RTL Math*, MTH–51, MTH–88
  - in degrees • *RTL Math*, MTH–49, MTH–87, MTH–127
  - in radians • *RTL Math*, MTH–47, MTH–86, MTH–124
  - of complex number • *RTL Math*, MTH–26, MTH–28
- Counting semaphore • *Programming Resources*, 4–17; *RTL Parallel Processing*, 4–10
  - operations on • *RTL Parallel Processing*, 4–10
- CPU
  - list • *Device Support (A)*, 1–10
  - per-CPU database • *Device Support (B)*, 1–12 to 1–19
    - locating • *Device Support (B)*, 2–31
- CPU$L_PHY_CPUID • *Device Support (B)*, 3–70
- CPU$Q_SWIQFL • *Device Support (A)*, E–14; *Device Support (B)*, 3–26, 3–30
- CPU$Q_WORK_IFQ • *Device Support (B)*, 1–17
- CPU (per-CPU database)
  - locating • *Device Support (A)*, E–7
- CPU context
  - changing • *System Dump Analyzer*, SDA–68, SDA–74, SDA–89, SDA–93, SDA–126
  - displaying • *System Dump Analyzer*, SDA–89
- CPUDISP macro • *Device Support (A)*, 5–6; *Device Support (B)*, 2–9 to 2–11
- CPU ID (CPU identification number) • *System Dump Analyzer*, SDA–89; *Device Support (B)*, 1–17, 3–70
- CPULOA.EXE
  - global symbols • *System Dump Analyzer*, SDA–60
- CPU time • *Convert*, CONV–24
- Crash dump
  - See also System failure
  - analysis • *System Dump Analyzer*, SDA–1 to SDA–165
  - incomplete • *System Dump Analyzer*, SDA–7

Crash dump (Cont.)
  short • *System Dump Analyzer,* SDA–7
Crash dump file
  header • *System Dump Analyzer,* SDA–106
/CRASH_DUMP qualifier • *System Dump Analyzer,* SDA–6
CRB$B_MASK • *Device Support (A),* 4–6, 16–8
CRB$L_DLCK • *Device Support (A),* 3–22
CRB$L_INTD • *Device Support (A),* 4–6; *Device Support (B),* 1–22 to 1–27
CRB$L_INTD+VEC$L_INITIAL • *Device Support (A),* 11–5
CRB$L_INTD+VEC$L_UNITINIT • *Device Support (A),* 11–5
CRB$L_LINK • *Device Support (A),* 15–13
CRB$L_WQBL • *Device Support (A),* 16–8
CRB$L_WQFL • *Device Support (A),* 4–6, 16–8; *Device Support (B),* 3–86, 3–91
CRB$V_UNINIT • *Device Support (A),* 16–8
CRB (channel request block) • *System Dump Analyzer,* SDA–99; *Device Support (A),* 1–6, 4–6 to 4–7; *Device Support (B),* 1–19 to 1–27
  alternate map register allocation information • *Device Support (A),* 14–20
  creation • *Device Support (A),* 12–4
  data path allocation information • *Device Support (A),* 14–17 to 14–18
  for generic VAXBI device • *Device Support (A),* 16–8
  fork block • *Device Support (A),* 3–24, 12–7; *Device Support (B),* 1–21
  for MBA • *Device Support (A),* 15–4, 15–7 to 15–8, 15–13, 15–15
  initializing • *Device Support (A),* 6–3; *Device Support (B),* 2–25
  map register allocation information • *Device Support (A),* 14–20
  periodic wakeup of • *Device Support (B),* 1–22
  primary • *Device Support (A),* 15–13; *Device Support (B),* 1–73
  reinitializing • *Device Support (A),* 6–3; *Device Support (B),* 2–25
  secondary • *Device Support (A),* 15–13; *Device Support (B),* 1–22
  synchronizing access to • *Device Support (A),* 3–16
CRC (Calculate Cyclic Redundancy Check) instruction • *MACRO,* 9–142
CR character • *File Def Language,* FDL–35
CREATE command • *Patch,* PAT–4, PAT–48; *File Def Language,* FDL–40, FDL–42; *System Dump Analyzer,* SDA–2
Created local label • *MACRO,* 4–7
Created local label (Cont.)
  range • *MACRO,* 3–7
CREATE/FDL
  See Create/FDL Utility
CREATE/FDL command • *Programming Resources,* 8–57; *RMS,* 4–9
Create/FDL Utility (CREATE/FDL) • *Programming Resources,* 1–39; *File Applications,* 1–14, 4–2, 4–17, 10–1; *File Def Language,* FDL–41, FDL–42
  creating a data file • *Programming Resources,* 8–57
  creating data files • *File Def Language,* FDL–41
  exiting • *File Def Language,* FDL–43
  invoking • *File Def Language,* FDL–43
  restrictions • *File Def Language,* FDL–43
Create file function • *I/O User's I,* 1–22
  directory entry creation • *I/O User's I,* 1–26
Create-if option • *File Applications,* 4–17, 4–27, 5–9; *RMS,* 4–1
  See also FAB$V_CIF option
$CREATE macro • *RMS,* 3–10
/CREATE qualifier • *Librarian,* LIB–12, LIB–17; *File Applications,* 4–11; *Convert,* CONV–8, CONV–17; *File Def Language,* FDL–42; *National Char Set,* NCS–24; *VAXTPU,* 5–7
  EDIT/FDL • *File Def Language,* FDL–48
Create service • *File Applications,* 4–17, 5–9; *RMS,* RMS–10
  condition values • *RMS,* RMS–19
  contrasted with Open service • *RMS,* 4–1
  control block input fields • *RMS,* RMS–11
  control block output fields • *RMS,* RMS–15
  for process-permanent files • *File Applications,* 6–21
  function • *RMS,* 4–1
  handling search list • *RMS,* RMS–11
  invoking • *RMS,* 4–1
  program example • *RMS,* 4–2
  prolog level • *RMS,* RMS–18
  using the create-if option • *RMS,* RMS–17
  using the NAM block • *RMS,* RMS–16
  using to create indexed files • *RMS,* RMS–18
  XAB override in various fields • *RMS,* RMS–11
"Create" string constant parameter to GET_INFO • *VAXTPU,* 7–177
CREATE_ARRAY built-in procedure • *VAXTPU,* 7–55 to 7–57
CREATE_BUFFER built-in procedure • *VAXTPU,* 7–58 to 7–62, 7–203
CREATE_IF attribute • *File Def Language,* FDL–19
CREATE_IF secondary attribute • *File Applications,* 4–27

CREATE_KEY_MAP built-in procedure • *VAXTPU,* 7–63 to 7–64
CREATE_KEY_MAP_LIST built-in procedure • *VAXTPU,* 7–65 to 7–66
CREATE_PROCESS built-in procedure • *VAXTPU,* 7–67 to 7–68
CREATE_RANGE built-in procedure • *VAXTPU,* 7–69 to 7–71
CREATE_WIDGET built-in procedure • *VAXTPU,* 7–72
  example of use • *VAXTPU,* B–4 to B–11
  using to specify callback routine • *VAXTPU,* 4–9
  using to specify resource values • *VAXTPU,* 4–12
CREATE_WINDOW built-in procedure • *VAXTPU,* 2–26, 7–77 to 7–79
CREATION attribute • *File Def Language,* FDL–16
Creation date and time field
  See XAB$Q_CDT field
Creation-time option • *File Applications,* 3–9, 4–1, 4–2, 4–17, 4–27, 4–28
$CRFCTLTABLE macro • *RTL Library,* 8–1, 8–2
$CRFFIELDEND macro • *RTL Library,* 8–1, 8–4
$CRFFIELD macro • *RTL Library,* 8–1, 8–3
Critical section
  definition of • *RTL Parallel Processing,* 1–2
.CROSS directive • *MACRO,* 6–16
Cross-reference directive
  .CROSS • *MACRO,* 6–16
  .NOCROSS • *MACRO,* 6–16
  (.NOCROSS) • *MACRO,* 6–66
Cross-reference of symbols • *Linker,* 5–1, LINK–5
  in map • *Linker,* 5–6
Cross-Reference Routines • *RTL Library,* 8–1
/CROSS_REFERENCE qualifier • *Librarian,* LIB–19; *Linker,* LINK–5
  using with /ONLY • *Librarian,* LIB–35
  using with /OUTPUT • *Librarian,* LIB–36
CROSS_WINDOW_BOUNDS keyword • *VAXTPU,* 7–361
"Cross_window_bounds" string constant parameter to GET_INFO • *VAXTPU,* 7–197
CSB (cluster system block) • *System Dump Analyzer,* SDA–82, SDA–87
CSID (cluster system identification number) • *System Dump Analyzer,* SDA–82, SDA–144
/CSID qualifier • *System Dump Analyzer,* SDA–82
CS keyword
  description • *National Char Set,* NCS–13
CSR (control and status register) • *I/O User's II,* 3–5; *Device Support (A),* 14–4, 14–23
  See also Device registers
CSR (control and status register) (Cont.)
  address • *Device Support (A),* 4–7, 8–4, 14–23; *Device Support (B),* 1–36
  bad address • *Device Support (B),* 1–36
  bit assignment • *I/O User's II,* 3–16
  displaying address • *Device Support (A),* 12–11
  fixed space • *Device Support (A),* 12–14
  floating space • *Device Support (A),* 12–14
  loading • *Device Support (A),* 8–5
  locating device registers from • *Device Support (A),* 14–23
  of LP11 printer • *Device Support (A),* 2–5
  specifying address • *Device Support (A),* 12–5
  specifying offset for multiunit controller • *Device Support (A),* 12–6
CTDRIVER • *I/O User's I,* 8–11, 8–35
CTG option • *File Def Language,* FDL–7, FDL–19
CTL$GL_CCBBASE • *Device Support (B),* 3–103
CTL$GL_PCB • *Device Support (A),* E–6
Ctrl/C • *Programming Resources,* 7–33; *Debugger,* 2–7, 10–5, 10–10, CD–38; *VAXTPU,* 4–20
  with case-style error handler • *VAXTPU,* 3–29, 3–30
  with procedural error handler • *VAXTPU,* 3–27, 3–28
Ctrl/W • *Debugger,* CD–40, CD–73
Ctrl/x
  See Terminal, control characters
Ctrl/Y • *Programming Resources,* 7–33; *Debugger,* 2–7, 3–3, 3–5, 10–15, CD–41
  with DECwindows • *Debugger,* 1–34
Ctrl/Z • *Programming Resources,* 7–5, 7–54; *Debugger,* 3–5, CD–40; *File Applications,* 4–4
  using as end-of-file marker • *RMS,* RMS–49
  using to terminate Get service • *RMS,* RMS–49
%CURDISP • *Debugger,* C–7
%CURLOC • *Debugger,* 4–8, D–5
Current
  display • *Debugger,* 7–3, 7–20, CD–123, CD–244
  entity • *Debugger,* 4–8, 4–20, D–5
    with DECwindows • *Debugger,* 1–8
  image • *Debugger,* 5–14, CD–142, CD–222
  language • *Debugger,* 4–10, CD–145, CD–226
  location • *Debugger,* 7–6, 7–9
  radix • *Debugger,* 4–11, CD–168, CD–240
  scope • *Debugger,* 5–11, CD–170, CD–241
  type • *Debugger,* 4–25, CD–195, CD–259
  value • *Debugger,* 4–6, D–5
Current buffer • *VAXTPU,* 7–59
  active editing point • *VAXTPU,* 2–4
  definition • *VAXTPU,* 7–80

- Current buffer direction • *VAXTPU*, 7–85
- Current context
  - current-record position • *File Applications*, 8–15
  - listed for VMS RMS services • *File Applications*, 8–14
  - next-record position • *File Applications*, 8–16
- Current date • *VAXTPU*, 7–138, 7–268, 7–271
- Current entity
  - field and buttons in main window
    - with DECwindows • *Debugger*, 1–8
- Current location counter • *MACRO*, 3–17
- Current location symbol (.) • *System Dump Analyzer*, SDA–13
- Current pointer position • *VAXTPU*, 7–252
- Current position option
  - See FAB$V_POS option
- /CURRENT qualifier • *Debugger*, 5–12, CD–170
- Current-record context • *File Applications*, 8–14
- Current-record position • *File Applications*, 8–3, 8–4
- "Current" string constant parameter to GET_INFO • *VAXTPU*, 7–166, 7–167, 7–169, 7–184, 7–191, 7–218
- Current time • *Programming Resources*, 3–23; *VAXTPU*, 7–138, 7–268, 7–271
- Current window • *VAXTPU*, 2–27, 7–77
- CURRENT_BUFFER built-in procedure • *VAXTPU*, 7–80
- CURRENT_CHARACTER built-in procedure • *VAXTPU*, 7–81 to 7–82
- CURRENT_COLUMN built-in procedure • *VAXTPU*, 7–83 to 7–84
- "Current_column" string constant parameter to GET_INFO • *VAXTPU*, 7–197, 7–222
- CURRENT_DIRECTION built-in procedure • *VAXTPU*, 7–85
- CURRENT_LINE built-in procedure • *VAXTPU*, 7–86 to 7–87
- CURRENT_OFFSET built-in procedure • *VAXTPU*, 7–88 to 7–89
- CURRENT_ROW built-in procedure • *VAXTPU*, 7–90 to 7–91
- "Current_row" string constant parameter to GET_INFO • *VAXTPU*, 7–197, 7–222
- %CURRENT_SCOPE_ENTRY • *Debugger*, D–11
- CURRENT_WINDOW built-in procedure • *VAXTPU*, 7–92 to 7–93
- %CURSCROLL • *Debugger*, C–7
- Cursor
  - detached
    - defining routine to handle • *VAXTPU*, 7–367
    - fetching action routine to handle • *VAXTPU*, 7–197
- Cursor
  - detached (Cont.)
    - fetching reason for • *VAXTPU*, 7–198
  - moving • *RTL Screen Management*, 4–3
  - turning on and off • *RTL Screen Management*, SMG–347
- Cursor movement • *Programming Resources*, 7–20; *VAXTPU*, 7–94, 7–96
  - free • *VAXTPU*, 7–95
- Cursor position
  - compared to editing point • *VAXTPU*, 6–10
  - effect of scrolling on • *VAXTPU*, 7–324
  - padding effects • *VAXTPU*, 6–11 to 6–12
- CURSOR_HORIZONTAL built-in procedure • *VAXTPU*, 7–94
- CURSOR_VERTICAL built-in procedure • *VAXTPU*, 7–96 to 7–98
- %CURVAL • *Debugger*, 4–6, D–5
- CVTBD (Convert Byte to D_floating) instruction • *MACRO*, 9–110
- CVTBF (Convert Byte to F_floating) instruction • *MACRO*, 9–110
- CVTBG (Convert Byte to G_floating) instruction • *MACRO*, 9–110
- CVTBH (Convert Byte to H_floating) instruction • *MACRO*, 9–110
- CVTBL (Convert Byte to Long) instruction • *MACRO*, 9–16
- CVTBW (Convert Byte to Word) instruction • *MACRO*, 9–16
- CVTDB (Convert D_floating to Byte) instruction • *MACRO*, 9–110
- CVTDF (Convert D_floating to F_floating) instruction • *MACRO*, 9–110
- CVTDH (Convert D_floating to H_floating) instruction • *MACRO*, 9–110
- CVTDL (Convert D_floating to Long) instruction • *MACRO*, 9–110
- CVTDW (Convert D_floating to Word) instruction • *MACRO*, 9–110
- CVTFB (Convert F_floating to Byte) instruction • *MACRO*, 9–110
- CVTFD (Convert F_floating to D_floating) instruction • *MACRO*, 9–110
- CVTFG (Convert F_floating to G_floating) instruction • *MACRO*, 9–110
- CVTFH (Convert F_floating to H_floating) instruction • *MACRO*, 9–110
- CVTFL (Convert F_floating to Long) instruction • *MACRO*, 9–110
- CVTFW (Convert F_floating to Word) instruction • *MACRO*, 9–110

CVTGB (Convert G_floating to Byte) instruction • *MACRO*, 9–110
CVTGF (Convert G_floating to F_floating) instruction • *MACRO*, 9–110
CVTGH (Convert G_floating to H_floating) instruction • *MACRO*, 9–110
CVTGL (Convert G_floating to Long) instruction • *MACRO*, 9–110
CVTGW (Convert G_floating to Word) instruction • *MACRO*, 9–110
CVTHB (Convert H_floating to Byte) instruction • *MACRO*, 9–110
CVTHD (Convert H_floating to D_floating) instruction • *MACRO*, 9–110
CVTHF (Convert H_floating to F_floating) instruction • *MACRO*, 9–110
CVTHG (Convert H_floating to G_floating) instruction • *MACRO*, 9–110
CVTHL (Convert H_floating to Long) instruction • *MACRO*, 9–110
CVTHW (Convert H_floating to Word) instruction • *MACRO*, 9–110
CVTLB (Convert Long to Byte) instruction • *MACRO*, 9–16
CVTLD (Convert Long to D_floating) instruction • *MACRO*, 9–110
CVTLF (Convert Long to F_floating) instruction • *MACRO*, 9–110
CVTLG (Convert Long to G_floating) instruction • *MACRO*, 9–110
CVTLH (Convert Long to H_floating) instruction • *MACRO*, 9–110
CVTLP (Convert Long to Packed) instruction • *MACRO*, 9–153
CVTLW (Convert Long to Word) instruction • *MACRO*, 9–16
CVT option • *File Def Language*, FDL–14
CVTPL (Convert Packed to Long) instruction • *MACRO*, 9–154
CVTPS (Convert Packed to Leading Separate Numeric) instruction • *MACRO*, 9–155
CVTPT (Convert Packed to Trailing Numeric) instruction • *MACRO*, 9–157
CVTRDL (Convert Rounded D_floating to Long) instruction • *MACRO*, 9–110
CVTRFL (Convert Rounded F_floating to Long) instruction • *MACRO*, 9–110
CVTRGL (Convert Rounded G_floating to Long) instruction • *MACRO*, 9–110
CVTRHL (Convert Rounded H_floating to Long) instruction • *MACRO*, 9–110
CVTSP (Convert Leading Separate Numeric to Packed) instruction • *MACRO*, 9–159
CVTTP (Convert Trailing Numeric to Packed) instruction • *MACRO*, 9–161
CVTWB (Convert Word to Byte) instruction • *MACRO*, 9–16
CVTWD (Convert Word to D_floating) instruction • *MACRO*, 9–110
CVTWF (Convert Word to F_floating) instruction • *MACRO*, 9–110
CVTWG (Convert Word to G_floating) instruction • *MACRO*, 9–110
CVTWH (Convert Word to H_floating) instruction • *MACRO*, 9–110
CVTWL (Convert Word to Long) instruction • *MACRO*, 9–16
Cyclic redundancy check instruction • *MACRO*, 9–141
Cyclic redundancy check table • *RTL Library*, LIB–33
Cylinder • *File Applications*, 1–5
  boundary • *File Applications*, 3–13
  options • *File Applications*, 4–31

D

DAN (data bucket area number)
  program example • *RMS*, 4–8
DAP (data access protocol) • *RMS*, 1–1
Data
  aligning • *Programming Resources*, 8–4
  corruption • *Analyze/RMS_File*, ARMS–14
  interprocess • *Programming Resources*, 5–13
  sharing • *Programming Resources*, 5–13
Data access protocol
  See DAP
Database
  compressing • *Programming Resources*, 8–26
  expanding • *Programming Resources*, 8–32
  record • *Programming Resources*, 8–10
Data bucket • *File Def Language*, FDL–27
  reclaiming • *Convert*, CONV–24
Data bucket area number
  See DAN
Data bucket area number field
  See XAB$B_DAN field
Data bucket fill size
  See DFL
Data bucket fill size field
  See XAB$W_DFL field
Data bucket size field
  See XAB$B_DBS field
DATA BUCKET structure • *File Applications*, 10–16, 10–20

Data buffer, LPA11-K • *I/O User's I*, 4–14
Data chaining • *I/O User's II*, 4–2, 6–26
Data check
  disk • *I/O User's I*, 3–15, 3–29, 3–30
  magnetic tape • *I/O User's I*, 6–8, 6–17, 6–18
Data compression • *File Applications*, 3–16
  See also DCX routines
  analysis preceding compression • *Utility Routines*, DCX–13
  compression algorithm
    submitting all data records • *Utility Routines*, DCX–15
  size of data after compression • *Utility Routines*, DCX–1
Data Compression/Expansion routines
  See DCX routines
Data compression facility • *Programming Resources*, 8–25
Data dependence • *RTL Parallel Processing*, 5–2 to 5–4
  antidependence • *RTL Parallel Processing*, 5–2
  control dependence • *RTL Parallel Processing*, 5–2, 5–3
  output dependence • *RTL Parallel Processing*, 5–2, 5–3
  true dependence • *RTL Parallel Processing*, 5–2
Data-expanded format
  using /DATA qualifier • *National Char Set*, NCS–26
Data expansion • *Utility Routines*, DCX–22
  See also DCX routines
  initializing • *Utility Routines*, DCX–25
Data file
  creating • *File Applications*, 4–17; *Convert*, CONV–1; *File Def Language*, FDL–39
  creating with FDL$CREATE routine • *File Applications*, 4–15, 4–18
  reorganizing • *File Applications*, 10–29
Data format in NCS library
  specifying with /DATA qualifier • *National Char Set*, NCS–26
Data level
  comparing for primary and alternate keys • *RMS*, 13–4
Data path • *Device Support (A)*, 1–22, 14–7 to 14–15, 14–17 to 14–19; *Device Support (B)*, 1–25 to 1–26
  See also Buffered data path
  See also Direct data path
  autopurging • *Device Support (B)*, 1–8, 2–3
  buffered • *Device Support (A)*, 14–3; *Device Support (B)*, 1–8, 2–3
  direct • *Device Support (B)*, 2–3
Data path (Cont.)
  mixed use of direct and buffered • *Device Support (A)*, 14–19
  purging • *Device Support (A)*, 10–2, 14–14, 14–19, 14–24 to 14–25; *Device Support (B)*, 2–51, 3–82 to 3–83
  speed • *Device Support (A)*, 14–10, 14–11, 14–15
Data path allocation bit map • *Device Support (B)*, 1–9
Data path register • *Device Support (A)*, 14–8, 14–15
  purge error • *Device Support (B)*, 3–83
Data path wait queue • *Device Support (A)*, 14–25, E–14; *Device Support (B)*, 1–7, 3–88, 3–97
/DATA qualifier • *Librarian*, LIB–20; *National Char Set*, NCS–26
  See also /COMPRESS qualifier
  using with /OUTPUT • *Librarian*, LIB–36
Data record • *Analyze/RMS_File*, ARMS–6; *File Def Language*, FDL–5
  analysis • *Utility Routines*, DCX–11
  compression • *Utility Routines*, DCX–1
  conversion • *Utility Routines*, CONV–1
  conversion statistics • *Utility Routines*, CONV–8
  expansion • *Utility Routines*, DCX–1
Data-reduced format
  using /DATA qualifier • *National Char Set*, NCS–26
Data reliability • *File Applications*, 9–11
Data security erase
  magnetic tape • *I/O User's I*, 6–27
Data storage • *Device Support (A)*, 5–1
  and file organization • *File Applications*, 3–2
  device specific • *Device Support (A)*, 4–5, 11–3; *Device Support (B)*, 1–41, 1–68, 2–22
Data storage directive
  .ADDRESS • *MACRO*, 6–4
  .ASCIC • *MACRO*, 6–8
  .ASCID • *MACRO*, 6–9
  .ASCII • *MACRO*, 6–10
  .ASCIZ • *MACRO*, 6–11
  .BYTE • *MACRO*, 6–14
  .D_FLOATING • *MACRO*, 6–20
  .F_FLOATING • *MACRO*, 6–35
  .G_FLOATING • *MACRO*, 6–36
  .H_FLOATING • *MACRO*, 6–38
  .LONG • *MACRO*, 6–56
  .OCTA • *MACRO*, 6–70
  .PACKED • *MACRO*, 6–74
  .QUAD • *MACRO*, 6–82
  .SIGNED_BYTE • *MACRO*, 6–91
  .SIGNED_WORD • *MACRO*, 6–92
  .WORD • *MACRO*, 6–102

Data structure • *Analyze/RMS_File*, ARMS–1; *Device Support (B)*, 1–1
See also I/O database
defining bit field within • *Device Support (B)*, 2–102 to 2–103
defining field within • *Device Support (B)*, 2–14, 2–15, 2–16
FAB (file access block) • *Programming Resources*, 1–36
formatting • *System Dump Analyzer*, SDA–56
global symbols • *System Dump Analyzer*, SDA–60
initializing • *Device Support (A)*, 6–1; *Device Support (B)*, 2–24 to 2–26
NAM (name block) • *Programming Resources*, 1–36
RAB (record access block) • *Programming Resources*, 1–36
stepping through a linked list • *System Dump Analyzer*, SDA–64
XAB (extended attribute block) • *Programming Resources*, 1–36
Data transfer
See also DMA transfer, PIO transfer
alignment • *Device Support (A)*, 14–3
buffering mechanisms • *Device Support (A)*, 17–15
byte aligned • *Device Support (A)*, 14–3, 14–22; *Device Support (B)*, 2–3, 3–78
byte count • *Device Support (B)*, 1–79, 1–83
byte offset • *Device Support (A)*, 14–13, 14–18; *Device Support (B)*, 1–79, 3–77
incomplete • *Device Support (A)*, 17–19
in reverse direction • *Device Support (A)*, 15–4, 15–15
longword-aligned 32-bit random-access • *Device Support (A)*, 14–11
mapping local buffer for • *Device Support (A)*, 17–27
mapping local buffer for SCSI port • *Device Support (A)*, 17–16 to 17–17; *Device Support (B)*, 2–77 to 2–79
maximum size of • *Device Support (A)*, 17–14, 17–19
mixing read and write functions in • *Device Support (A)*, 14–10
negative byte count • *Device Support (B)*, 3–32, 3–35, 3–41, 3–43, 3–46, 3–55, 3–56, 3–59
overlapping with seek operation • *Device Support (A)*, 8–2
performing • *Device Support (A)*, 17–13 to 17–19
size • *Device Support (A)*, 14–23
speed • *Device Support (A)*, 14–10, 14–11, 14–15
starting address • *Device Support (A)*, 14–22 to 14–23; *Device Support (B)*, 1–79
Data transfer (Cont.)
to randomly ordered addresses • *Device Support (A)*, 14–10
unmapping local buffer • *Device Support (A)*, 17–17, 17–28; *Device Support (B)*, 2–91
word aligned • *Device Support (A)*, 14–3; *Device Support (B)*, 3–78
zero byte count • *Device Support (B)*, 3–32, 3–41, 3–55
Data transfer command table
LPA11-K • *I/O User's I*, 4–11
Data transfer mode • *I/O User's II*, 3–4
as controlled by a third-party SCSI class driver • *Device Support (A)*, 17–13; *Device Support (B)*, 2–88
as controlled by the generic SCSI class driver • *I/O User's I*, 11–7, 11–13
asynchronous • *I/O User's I*, 11–7, 11–13; *Device Support (A)*, 17–13; *Device Support (B)*, 2–88
determining setting of • *Device Support (B)*, 2–75
synchronous • *I/O User's I*, 11–7, 11–13; *Device Support (A)*, 17–13; *Device Support (B)*, 2–88
Data transfers
meaning of terms read and write • *I/O User's II*, 3–5
Data transfer start command
LPA11-K • *I/O User's I*, 4–12
Data transfer stop command
LPA11-K • *I/O User's I*, 4–14
Data type • *Modular Procedures*, B–6; *Routines Intro*, 2–15; *File Applications*, 3–16; *MACRO*, 8–1; *VAXTPU*, 1–6 to 1–7
See also Type
Ada declaration • *Routines Intro*, A–13
APL declaration • *Routines Intro*, A–15
atomic • *Routines Intro*, 2–15
DSC$K_DTYPE_B • *Routines Intro*, 2–16
DSC$K_DTYPE_BU • *Routines Intro*, 2–16
DSC$K_DTYPE_CIT • *Routines Intro*, 2–17
DSC$K_DTYPE_D • *Routines Intro*, 2–16
DSC$K_DTYPE_DC • *Routines Intro*, 2–17
DSC$K_DTYPE_F • *Routines Intro*, 2–16
DSC$K_DTYPE_FC • *Routines Intro*, 2–16
DSC$K_DTYPE_G • *Routines Intro*, 2–16
DSC$K_DTYPE_GC • *Routines Intro*, 2–17
DSC$K_DTYPE_H • *Routines Intro*, 2–16
DSC$K_DTYPE_HC • *Routines Intro*, 2–17
DSC$K_DTYPE_L • *Routines Intro*, 2–16
DSC$K_DTYPE_LU • *Routines Intro*, 2–16
DSC$K_DTYPE_O • *Routines Intro*, 2–16

Data type
  atomic (Cont.)
    DSC$K_DTYPE_OU • *Routines Intro,* 2–16
    DSC$K_DTYPE_Q • *Routines Intro,* 2–16
    DSC$K_DTYPE_QU • *Routines Intro,* 2–16
    DSC$K_DTYPE_W • *Routines Intro,* 2–16
    DSC$K_DTYPE_WU • *Routines Intro,* 2–16
    DSC$K_DTYPE_Z • *Routines Intro,* 2–16
  BASIC declaration • *Routines Intro,* A–18
  BLISS declaration • *Routines Intro,* A–22
  byte • *MACRO,* 8–1
  C declaration • *Routines Intro,* A–25
  character string • *MACRO,* 8–7
  checking • *VAXTPU,* 4–12, 7–432
  COBOL declaration • *Routines Intro,* A–28
  COBOL intermediate temporary • *Routines Intro,* 2–20
  code • *Routines Intro,* 1–8
    facility-specific • *Routines Intro,* 2–19
    reserved • *Routines Intro,* 2–20
  definition • *VAXTPU,* 2–1
  floating-point • *MACRO,* 8–3, 8–4, 8–5, 9–101
  FORTRAN declaration • *Routines Intro,* A–31
  integer • *MACRO,* 8–1
  keywords
    ARRAY • *VAXTPU,* 2–2 to 2–3
    BUFFER • *VAXTPU,* 2–3 to 2–4
    INTEGER • *VAXTPU,* 2–5
    KEYWORD • *VAXTPU,* 2–5 to 2–7
    LEARN • *VAXTPU,* 2–7 to 2–8
    MARK • *VAXTPU,* 2–8 to 2–10
    PATTERN • *VAXTPU,* 2–11 to 2–20
    PROCESS • *VAXTPU,* 2–20 to 2–21
    PROGRAM • *VAXTPU,* 2–21
    RANGE • *VAXTPU,* 2–21 to 2–22
    STRING • *VAXTPU,* 2–23 to 2–24
    UNSPECIFIED • *VAXTPU,* 2–24
    WIDGET • *VAXTPU,* 2–24 to 2–25
    WINDOW • *VAXTPU,* 2–25 to 2–29
  leading separate numeric string • *MACRO,* 8–11
  longword • *MACRO,* 8–2
  MACRO declaration • *Routines Intro,* A–36
  miscellaneous • *Routines Intro,* 2–18
    DSC$K_DTYPE_ADT • *Routines Intro,* 2–19
    DSC$K_DTYPE_BLV • *Routines Intro,* 2–19
    DSC$K_DTYPE_BPV • *Routines Intro,* 2–19
    DSC$K_DTYPE_DSC • *Routines Intro,* 2–19
    DSC$K_DTYPE_ZEM • *Routines Intro,* 2–19
    DSC$K_DTYPE_ZI • *Routines Intro,* 2–19
  octaword • *MACRO,* 8–3
  packed decimal string • *MACRO,* 8–13
  Pascal declaration • *Routines Intro,* A–38

Data type (Cont.)
  PL/I declaration • *Routines Intro,* A–42
  quadword • *MACRO,* 8–2
  RPG II declaration • *Routines Intro,* A–48
  SCAN declaration • *Routines Intro,* A–51
  string • *Routines Intro,* 2–17; *MACRO,* 8–7
    DSC$K_DTYPE_NL • *Routines Intro,* 2–18
    DSC$K_DTYPE_NLO • *Routines Intro,* 2–18
    DSC$K_DTYPE_NR • *Routines Intro,* 2–18
    DSC$K_DTYPE_NRO • *Routines Intro,* 2–18
    DSC$K_DTYPE_NU • *Routines Intro,* 2–18
    DSC$K_DTYPE_NZ • *Routines Intro,* 2–18
    DSC$K_DTYPE_P • *Routines Intro,* 2–18
    DSC$K_DTYPE_T • *Routines Intro,* 2–17
    DSC$K_DTYPE_V • *Routines Intro,* 2–18
    DSC$K_DTYPE_VT • *Routines Intro,* 2–17, 2–21
    DSC$K_DTYPE_VU • *Routines Intro,* 2–18
  trailing numeric string • *MACRO,* 8–8
  variable-length bit field • *MACRO,* 8–6
  varying character string • *Routines Intro,* 2–21
    DSC$K_DTYPE_VT • *Routines Intro,* 2–21
  VAX standard • *Routines Intro,* 1–8
  VMS • *Routines Intro,* A–1
    access_bit_names • *Routines Intro,* A–2
    access_mode • *Routines Intro,* A–2
    address • *Routines Intro,* A–2t
    address_range • *Routines Intro,* A–2t
    arg_list • *Routines Intro,* A–2t
    ast_procedure • *Routines Intro,* A–2t
    boolean • *Routines Intro,* A–2t
    byte_signed • *Routines Intro,* A–2t
    channel • *Routines Intro,* A–2t
    char_string • *Routines Intro,* A–2t
    complex_number • *Routines Intro,* A–3t
    cond_value • *Routines Intro,* A–4t
    context • *Routines Intro,* A–5t
    date_time • *Routines Intro,* A–5t
    device_name • *Routines Intro,* A–5t
    ef_cluster_name • *Routines Intro,* A–5t
    ef_number • *Routines Intro,* A–5t
    exit_handler_block • *Routines Intro,* A–5t
    fab • *Routines Intro,* A–5t
    file_protection • *Routines Intro,* A–5t
    floating_point • *Routines Intro,* A–6t
    function_code • *Routines Intro,* A–7t
    identifier • *Routines Intro,* A–7t
    io_status_block • *Routines Intro,* A–7t
    item_list_2 • *Routines Intro,* A–8t
    item_list_3 • *Routines Intro,* A–8t
    item_list_pair • *Routines Intro,* A–9t

Data type
  VMS (Cont.)
    item_quota_list • *Routines Intro*, A–9t
    lock_id • *Routines Intro*, A–9t
    lock_status_block • *Routines Intro*, A–9t
    lock_value_block • *Routines Intro*, A–10t
    logical_name • *Routines Intro*, A–10t
    longword_signed • *Routines Intro*, A–10t
    longword_unsigned • *Routines Intro*, A–10t
    mask_byte • *Routines Intro*, A–10t
    mask_longword • *Routines Intro*, A–10t
    mask_word • *Routines Intro*, A–10t
    null_arg • *Routines Intro*, A–10t
    octaword_signed • *Routines Intro*, A–10t
    octaword_unsigned • *Routines Intro*, A–10t
    page_protection • *Routines Intro*, A–10t
    procedure • *Routines Intro*, A–11t
    process_id • *Routines Intro*, A–11t
    process_name • *Routines Intro*, A–11t
    quadword_signed • *Routines Intro*, A–11t
    quadword_unsigned • *Routines Intro*, A–11t
    quad_longword • *Routines Intro*, A–10t
    rab • *Routines Intro*, A–12t
    rights_holder • *Routines Intro*, A–11t
    rights_id • *Routines Intro*, A–12t
    section_id • *Routines Intro*, A–12t
    section_name • *Routines Intro*, A–12t
    system_access_id • *Routines Intro*, A–12t
    time_name • *Routines Intro*, A–12t
    transaction_id • *Routines Intro*, A–12t
    uic • *Routines Intro*, A–12t
    user_arg • *Routines Intro*, A–13t
    varying_arg • *Routines Intro*, A–13t
    vector_byte_signed • *Routines Intro*, A–13t
    vector_byte_unsigned • *Routines Intro*, A–13t
    vector_longword_signed • *Routines Intro*, A–13t
    vector_longword_unsigned • *Routines Intro*, A–13t
    vector_quadword_signed • *Routines Intro*, A–13t
    vector_quadword_unsigned • *Routines Intro*, A–13t
    vector_word_signed • *Routines Intro*, A–13t
    vector_word_unsigned • *Routines Intro*, A–13t
    word_signed • *Routines Intro*, A–13t
    word_unsigned • *Routines Intro*, A–13t
  VMS Usage • *Routines Intro*, 1–7
  word • *MACRO*, 8–2
Data type of key field
  See XAB$B_DTP field
Data underrun/overrun
  with LPA11-K • *I/O User's I*, 4–12
DATA_AREA attribute • *File Def Language*, FDL–27, FDL–28
DATA_AREA secondary attribute • *File Applications*, 3–24
DATA_FILL attribute • *File Def Language*, FDL–4, FDL–27
DATA_KEY_COMPRESSION attribute • *File Def Language*, FDL–4, FDL–27
DATA_RECORD_COMPRESSION attribute • *File Def Language*, FDL–4, FDL–27
DATA_RECORD_COUNT attribute • *File Def Language*, FDL–5
DATA_SPACE_OCCUPIED attribute • *File Def Language*, FDL–5
Date
  getting current system • *System Services Intro*, 9–2
  inserting with FAO • *VAXTPU*, 7–138
  inserting with MESSAGE • *VAXTPU*, 7–268
  inserting with MESSAGE_TEXT • *VAXTPU*, 7–271
  Smithsonian base • *System Services Intro*, 9–2
  system format • *System Services Intro*, 9–2
Date and time extended address block
  See XABDAT block
DATE attribute • *File Def Language*, FDL–2, FDL–15
Date-information option • *File Applications*, 4–28
DATE primary • *File Applications*, 4–28
Date/Time routine
  LIB$DATE_TIME • *RTL Library*, LIB–80
  LIB$DAY • *RTL Library*, LIB–82
  LIB$DAY_OF_WEEK • *RTL Library*, LIB–84
date_time data type • *Routines Intro*, A–5t
/DATE_TIME qualifier • *Debugger*, CD–62, CD–86
DAT file type • *Analyze/RMS_File*, ARMS–10
DAT_NCMPR option • *File Def Language*, FDL–27
DBG$DECW$DISPLAY
  with DECwindows • *Debugger*, 1–34, 1–35, 1–36, D–1
DBG$INIT • *Debugger*, 8–4, D–1
DBG$INPUT • *Debugger*, 9–5, D–1
DBG$OUTPUT • *Debugger*, 9–5, D–1
DBG$PROCESS • *Debugger*, 2–6, 10–1, 10–10, D–1
  with DECwindows • *Debugger*, 1–3, 1–30
$DCDEF macro • *Device Support (B)*, 1–76, 2–3, 2–21
DCL (DIGITAL Command Language)
  command language routines • *Command Def*, CDU–17
  command processing • *Command Def*, CDU–1 to CDU–2

DCL command line
  overriding /RECOVER qualifiers on • *VAXTPU*, 7–408
DCL command procedure
  example • *VAXTPU*, A–5
DCL commands
  ANALYZE/RMS_FILE • *Programming Resources*, 8–55
  ASSIGN • *Linker*, LINK–21
  CONVERT/FDL • *Programming Resources*, 8–58
  CREATE/FDL • *Programming Resources*, 8–57
  DEFINE • *Linker*, LINK–21
  EDIT/FDL • *Programming Resources*, 8–55
  LIBRARY • *Linker*, 2–3
  RUN • *Linker*, 2–5
  SET VERIFY • *Linker*, 3–4
DCL Command string
  See Command string
DCLDEF.STB • *System Dump Analyzer*, SDA–60
DCL interpreter
  global symbols • *System Dump Analyzer*, SDA–60
DCX$ANALYZE_DATA routine • *Utility Routines*, DCX–11
DCX$ANALYZE_DONE routine • *Utility Routines*, DCX–13
DCX$ANALYZE_INIT routine • *Utility Routines*, DCX–14
DCX$COMPRESS_DATA routine • *Utility Routines*, DCX–17
DCX$COMPRESS_DONE routine • *Utility Routines*, DCX–19
DCX$COMPRESS_INIT routine • *Utility Routines*, DCX–20
DCX$EXPAND_DATA routine • *Utility Routines*, DCX–22
DCX$EXPAND_DONE routine • *Utility Routines*, DCX–24
DCX$EXPAND_INIT routine • *Utility Routines*, DCX–25
DCX$MAKE_MAP routine • *Utility Routines*, DCX–27
DCX (Data/Expansion) routine • *Programming Resources*, 8–25
DCX routines
  examples • *Utility Routines*, DCX–2 to DCX–10
  introduction • *Utility Routines*, DCX–1
  procedure for use • *Utility Routines*, DCX–1
  when to use • *Utility Routines*, DCX–1
  with multiple streams of data records • *Utility Routines*, DCX–1
DDB$L_LINK • *Device Support (A)*, 11–5
DDB$L_UCB • *Device Support (A)*, 11–5
DDB$T_DRVNAME • *Device Support (A)*, 4–8
DDB$T_NAME • *Device Support (A)*, 4–8
DDB (device data block) • *System Dump Analyzer*, SDA–99; *Device Support (A)*, 1–5, 4–8, 11–5; *Device Support (B)*, 1–27 to 1–28
  address • *Device Support (B)*, 1–74
  creation • *Device Support (A)*, 12–4
  initializing • *Device Support (A)*, 6–3; *Device Support (B)*, 2–25
  reinitializing • *Device Support (A)*, 6–3; *Device Support (B)*, 2–25
DDCMP (DIGITAL Data Communications Message Protocol) • *I/O User's II*, 1–1, 2–1
DDI (DR32 device interconnect) • *I/O User's II*, 4–1, 4–2
  status returns • *I/O User's II*, 4–37
DDT$L_ALTSTART • *Device Support (A)*, 7–5; *Device Support (B)*, 4–2
DDT$L_CANCEL • *Device Support (B)*, 4–4
DDT$L_CLONEDUCB • *Device Support (B)*, 4–6
DDT$L_REGDUMP • *Device Support (B)*, 4–15
DDT$L_START • *Device Support (B)*, 4–17
DDT$L_UNITINIT • *Device Support (A)*, 11–5; *Device Support (B)*, 4–22
DDT$L_UNSOLINT • *Device Support (B)*, 4–24
DDT$W_ERRORBUF • *Device Support (A)*, 11–9, 17–21
DDT (driver dispatch table) • *System Dump Analyzer*, SDA–99; *Device Support (A)*, 1–2, 11–1, 11–10; *Device Support (B)*, 1–29 to 1–31, 3–102
  address • *Device Support (A)*, 6–3; *Device Support (B)*, 1–28, 1–80, 2–25
  creating • *Device Support (A)*, 6–3 to 6–4, 11–4; *Device Support (B)*, 2–12 to 2–13
  of terminal class driver • *Device Support (A)*, 18–19
  relocating addresses specified in • *Device Support (A)*, 11–4
DDTAB macro • *Device Support (A)*, 11–9, 12–1; *Device Support (B)*, 2–12 to 2–13, 3–102
  example • *Device Support (B)*, 2–13
Deaccess file function • *I/O User's I*, 1–28
Deadlock • *Modular Procedures*, 3–21; *RTL Parallel Processing*, 5–4
  avoidance • *RTL Parallel Processing*, 5–5
  detection and recovery • *RTL Parallel Processing*, 5–5
  prevention • *RTL Parallel Processing*, 5–4
Deadlock detection • *System Services Intro*, 12–6
DEBNA driver
  See Ethernet/802 drivers
$DEBUG$INI$ buffer • *VAXTPU*, 4–22
DEBUG command • *Debugger*, 3–3, 10–15, CD–41; *VAXTPU*, 4–35

DEBUG command (Cont.)
with DECwindows • *Debugger*, 1–34
.DEBUG directive • *MACRO*, 6–18
Debug directive (.DEBUG) • *MACRO*, 6–18
Debugger • *Programming Resources*, 1–14 to 1–16; *Debugger*, 1–1
See also Delta/XDelta Utility
See also Symbolic debugger
command interface • *Debugger*, 2–1
with DECwindows • *Debugger*, 1–28, 1–35
DECwindows interface • *Debugger*, 1–1
displaying DECwindows interface on other workstation • *Debugger*, 1–36
including • *Linker*, LINK–6
invoking • *VAXTPU*, 4–33
invoking from DECwindows FileView window • *Debugger*, 1–33
module name • *MACRO*, 6–23
routine name • *MACRO*, 6–23
symbol table • *Linker*, 6–18
using with completion status codes • *RMS*, A–2
Debugger command
dictionary • *Debugger*, CD–3
format • *Debugger*, CD–3
repeating • *Debugger*, CD–103, CD–115, CD–277
summary • *Debugger*, 2–27
with DECwindows • *Debugger*, 1–28
Debugger window (DECwindows interface) • *Debugger*, 1–5
Debugging • *Debugger*, 1–1; *VAXTPU*, 4–33 to 4–37
at elevated IPL • *Delta/XDelta*, DELTA–1
at IPL 0 • *Delta/XDelta*, DELTA–1
ATTACH command • *VAXTPU*, 4–36
CANCEL BREAKPOINT command • *VAXTPU*, 4–36
command files • *VAXTPU*, 4–34
condition handler • *Programming Resources*, 9–20
DECwindows application • *Debugger*, 1–36
DEPOSIT command • *VAXTPU*, 4–36
device driver • *Device Support (A)*, 13–1 to 13–30
DISPLAY SOURCE command • *VAXTPU*, 4–36
EXAMINE command • *VAXTPU*, 4–36
exit handler • *Programming Resources*, 9–30
GO command • *VAXTPU*, 4–34, 4–36
HELP command • *VAXTPU*, 4–36
privileged code • *Delta/XDelta*, DELTA–1
program • *VAXTPU*, 4–35
QUIT command • *VAXTPU*, 4–36
SCROLL command • *VAXTPU*, 4–37
section files • *VAXTPU*, 4–34

Debugging (Cont.)
SET BREAK POINT command • *VAXTPU*, 4–34, 4–37
SET WINDOW command • *VAXTPU*, 4–37
SHIFT command • *VAXTPU*, 4–37
SHOW BREAKPOINTS command • *VAXTPU*, 4–37
source code • *VAXTPU*, 4–35
SPAWN command • *VAXTPU*, 4–37
STEP command • *VAXTPU*, 4–35, 4–37
to examine contents of local variable • *VAXTPU*, 4–36
TPU command • *VAXTPU*, 4–37
user mode programs • *Delta/XDelta*, DELTA–1
Debugging configuration
default • *Debugger*, 2–6, 10–10
with DECwindows • *Debugger*, 1–3
multiprocess • *Debugger*, 10–1, 10–10
with DECwindows • *Debugger*, 1–30
Debugging programs that use VM zones • *RTL Library*, 6–1
DEBUG keyword • *VAXTPU*, 7–362, 7–363, 7–364
DEBUGON procedure • *VAXTPU*, 4–35
/DEBUG qualifier • *Debugger*, 3–1, 5–2, 5–4, 6–1; *Linker*, LINK–6; *VAXTPU*, 4–33, 5–8
shareable image • *Debugger*, 5–12
with DECwindows • *Debugger*, 1–3
Debug symbol table
See DST
DEBUG_LINE built-in procedure • *VAXTPU*, 7–99
%DEC • *Debugger*, 4–12, D–5
DEC026 card reader code • *I/O User's I*, 2–2, 2–8
DEC029 card reader code • *I/O User's I*, 2–2, 2–8
DECB (Decrement Byte) instruction • *MACRO*, 9–17
Decimal/hexadecimal conversion • *MACRO*, B–2
table • *MACRO*, B–1
DECIMAL mode • *Patch*, PAT–17
Decimal number • *File Def Language*, FDL–2
Decimal overflow detection • *RTL Library*, LIB–104
Decimal overflow enable (DV) • *MACRO*, 8–16
/DECIMAL qualifier • *Debugger*, 4–12, CD–81, CD–83, CD–86
/DECIMAL qualifier
with DELETE command • *Patch*, PAT–52
with DEPOSIT command • *Patch*, PAT–55
with EXAMINE command • *Patch*, PAT–62
with INSERT command • *Patch*, PAT–68
with REPLACE command • *Patch*, PAT–72
with SET MODE command • *Patch*, PAT–76
with VERIFY command • *Patch*, PAT–90
Decimal string descriptor • *Routines Intro*, 2–30
Decimal string instructions • *MACRO*, 9–144

- Decimal text
  - converting to binary • *RTL Library,* LIB–76
- Decimal value
  - of an expression • *System Dump Analyzer,* SDA–48
- DECIMAL value • *File Def Language,* FDL–31
- DECL (Decrement Long) instruction • *MACRO,* 9–17
- DECLARE command • *Debugger,* 8–2, CD–44
- DEC Multinational Character Set • *National Char Set,* NCS–3; *I/O User's I,* B–1; *VAXTPU,* 3–1 to 3–2, E–1 to E–8
  - string comparison • *RTL String Manipulation,* STR–11, STR–17
  - string conversion • *RTL String Manipulation,* STR–89
  - using • *RMS,* 2–7
- DECnet data structures
  - global symbols • *System Dump Analyzer,* SDA–60
- DECnet remote file access
  - specifying maximum record size • *RMS,* 5–22
- DECnet–VAX
  - using the Analyze/RMS_File Utility with • *Analyze/RMS_File,* ARMS–7
  - using the Convert/Reclaim Utility (CONVERT /RECLAIM) with • *Convert,* CONV–3
- Decomposition • *RTL Parallel Processing,* 5–1
- DECtalk device
  - checking hardware status • *RTL DECtalk,* DTK–5
  - connecting a terminal to • *RTL DECtalk,* 1–2 to 1–3
  - controlling the terminal • *RTL DECtalk,* 1–2 to 1–3
  - initializing • *RTL DECtalk,* 1–1, DTK–10
  - mode of operation • *RTL DECtalk,* 1–1 to 1–2
    - setting terminal attributes • *RTL DECtalk,* 1–3, DTK–25, DTK–29
    - setting terminal logging • *RTL DECtalk,* 1–2 to 1–3, DTK–22
  - specifying an output destination • *RTL DECtalk,* 1–1
  - voice characteristics • *RTL DECtalk,* 1–2, DTK–31
  - voice identifier • *RTL DECtalk,* 1–1
- DECtalk dictionary • *RTL DECtalk,* 1–4
  - loading • *RTL DECtalk,* DTK–12
- DECtalk index • *RTL DECtalk,* 1–4
  - returning last spoken • *RTL DECtalk,* 1–4, DTK–18
  - setting • *RTL DECtalk,* 1–4, DTK–19
- DECtalk routine • *RTL DECtalk,* 1–1
  - controlling the speech • *RTL DECtalk,* 1–4 to 1–5, DTK–27
    - speaking phonemic text • *RTL DECtalk,* DTK–35
    - speaking text • *RTL DECtalk,* DTK–37
- DECtalk routine
  - controlling the speech (Cont.)
    - speaking text in a file • *RTL DECtalk,* DTK–33
    - spelling text • *RTL DECtalk,* DTK–39
  - initializing • *RTL DECtalk,* 1–1, DTK–10
  - overview of • *RTL DECtalk,* 1–1
  - terminating • *RTL DECtalk,* 1–4, DTK–41
  - using the telephone • *RTL DECtalk,* 1–5
    - answering the phone • *RTL DECtalk,* 1–5, DTK–3
    - dialing the phone • *RTL DECtalk,* 1–5, DTK–7
    - hanging up the phone • *RTL DECtalk,* 1–5, DTK–9
    - recognizing the keypad • *RTL DECtalk,* 1–5, DTK–20
    - using keypad for input • *RTL DECtalk,* 1–5, DTK–14, DTK–16
  - writing an exit handler • *RTL DECtalk,* 1–6
- DECW (Decrement Word) instruction • *MACRO,* 9–17
- DECwindows
  - debugging DECwindows application • *Debugger,* 1–36
  - VAXTPU
    - determining if present • *VAXTPU,* 7–197
    - invoking with /DISPLAY • *VAXTPU,* 5–8
    - sample uses of built-ins • *VAXTPU,* B–1 to B–33
- DECwindows interface
  - debugger • *Debugger,* 1–1
    - displaying on other workstation • *Debugger,* 1–36
- DEC_CRT2 mode • *VAXTPU,* C–3
- "Dec_crt2" string constant parameter to GET_INFO • *VAXTPU,* 7–197
- DEC_CRT mode • *VAXTPU,* C–2
- "Dec_crt" string constant parameter to GET_INFO • *VAXTPU,* 7–197
- Default
  - condition handlers • *Routines Intro,* 2–51
- DEFAULT clause
  - for DEFINE TYPE statement • *Command Def,* CDU–28
  - for PARAMETER clause • *Command Def,* CDU–23, CDU–32
  - for QUALIFIER clause • *Command Def,* CDU–25, CDU–33
  - for VALUE clause • *Command Def,* CDU–24, CDU–26, CDU–29, CDU–33, CDU–34
- .DEFAULT directive • *MACRO,* 6–19
- Default directory
  - fetching in VAXTPU • *VAXTPU,* 7–206
  - setting in VAXTPU • *VAXTPU,* 7–366

Default displacement length directive (.DEFAULT) • *MACRO*, 6–19
Default-extension option • *File Applications*, 4–31
Default extension quantity • *File Def Language*, FDL–20
Default extension quantity field
  See FAB$W_DEQ field
Default file extension quantity field in XABFHC
  See XAB$W_DXQ field
Default file naming algorithm
  buffer change journal • *VAXTPU*, 1–12
Default file specification • *File Applications*, 5–4, 6–1 to 6–4, 9–7
  See also File specification
  journal file • *Patch*, PAT–29
  output image file • *Patch*, PAT–32
Default file specification string address field
  See FAB$L_DNA field
Default file specification string size field
  See FAB$B_DNS field
Default file type • *Librarian*, LIB–1, LIB–11
  for NCS definition files specified by /OUTPUT qualifier • *National Char Set*, NCS–39
  for NCS input files • *National Char Set*, NCS–21
  for NCS library • *National Char Set*, NCS–33
  for NCS library listing output file • *National Char Set*, NCS–34
  for NCS library specified by /COMPRESS qualifier • *National Char Set*, NCS–39
  for output files created by /MACRO qualifier • *National Char Set*, NCS–28
  for output files created by /OUTPUT qualifier • *National Char Set*, NCS–28
Default form • *System Services*, SYS–463
Default global buffer count field
  See XAB$W_GBC field
Default image map • *Linker*, 1–12
Default insertion
  in lieu of module replacement • *National Char Set*, NCS–40
Default library file type • *Librarian*, LIB–11
Default logical name table
  group • *System Services Intro*, 6–6
  job • *System Services Intro*, 6–5
  process • *System Services Intro*, 6–4
  system • *System Services Intro*, 6–6
Default map • *Linker*, 5–1
  module information in • *Linker*, 5–2, 5–3
  sections in • *Linker*, 5–2
  symbols cross-referenced in • *Linker*, LINK–5
Default output file name
  ANALYZE/RMS_FILE • *Analyze/RMS_File*, ARMS–16
Default patch area • *Patch*, PAT–18
Default protection • *File Def Language*, FDL–23
Default protection ACE • *System Services Intro*, 3–21
/DEFAULT qualifier • *Debugger*, CD–86
Default result
  vector arithmetic exceptions • *MACRO*, 10–6, 10–30, 10–68
$DEFAULTS$ buffer • *VAXTPU*, 4–32
Default system library
  linker's search of • *Linker*, LINK–29
Default system macro library • *System Services Intro*, 2–5
Default user library
  definition of • *Linker*, LINK–21
  linker's search of • *Linker*, LINK–21, LINK–22, LINK–29
Default values
  AREA • *File Def Language*, FDL–6
  DATE • *File Def Language*, FDL–15
  FILE • *File Def Language*, FDL–16
  key • *File Def Language*, FDL–26
  overriding with /COMPRESS qualifier • *National Char Set*, NCS–24
  RECORD • *File Def Language*, FDL–33
  SYSTEM • *File Def Language*, FDL–38
DEFAULT_DIRECTORY parameter to SET built-in procedure • *VAXTPU*, 7–366
“default_directory” string constant parameter to GET_INFO • *VAXTPU*, 7–206
DEFAULT_NAME attribute • *File Def Language*, FDL–19
$DEFEND macro • *Device Support (B)*, 1–70, 2–15
  example • *Device Support (B)*, 2–16
Deferred write option
  See FAB$V_DFW option
Deferred-write processing • *File Applications*, 9–9
DEFERRED_WRITE attribute • *File Def Language*, FDL–19
DEFERRED_WRITE secondary attribute • *File Applications*, 7–19, 7–20
DEFINE command • *Debugger*, 8–6, CD–47; *Linker*, LINK–21; *Patch*, PAT–50; *System Services Intro*, 6–2; *File Applications*, 4–14, 6–15; *System Dump Analyzer*, SDA–43
  creating user-defined symbols • *Patch*, PAT–5
  displaying default qualifiers for • *Debugger*, CD–216
  examples • *Patch*, PAT–51
  setting default qualifiers for • *Debugger*, CD–138

DEFINE command (Cont.)
  symbols defined • *Patch*, PAT–11
  /TRANSLATION_ATTRIBUTES qualifier • *File Applications*, 5–7
/DEFINED qualifier • *Debugger*, CD–250
"Defined" string constant parameter to GET_INFO • *VAXTPU*, 7–190
DEFINE/KEY command • *Debugger*, 8–8, CD–50
DEFINE/PROCESS_GROUP command • *Debugger*, 10–14, CD–54
DEFINE SYNTAX statement
  example • *Command Def*, CDU–5, CDU–27
  format • *Command Def*, CDU–5
  table of syntax changes • *Command Def*, CDU–20 to CDU–22
  with DISALLOW and NODISALLOWS clauses • *Command Def*, CDU–22
  with IMAGE clause • *Command Def*, CDU–23
  with PARAMETER and NOPARAMETER clauses • *Command Def*, CDU–23
  with PARAMETER clause • *Command Def*, CDU–21
  with QUALIFIER and NOQUALIFIERS clauses • *Command Def*, CDU–24
  with ROUTINE clause • *Command Def*, CDU–26
  with SYNTAX keyword • *Command Def*, CDU–28
DEFINE TYPE statement
  acceptable keyword clauses • *Command Def*, CDU–28
  acceptable type-clause • *Command Def*, CDU–28
  defining qualifier keywords • *Command Def*, CDU–30
  format • *Command Def*, CDU–7
  keywords referenced by VALUE • *Command Def*, CDU–28
  with DEFAULT clause • *Command Def*, CDU–28
  with DEFINE VERB statement • *Command Def*, CDU–7
  with LABEL clause • *Command Def*, CDU–28
  with NEGATABLE and NONNEGATABLE clauses • *Command Def*, CDU–28
  with SYNTAX clause • *Command Def*, CDU–28
  with VALUE clause • *Command Def*, CDU–7
DEFINE VERB statement
  example • *Command Def*, CDU–7, CDU–8
  format • *Command Def*, CDU–8
  with DEFAULT clause • *Command Def*, CDU–30
  with DEFINE SYNTAX statement • *Command Def*, CDU–6
  with DISALLOW and NODISALLOWS clauses • *Command Def*, CDU–31
  with IMAGE clause • *Command Def*, CDU–31
DEFINE VERB statement (Cont.)
  with PARAMETER and NOPARAMETERS clauses • *Command Def*, CDU–32
  with QUALIFIER and NOQUALIFIERS clauses • *Command Def*, CDU–33
  with ROUTINE clause • *Command Def*, CDU–35
  with SYNONYM clause • *Command Def*, CDU–35
DEFINE_KEY built-in procedure • *VAXTPU*, 7–100 to 7–104
DEFINE_WIDGET_CLASS built-in procedure • *VAXTPU*, 7–105
  example of use • *VAXTPU*, B–4 to B–11
$DEFINI macro • *Device Support (B)*, 1–70, 2–16
  example • *Device Support (B)*, 2–16
Definition
  built-in • *National Char Set*, NCS–7
Definition file
  characteristics • *National Char Set*, NCS–4
  example • *National Char Set*, NCS–5
  format • *National Char Set*, NCS–4
  generated by /OUTPUT qualifier • *National Char Set*, NCS–39
  how to build • *National Char Set*, NCS–4
  language notation • *National Char Set*, NCS–6
  naming • *National Char Set*, NCS–4
  output from NCS library
    See /OUTPUT qualifier
  structure • *National Char Set*, NCS–4
Definition module
  deleting from NCS library
    See /DELETE qualifier
  extracting from NCS library
    See /EXTRACT qualifier
  inserting in NCS library
    See /INSERT qualifier
  replacing
    See /REPLACE qualifier
  specifying name length • *National Char Set*, NCS–24
Definition path • *Command Def*, CDU–12
Definition statements • *Message*, MSG–3
$DEF macro • *Device Support (B)*, 1–70, 2–14
  example • *Device Support (B)*, 2–16
DELETE access • *File Def Language*, FDL–23
DELETE attribute • *File Def Language*, FDL–3, FDL–37
DELETE built-in procedure • *VAXTPU*, 7–107 to 7–110
DELETE command • *Debugger*, 8–6, CD–57; *Patch*, PAT–52; *File Applications*, 10–28; *File Def Language*, FDL–60

- Delete file function • *I/O User's I*, 1–29
- DELETE key • *I/O User's I*, 8–4
- DELETE/KEY command • *Debugger*, 8–9, CD–59
- Delete on close option
  - See FAB$V_DLT option
- /DELETE qualifier • *Command Def*, CDU–39; *Librarian*, LIB–21
  - for deleting definition modules from NCS library • *National Char Set*, NCS–27
  - LIBRARY command • *Programming Resources*, 5–2
- DELETE secondary attribute • *File Applications*, 7–3
- Delete service • *File Applications*, 8–2, 8–5; *RMS*, RMS–21
  - condition values • *RMS*, RMS–22
    - See also Completion status code
  - control block input fields • *RMS*, RMS–22
  - control block output fields • *RMS*, RMS–22
  - high-level language equivalents • *File Applications*, 8–2
  - program example • *RMS*, 4–19
  - requirements • *RMS*, RMS–22
  - run-time options • *File Applications*, 9–20
  - use restrictions • *RMS*, RMS–21
- Delete service option
  - See FAB$V_DEL option
- Delete sharing option
  - See FAB$V_SHRDEL option
- DELETE_ON_CLOSE attribute • *File Def Language*, FDL–19, FDL–24
- Deleting a PPL$ application • *RTL Parallel Processing*, 2–1, 2–2
- Deleting a subordinate • *RTL Parallel Processing*, 2–3
- Deleting records • *VAXTPU*, 6–5
- Deletion
  - buffer • *VAXTPU*, 2–4
  - line terminator • *VAXTPU*, 7–28
  - marker • *VAXTPU*, 2–10
  - operations • *RTL Screen Management*, 2–7
  - range • *VAXTPU*, 2–22, 7–70
  - subprocess • *VAXTPU*, 7–67
  - VAXTPU structure • *VAXTPU*, 7–109
  - window • *VAXTPU*, 2–28
- Delimiters • *Patch*, PAT–20, PAT–23
  - ASCII data entry • *Patch*, PAT–16
  - for specifying multiple definition modules • *National Char Set*, NCS–27, NCS–28, NCS–32, NCS–38
  - for specifying multiple input files • *National Char Set*, NCS–21
  - string argument • *MACRO*, 4–3
- Delimiters (Cont.)
  - using in control block arguments • *RMS*, 3–5, 3–6, 3–7
- DEL option • *File Def Language*, FDL–3, FDL–37
- DELQA driver
  - See Ethernet/802 drivers
- DELTA
  - See Delta/XDelta Utility
- Delta time • *Programming Resources*, 3–23; *System Services Intro*, 9–2
  - as input to SYS$BINTIM • *System Services*, SYS–28
  - converting to numeric • *System Services*, SYS–366
  - examples • *System Services Intro*, 9–4
  - in system format • *System Services Intro*, 9–3
- DELTA/XDELTA
  - See Delta/XDelta Utility
- Delta/XDelta Utility (DELTA/XDELTA) • *Programming Resources*, 1–15; *Device Support (A)*, 13–1 to 13–22
  - base register • *Device Support (A)*, 13–13
    - predefined • *Device Support (A)*, 13–13
    - X4 • *Device Support (A)*, 13–13
    - X5 • *Device Support (A)*, 13–13
    - XE • *Device Support (A)*, 13–13
    - XF • *Device Support (A)*, 13–13
  - changing contents of location using • *Device Support (A)*, 13–15, 13–16
  - closing location using • *Device Support (A)*, 13–16
  - commands
    - executing string • *Device Support (A)*, 13–19, 13–20
    - indirect • *Device Support (A)*, 13–17
    - predefined in XE and XF • *Device Support (A)*, 13–13
    - summary • *Device Support (A)*, 13–10 to 13–12
  - depositing command string in system patch space for use by • *Device Support (A)*, 13–20
  - displaying contents of address range using • *Device Support (A)*, 13–16
  - displaying contents of location using • *Device Support (A)*, 13–16
  - exiting from DELTA • *Delta/XDelta*, DELTA–2
  - exiting from XDELTA • *Delta/XDelta*, DELTA–8
  - expressions • *Device Support (A)*, 13–12
  - formats
    - address display • *Device Support (A)*, 13–15
    - instruction display • *Device Support (A)*, 13–16
  - guidelines • *Device Support (A)*, 13–21 to 13–22
  - invoking DELTA • *Delta/XDelta*, DELTA–1

- Delta/XDelta Utility (DELTA/XDELTA) (Cont.)
  - invoking XDELTA • *Delta/XDelta,* DELTA–2
  - prefixes
    - G • *Device Support (A),* 13–13
    - H • *Device Support (A),* 13–13
  - setting PC with • *Device Support (A),* 13–18
  - stepping through code with • *Device Support (A),* 13–19
  - symbols
    - period (.) • *Device Support (A),* 13–13
    - Q • *Device Support (A),* 13–13, 13–16, 13–17
  - using in multiprocessing environment • *Device Support (A),* 13–7, E–20
  - values • *Device Support (A),* 13–12
- DELUA driver
  - See Ethernet/802 drivers
- Demand-zero compression
  - cessation of • *Linker,* 3–10
  - conditions for • *Linker,* 6–19
  - control of by option • *Linker,* 1–8, 3–7
  - definition of • *Linker,* 1–8, 3–7
- Demand-zero image section • *Linker,* 1–8, 3–7
- Dependences
  - vector results • *MACRO,* 10–24
- Deposit
  - DEPOSIT command • *Debugger,* 4–3, CD–61
  - instruction • *Debugger,* 4–22, 11–13
    - with DECwindows • *Debugger,* 1–26
  - into address • *Debugger,* 4–25
    - with DECwindows • *Debugger,* 1–26
  - into register • *Debugger,* 4–23, 11–4
    - with DECwindows • *Debugger,* 1–26
  - into variable • *Debugger,* 4–3, 4–15
    - with DECwindows • *Debugger,* 1–25
  - into vector register • *Debugger,* 11–4
  - vector instruction • *Debugger,* 11–13
- Deposit ASCII String command • *Delta/XDelta,* DELTA–37
- DEPOSIT command • *Debugger,* 4–3, CD–61; *Patch,* PAT–55
  - patch area operations • *Patch,* PAT–18
  - /PATCH_AREA • *Patch,* PAT–57
  - with VERIFY command • *Patch,* PAT–91
- DEPTH attribute • *File Def Language,* FDL–5
- DEQNA driver
  - See Ethernet/802 drivers
- Descriptor • *RTL String Manipulation,* 2–7
  - analysis of • *RTL String Manipulation,* 2–4
  - array • *Routines Intro,* 2–25
  - class and data type • *RTL Intro,* 3–10
  - class codes • *Routines Intro,* 1–11
    - facility-specific • *Routines Intro,* 2–43
- Descriptor
  - class codes (Cont.)
    - reserved • *Routines Intro,* 2–44
  - decimal string • *Routines Intro,* 2–30
  - dynamic string • *Routines Intro,* 2–24
  - fields of • *RTL Intro,* 3–7
  - fixed-length • *Routines Intro,* 2–23
  - format • *Routines Intro,* 2–21
    - DSC$A_POINTER • *Routines Intro,* 2–23
    - DSC$B_CLASS • *Routines Intro,* 2–23
    - DSC$B_DTYPE • *Routines Intro,* 2–23
    - DSC$K_CLASS_A • *Routines Intro,* 2–25
    - DSC$K_CLASS_D • *Routines Intro,* 2–24
    - DSC$K_CLASS_J • *Routines Intro,* 2–29
    - DSC$K_CLASS_NCA • *Routines Intro,* 2–31
    - DSC$K_CLASS_P • *Routines Intro,* 2–29
    - DSC$K_CLASS_S • *Routines Intro,* 2–23
    - DSC$K_CLASS_SB • *Routines Intro,* 2–41
    - DSC$K_CLASS_SD • *Routines Intro,* 2–30
    - DSC$K_CLASS_UBA • *Routines Intro,* 2–38
    - DSC$K_CLASS_UBS • *Routines Intro,* 2–37
    - DSC$K_CLASS_UBSB • *Routines Intro,* 2–42
    - DSC$K_CLASS_V • *Routines Intro,* 2–25
    - DSC$K_CLASS_VS • *Routines Intro,* 2–34
    - DSC$K_CLASS_VSA • *Routines Intro,* 2–35
    - DSC$W_LENGTH • *Routines Intro,* 2–23
    - prototype • *Routines Intro,* 2–22
  - label • *Routines Intro,* 2–29
  - noncontiguous array • *Routines Intro,* 2–31
  - patch area • *Patch,* PAT–18
  - procedure • *Routines Intro,* 2–29
  - string with bounds • *Routines Intro,* 2–41
  - unaligned bit array • *Routines Intro,* 2–38
  - unaligned bit string • *Routines Intro,* 2–37
  - unaligned bit string with bounds • *Routines Intro,* 2–42
  - variable buffer • *Routines Intro,* 2–25
  - varying string • *Routines Intro,* 2–34
  - varying string array • *Routines Intro,* 2–35
- Design graphics mode • *File Applications,* 4–11
- Design mnemonic • *File Applications,* 4–14
- Design stage • *Modular Procedures,* 2–1
- Destination file specification
  - requirement • *National Char Set,* NCS–36
- DESVA driver
  - See Ethernet/802 drivers
- Detached cursor
  - defining routine to handle • *VAXTPU,* 7–367
  - fetching action routine to handle • *VAXTPU,* 7–197
  - fetching reason for • *VAXTPU,* 7–198
- Detached process • *System Services Intro,* 8–2, 8–7; *System Services,* SYS–99

Detached process (Cont.)
creating • *Programming Resources*, 2–7
definition of • *RTL Parallel Processing*, 1–2
DETACHED_ACTION parameter to SET built-in • *VAXTPU*, 7–367
“detached_action” string constant parameter to GET_INFO • *VAXTPU*, 7–197
“detached_reason” string constant parameter to GET_INFO • *VAXTPU*, 7–198
DEUNA driver
See Ethernet/802 drivers
DEV$V_AVL • *Device Support (A)*, 18–22
DEV$V_ELG • *Device Support (A)*, 11–9; *Device Support (B)*, 3–8
DEV$V_NET • *Device Support (A)*, 18–13
DEV$V_RED • *Device Support (A)*, 18–22
$DEVDEF macro • *Device Support (B)*, 1–74, 1–75
source of DEV field bit definitions • *RMS*, 5–7
Developing a parallel processing application
items to consider • *RTL Parallel Processing*, 5–1
Device
See also Device unit
allocating • *System Services Intro*, 7–22; *System Services*, SYS–12
allocation class • *Device Support (B)*, 1–28
associated mailbox • *Device Support (B)*, 1–77
bus • *Device Support (B)*, 1–76
byte-addressable • *Device Support (A)*, 14–22
card reader • *Device Support (B)*, 1–76
cluster accessible • *Device Support (B)*, 1–73
cluster available • *Device Support (B)*, 1–75
deallocating • *System Services Intro*, 7–24; *System Services*, SYS–117
default name • *System Services Intro*, 7–27
Digital-supplied • *Device Support (A)*, 12–15
directory structured • *Device Support (B)*, 1–74
disk • *Device Support (B)*, 1–76, 3–51, 3–95
displaying SDA information • *System Dump Analyzer*, SDA–98
dual-pathed • *System Services*, SYS–207
dual-ported • *Device Support (B)*, 1–74, 1–75
file structured • *Device Support (A)*, 2–3, 4–10; *Device Support (B)*, 1–28, 1–74
getting information about • *System Services Intro*, 7–27
asynchronously • *System Services*, SYS–203
synchronously • *System Services*, SYS–221
implicit allocation • *System Services Intro*, 7–23
input • *Device Support (B)*, 1–75
line printer • *Device Support (B)*, 1–76
lock name • *System Services*, SYS–210
mailbox • *Device Support (B)*, 1–75, 1–76

Device (Cont.)
mounted • *Device Support (B)*, 1–75, 1–78
mounted foreign • *Device Support (B)*, 1–75
name • *System Services Intro*, 7–26
network • *Device Support (B)*, 1–74
offsettable • *Device Support (A)*, 16–10
on VAXBI bus • *Device Support (A)*, 16–2
output • *Device Support (B)*, 1–75
protection • *System Services Intro*, 7–6
random access • *Device Support (B)*, 1–75
real time • *Device Support (B)*, 1–75, 1–76
record oriented • *Device Support (B)*, 1–74
reference count • *Device Support (B)*, 1–79
SCSI • *Device Support (A)*, 16–30
sequential block-oriented • *Device Support (B)*, 1–74
served • *System Services*, SYS–214
shareable • *Device Support (B)*, 1–75
spooled • *Device Support (B)*, 1–74
synchronous communications • *Device Support (B)*, 1–76
tape • *Device Support (B)*, 1–76, 3–95
terminal • *Device Support (B)*, 1–74, 1–76
timed out • *Device Support (B)*, 1–78
word-aligned • *Device Support (A)*, 14–18
workstation • *Device Support (B)*, 1–76
Device access
controlling through access control lists • *Utility Routines*, ACL–1
Device activation bit mask • *Device Support (A)*, 8–4
Device affinity • *Device Support (B)*, 1–75, 3–71
Device allocation lock • *Device Support (B)*, 1–73
DEVICE attribute • *File Def Language*, FDL–38
Device characteristics • *Device Support (A)*, 7–9; *Device Support (B)*, 1–74 to 1–75
asynchronous DDCMP driver • *I/O User's II*, 5–2
card reader • *I/O User's I*, 2–5
disk • *I/O User's I*, 3–22
DMC11/DMR11 driver • *I/O User's II*, 1–3
DMP11/DMF32 driver • *I/O User's II*, 2–3
DR11–W/DRV11–WA driver • *I/O User's II*, 3–8
DR32 driver • *I/O User's II*, 4–3
Ethernet/802 drivers • *I/O User's II*, 6–14
line printer • *I/O User's I*, 5–3
LPA11-K device • *I/O User's I*, 4–5
magnetic tape • *I/O User's I*, 6–11
mailbox • *I/O User's I*, 7–4
pseudoterminal • *I/O User's I*, 9–3
retrieving • *Device Support (B)*, 3–49
setting • *Device Support (B)*, 3–50 to 3–51
specifying • *Device Support (A)*, 6–3; *Device Support (B)*, 2–25

Device characteristics (Cont.)
terminal • *I/O User's I*, 8–20
Device characteristics field
See FAB$L_DEV field
Device class • *Device Support (B)*, 1–76
specifying • *Device Support (A)*, 6–3; *Device Support (B)*, 2–25
Device controller • *Device Support (A)*, 1–5, 1–6; *Device Support (B)*, 1–19
See also Controller initialization routine
See also MBA
initializing • *Device Support (A)*, 11–1
intelligent • *Device Support (A)*, 1–22
multiunit • *Device Support (A)*, 3–26, 4–6, 4–16, 8–2, 8–6, 9–8; *Device Support (B)*, 1–36, 1–74, 1–77
number of units created for • *Device Support (A)*, 12–6; *Device Support (B)*, 2–22
number of units supported by • *Device Support (B)*, 1–34, 1–36, 1–37, 2–22
reinitializing • *Device Support (B)*, 2–22
single-unit • *Device Support (A)*, 3–26, 4–7, 10–2, 11–2, 11–3, 12–2; *Device Support (B)*, 1–36
status • *Device Support (B)*, 1–21
synchronizing access to • *Device Support (A)*, 3–16
Device controller channel wait queue • *Device Support (A)*, 3–27
Device controller data channel • *Device Support (A)*, 4–6 to 4–7, 15–14, 15–15
See also Secondary controller data channel
obtaining ownership of • *Device Support (A)*, 3–26, 4–6, 8–2 to 8–4; *Device Support (B)*, 1–36, 2–62, 3–100 to 3–101
owner • *Device Support (A)*, 4–7
releasing • *Device Support (A)*, 3–27, 8–6, 10–2; *Device Support (B)*, 2–54, 3–86
releasing before waiting for interrupt • *Device Support (B)*, 3–105
relinquishing ownership • *Device Support (B)*, 2–104
requesting • *Device Support (A)*, 8–2
retaining ownership • *Device Support (B)*, 2–104
retaining while waiting for interrupt • *Device Support (B)*, 3–105
unavailability • *Device Support (A)*, 8–3
Device controller data channel wait queue • *Device Support (A)*, 8–3; *Device Support (B)*, 1–21, 3–86, 3–91, 3–101
Device database • *Device Support (A)*, 3–6, 3–16, E–9
synchronizing access to • *Device Support (A)*, 3–22; *Device Support (B)*, 2–17 to 2–18

Device data block
See DDB
Device driver • *Device Support (A)*, 1–1
assembling with SYS$LIBRARY:LIB.MLB • *Device Support (A)*, 12–1, E–7
asynchronous nature • *Device Support (A)*, 1–1, 1–9, 5–1
base address of driver prologue table (DPT) • *System Dump Analyzer*, SDA–13
branching on adapter characteristics • *Device Support (B)*, 2–2 to 2–4
branching on processor type • *Device Support (B)*, 2–9 to 2–11
calculating base address • *Device Support (A)*, 13–7
coding conventions • *Device Support (A)*, 5–1 to 5–3, 12–1, 13–22, 13–22 to 13–23
components • *Device Support (A)*, 1–2 to 1–4, 5–1
context • *Device Support (A)*, 1–7 to 1–9
converting uniprocessing to multiprocessing • *Device Support (A)*, E–8 to E–20
debugging • *Device Support (A)*, 13–1 to 13–22
displaying address of • *Device Support (A)*, 12–12
entry points • *Device Support (A)*, 1–2, 6–3 to 6–4; *Device Support (B)*, 1–29, 4–1 to 4–24
example • *Device Support (A)*, C–1 to C–29, D–1 to D–26
flow • *Device Support (A)*, 1–9, 1–23 to 1–25
for generic VAXBI device • *Device Support (A)*, 16–1 to 16–30; *Device Support (B)*, 3–107
for MASSBUS device • *Device Support (A)*, 15–1 to 15–17
for Q22 bus device • *Device Support (A)*, 14–1 to 14–36
for UNIBUS device • *Device Support (A)*, 14–1 to 14–36
functions • *Device Support (A)*, 1–2
hardware considerations • *Device Support (A)*, 1–10 to 1–20
implementing a conditional wait • *Device Support (B)*, 2–92, 2–94
linking with SYS$SYSTEM:SYS.STB • *Device Support (A)*, 12–1, 13–7, E–8
loading • *Device Support (A)*, 6–1, 11–3 to 11–5, 12–1 to 12–23, 13–5, 15–7 to 15–8; *Device Support (B)*, 1–33
locating • *System Dump Analyzer*, SDA–13
locating a failing instruction • *System Dump Analyzer*, SDA–24
machine independence • *Device Support (A)*, 1–10, 5–5 to 5–6, 14–16; *Device Support (B)*, 2–2 to 2–4, 2–9 to 2–11

Device driver (Cont.)
- maximum number of supported units • *Device Support (A)*, 6–2
- multiprocessor • *Device Support (A)*, 12–13, E–1, E–3
- name • *Device Support (A)*, 4–8, 6–2, 12–3, 12–6, 12–7, 12–12; *Device Support (B)*, 1–28, 1–34, 2–22
- program sections • *Device Support (A)*, 6–4, 12–1, 13–7; *Device Support (B)*, 2–13, 2–21
- reloading • *Device Support (A)*, 12–7 to 12–8
- size • *Device Support (A)*, 5–1; *Device Support (B)*, 1–33
- storing data from • *Device Support (A)*, 5–1
- suspending • *Device Support (A)*, 2–6, 8–6 to 8–7, 14–24; *Device Support (B)*, 1–73
- synchronization flow • *Device Support (A)*, 3–17 to 3–21
- synchronization methods used by • *Device Support (A)*, 1–7, 3–1 to 3–27
- template for • *Device Support (A)*, A–1 to A–10
- uniprocessor • *Device Support (A)*, 12–13, E–1, E–3
- unloading • *Device Support (B)*, 1–33, 2–22
- updating old code • *Device Support (A)*, E–1

Device driver image • *Patch*, PAT–3, PAT–19

Device driver routine
- address • *System Dump Analyzer*, SDA–99

Device identification field
- See NAM$T_DVI field

Device interrupt • *Device Support (A)*, 1–6, 3–6, 4–16, 9–1 to 9–8, 14–26 to 14–34
- See also Interrupt service routine
- destination for VAXBI node • *Device Support (A)*, 16–10
- direct-vector • *Device Support (A)*, 14–3, 14–27, 14–29, 14–31; *Device Support (B)*, 1–7, 1–8, 1–25, 2–3
- disabling • *Device Support (A)*, 5–4, 10–4
- enabling • *Device Support (A)*, 2–5, 11–2
- expected • *Device Support (A)*, 8–7, 9–3 to 9–4; *Device Support (B)*, 1–77, 3–105
- multilevel Q22 bus • *Device Support (A)*, 14–31, 14–33 to 14–36; *Device Support (B)*, 1–22
- non-direct-vector • *Device Support (A)*, 14–3, 14–28, 14–29, 14–31; *Device Support (B)*, 1–7, 1–25
- on MASSBUS • *Device Support (A)*, 15–9
- servicing • *Device Support (A)*, 2–6 to 2–7
- unsolicited • *Device Support (A)*, 9–4 to 9–8; *Device Support (B)*, 1–30

Device interrupt (Cont.)
- waiting for • *Device Support (A)*, 2–5 to 2–6, 4–16, 8–6 to 8–7, 14–24; *Device Support (B)*, 2–105, 3–104 to 3–106

Device interrupt vector • *Device Support (A)*, 14–26, 16–9, 16–10 to 16–11
- connecting to • *Device Support (A)*, 19–7 to 19–25
- for generic VAXBI device • *Device Support (A)*, 16–15
- multiple • *Device Support (A)*, 14–31, 16–9
- specifying address • *Device Support (A)*, 12–6
- specifying multiple • *Device Support (A)*, 12–6

Device IPL • *Device Support (A)*, 3–6, 9–1; *Device Support (B)*, 1–77, 2–17 to 2–18
- specifying • *Device Support (A)*, 6–2; *Device Support (B)*, 2–25

DEVICE keyword
- with FILE_PARSE • *VAXTPU*, 7–140
- with FILE_SEARCH • *VAXTPU*, 7–143

Device lock • *Device Support (A)*, 3–6, 3–13, 3–16 to 3–17, 8–5; *Device Support (B)*, 1–68, 1–77, 3–105
- See also Spin lock
- acquisition IPL • *Device Support (B)*, 3–113
- address • *Device Support (A)*, 3–22; *Device Support (B)*, 1–22, 1–36, 1–74
- multiple acquisition of • *Device Support (B)*, 2–19, 3–117
- obtaining • *Device Support (A)*, 3–10; *Device Support (B)*, 2–17 to 2–18, 3–110, 3–113
- ownership • *Device Support (A)*, 3–17
- rank • *Device Support (A)*, 3–17
- releasing • *Device Support (A)*, 3–10; *Device Support (B)*, 2–19 to 2–20, 3–115
- restoring • *Device Support (B)*, 2–19, 3–117

DEVICELOCK macro • *Device Support (A)*, 3–9, 3–10, E–4, E–9, E–10, E–11; *Device Support (B)*, 2–17 to 2–18, 2–66, 2–104, 3–110, 3–113
- example • *Device Support (B)*, 2–18, 2–20, 2–66
- used by interrupt service routine • *Device Support (A)*, 9–3

Device mode • *Device Support (A)*, 7–9

Device name • *Device Support (A)*, 1–5; *Device Support (B)*, 1–28

Device name address descriptor
- See NAM$L_DEV descriptor

Device name address field
- See NAM$L_DEV field

Device name length field
- See NAM$B_DEV field

Device name size descriptor
- See NAM$B_DEV descriptor

Device registers • *Device Support (A)*, 1–6, 1–21 to 1–22, 14–23
- accessing • *Device Support (A)*, 2–5, 4–7, 13–21 to 13–22, 14–4, 14–23, 16–5, 19–1; *Device Support (B)*, 1–25, 1–36, 2–17 to 2–18
- clearing error status • *Device Support (A)*, 11–2
- modification by power failure • *Device Support (A)*, 8–5
- modifying • *Device Support (A)*, 5–4
- of LP11 printer • *Device Support (A)*, 2–5
- rules for referencing • *Device Support (A)*, 5–3 to 5–5, 14–4
- saving the value of • *Device Support (A)*, 11–11; *Device Support (B)*, 4–16
- synchronizing access to • *Device Support (A)*, 3–6, 3–16, 8–5

Device timeout
- See Timeout

Device timeout bit
- See UCB$V_TIMOUT

Device types • *Programming Resources*, 7–50; *Device Support (B)*, 1–76
- specifying • *Device Support (A)*, 6–3; *Device Support (B)*, 2–25

Device unit • *Device Support (A)*, 1–5; *Device Support (B)*, 1–68
- See also UCB
- See also Unit initialization routine
- activating • *Device Support (A)*, 2–5, 8–4 to 8–5, 14–23
- allocating • *Device Support (B)*, 1–74, 1–75, 1–77
- autoconfiguring • *Device Support (A)*, 12–22 to 12–23; *Device Support (B)*, 2–22
- busy indicator • *Device Support (B)*, 1–78
- CSR address • *Device Support (A)*, 12–11
- deaccessing • *Device Support (B)*, 1–12
- deallocating • *Device Support (B)*, 1–78
- description • *Device Support (A)*, 4–5
- error retry count • *Device Support (B)*, 1–79
- initializing • *Device Support (A)*, 11–1
- marking available • *Device Support (B)*, 1–75
- marking on line • *Device Support (A)*, 11–2; *Device Support (B)*, 1–78
- name • *Device Support (A)*, 4–8
- number • *Device Support (B)*, 1–77
- operations count • *Device Support (B)*, 3–95
- reference count • *Device Support (A)*, 11–7; *Device Support (B)*, 4–4
- reinitializing • *Device Support (B)*, 2–22
- status • *Device Support (A)*, 4–5; *Device Support (B)*, 1–77 to 1–79
- vector address • *Device Support (A)*, 12–11

DEVICEUNLOCK macro • *Device Support (A)*, 3–10, E–4, E–10, E–11; *Device Support (B)*, 2–19 to 2–20, 2–66, 3–115, 3–117
- example • *Device Support (B)*, 2–18, 2–20, 2–66
- issued by IOC$WFIKPCH and IOC$WFIRLCH • *Device Support (B)*, 3–105

device_name data type • *Routines Intro*, A–5t

DFL (data bucket fill size)
- program example • *RMS*, 4–8

DFW option • *File Def Language*, FDL–19

DHU11 device • *I/O User's I*, 8–1

DHV11 device • *I/O User's I*, 8–1

Diagnostic buffer • *Device Support (A)*, 4–20; *Device Support (B)*, 1–40, 1–42, 1–79, 1–83, 3–71
- copied to process space • *Device Support (B)*, 3–73
- filling • *Device Support (B)*, 3–69
- size • *Device Support (B)*, 1–30
- specifying • *Device Support (A)*, 4–10, 6–4

Diagnostic register
- See MBA$L_DR

Dial-up line • *I/O User's I*, 8–13

DIBOL
- See VAX DIBOL

DIFFERENCES/SLP DCL command • *SUMSLP*, SUM–3

DIGITAL Command Language
- See DCL

Digital-private escape sequence • *I/O User's I*, B–9

DIGITAL Storage Architecture disks • *I/O User's I*, 3–19

DIOLM (direct I/O limit) quota • *System Services Intro*, 7–3
- adjusting • *Device Support (A)*, 4–20
- charging • *Device Support (A)*, 4–9, 4–12
- checking • *Device Support (A)*, 4–9

Direct assignment statement • *MACRO*, 1–1, 3–17

Direct data path • *Device Support (A)*, 14–7, 14–10
- See also Data path
- functions • *Device Support (A)*, 14–10
- odd transfer • *Device Support (B)*, 1–8
- purging • *Device Support (A)*, 14–19, 14–24 to 14–25
- requesting • *Device Support (A)*, 14–18
- speed • *Device Support (A)*, 14–10

Direct I/O • *Device Support (A)*, 1–22, 1–23, 7–4, 16–19; *Device Support (B)*, 1–40, 1–79
- additional buffer regions for • *Device Support (B)*, 1–42 to 1–44
- checking accessibility of process buffer for • *Device Support (B)*, 3–43 to 3–44, 3–56 to 3–57

Direct I/O (Cont.)
FDT routines for • *Device Support (A)*, 7–6, 7–9
locking a process buffer for • *Device Support (B)*, 3–31 to 3–33, 3–34 to 3–36, 3–40 to 3–42, 3–45 to 3–47, 3–54 to 3–55, 3–58 to 3–60
postprocessing • *Device Support (B)*, 3–72
reasons for using • *Device Support (A)*, 1–22 to 1–23, 6–7, 6–8
unlocking process buffer • *Device Support (B)*, 3–109
Direct I/O count • *Convert*, CONV–24
Direct I/O quota • *I/O User's I*, 3–24, 6–13
Direct input/output operation • *Programming Resources*, 3–20
Direction
of buffer • *VAXTPU*, 7–85
setting • *VAXTPU*, 7–379
"Direction" string constant parameter to GET_INFO • *VAXTPU*, 7–171
Directive • *Message*, MSG–2; *MACRO*, 1–1, 6–1
See also Message Utility
as operator • *MACRO*, 2–3
.END • *Programming Resources*, 9–8
.FACILITY • *Programming Resources*, 9–7
general assembler • *MACRO*, 1–1, 6–1
macro • *MACRO*, 1–1, 6–1, 6–3
.SEVERITY • *Programming Resources*, 9–8
summary • *MACRO*, C–1
SYS$FAO • *System Services*, SYS–167
.TITLE • *Programming Resources*, 9–9
Direct memory access transfer
See DMA transfer
Directory • *File Applications*, 6–12
creating • *RTL Library*, LIB–36
default
fetching in VAXTPU • *VAXTPU*, 7–206
setting in VAXTPU • *VAXTPU*, 7–366
Directory address descriptor
See NAM$L_DIR descriptor
Directory entry
creation • *I/O User's I*, 1–26
protection • *I/O User's I*, 1–9
Directory identification field
See NAM$W_DID field
DIRECTORY keyword
with FILE_PARSE • *VAXTPU*, 7–140
with FILE_SEARCH • *VAXTPU*, 7–143
Directory logical name table
process • *System Services Intro*, 6–3
system • *System Services Intro*, 6–3
Directory lookup subfunction • *I/O User's I*, 1–7
Directory lookup subfunction (Cont.)
directory entry protection • *I/O User's I*, 1–9
Directory name length address field
See NAM$L_DIR field
Directory name length field
See NAM$B_DIR field
/DIRECTORY qualifier • *Debugger*, CD–224
Directory sequence number • *Device Support (B)*, 1–82, 1–83
Directory size descriptor
See NAM$B_DIR descriptor
Directory specification
normal • *File Applications*, 6–12 to 6–14
rooted • *File Applications*, 6–15 to 6–20
Directory tree • *File Applications*, 6–12
DIRECTORY_ENTRY attribute • *File Def Language*, FDL–19, FDL–20
DIRECTORY_ENTRY secondary attribute • *File Applications*, 4–28
/DIRECT qualifier • *Debugger*, CD–250
Direct-vector interrupt • *Device Support (A)*, 13–9, 14–3, 14–27, 14–29, 14–31; *Device Support (B)*, 1–7, 1–8, 1–25, 2–3
Disable assembler functions directive (.DISABLE) • *MACRO*, 6–21
DISABLE AST command • *Debugger*, 9–17, CD–68
Disabled fault
vector processor • *MACRO*, 10–31, 10–32
.DISABLE directive • *MACRO*, 6–21
DISALLOW clause • *Command Def*, CDU–9 to CDU–13
definition path • *Command Def*, CDU–12
for DEFINE SYNTAX statement • *Command Def*, CDU–22
for DEFINE VERB statement • *Command Def*, CDU–31
keyword path • *Command Def*, CDU–11
operators for • *Command Def*, CDU–13
DISCONNECT command • *I/O User's I*, 8–17
Disconnect feature
determining setting of • *Device Support (B)*, 2–75
enabling • *I/O User's I*, 11–13; *Device Support (A)*, 17–14; *Device Support (B)*, 2–88
Disconnect service • *File Applications*, 8–5; *RMS*, RMS–23
condition values • *RMS*, RMS–24
See also Completion status code
control block input fields • *RMS*, RMS–24
control block output fields • *RMS*, RMS–24
program example • *RMS*, 4–12
using with multiple RABs • *RMS*, RMS–24

- Disk
  - See also DSA disk
  - ACP control function • *I/O User's I*, 1–32
  - ACP operation
    - creating file • *I/O User's I*, 1–24
    - deaccessing file • *I/O User's I*, 1–28
  - available function • *I/O User's I*, 3–33
  - Backup Utility • *I/O User's I*, 3–21
  - compact disc • *I/O User's I*, 3–8
  - data check • *I/O User's I*, 3–15, 3–29, 3–30
  - device characteristics • *I/O User's I*, 3–22
  - driver • *I/O User's I*, 3–1
    - SCSI • *I/O User's I*, 3–22
    - VAXstation 2000 and MicroVAX 2000 • *I/O User's I*, 3–21
  - dual-pathed • *I/O User's I*, 3–11
    - DSA disks • *I/O User's I*, 3–14
  - dual porting • *I/O User's I*, 3–12
    - DSA disks • *I/O User's I*, 3–14
    - HSC disks • *I/O User's I*, 3–15
    - restrictions for use • *I/O User's I*, 3–13
  - error recovery • *I/O User's I*, 3–17
  - features • *I/O User's I*, 3–11
  - file attributes • *I/O User's I*, 3–16
  - function codes • *I/O User's I*, 3–24, 3–25, A–2
  - function modifiers
    - IO$M_DATACHECK • *I/O User's I*, 3–15, 3–29, 3–30
    - IO$M_DELDATA • *I/O User's I*, 3–30
    - IO$M_ERASE • *I/O User's I*, 3–27, 3–31
    - IO$M_INHRETRY • *I/O User's I*, 3–17, 3–29, 3–30
  - HSC40 controller • *I/O User's I*, 3–3
  - HSC50 controller • *I/O User's I*, 3–3
  - HSC70 controller • *I/O User's I*, 3–3
  - I/O functions • *I/O User's I*, 3–24
    - See also ACP-QIO interface
    - arguments • *I/O User's I*, 3–26 to 3–29
    - IO$_ACPCONTROL • *I/O User's I*, 1–32
    - IO$_AVAILABLE • *I/O User's I*, 3–33
    - IO$_FORMAT • *I/O User's I*, 3–31
    - IO$_PACKACK • *I/O User's I*, 3–32
    - IO$_READLBLK • *I/O User's I*, 3–29
    - IO$_READPBLK • *I/O User's I*, 3–29
    - IO$_READVBLK • *I/O User's I*, 3–29
    - IO$_SEARCH • *I/O User's I*, 3–31
    - IO$_SEEK • *I/O User's I*, 3–33
    - IO$_SENSECHAR • *I/O User's I*, 3–31
    - IO$_SENSEMODE • *I/O User's I*, 3–31
    - IO$_SETPRFPTH • *I/O User's I*, 3–34
    - IO$_UNLOAD • *I/O User's I*, 3–32
    - IO$_WRITECHECK • *I/O User's I*, 3–33

- Disk
  - I/O functions (Cont.)
    - IO$_WRITELBLK • *I/O User's I*, 3–30
    - IO$_WRITEPBLK • *I/O User's I*, 3–30
    - IO$_WRITEVBLK • *I/O User's I*, 3–30
  - I/O status block • *I/O User's I*, 3–36
  - KDA50 controller • *I/O User's I*, 3–3
  - KDB50 controller • *I/O User's I*, 3–3
  - KFQSA adapter • *I/O User's I*, 3–5
  - offset recovery • *I/O User's I*, 3–16
  - pack acknowledge function • *I/O User's I*, 3–32
  - port access mode • *I/O User's I*, 3–12
  - port selection • *I/O User's I*, 3–12
  - programming example • *I/O User's I*, 3–37
  - quotas • *I/O User's I*, 1–33 to 1–34, 3–24
  - RA60 • *I/O User's I*, 3–5
  - RA70 • *I/O User's I*, 3–5
  - RA90 • *I/O User's I*, 3–5
  - RB02 • *I/O User's I*, 3–6
  - RC25 • *I/O User's I*, 3–6
  - RCT (replacement and caching table) • *I/O User's I*, 3–20
  - RD53 • *I/O User's I*, 3–6
  - RD54 • *I/O User's I*, 3–6
  - read function • *I/O User's I*, 3–29
  - RF30 • *I/O User's I*, 3–7
  - RF31
    - failover • *I/O User's I*, 3–15
  - RF70
    - failover • *I/O User's I*, 3–15
  - RF71 • *I/O User's I*, 3–7
  - RM03 • *I/O User's I*, 3–7
  - RM05 • *I/O User's I*, 3–7
  - RP05 • *I/O User's I*, 3–7
  - RP06 • *I/O User's I*, 3–7
  - RP07 • *I/O User's I*, 3–7
  - RQDX3 controller • *I/O User's I*, 3–5
  - RRD40 CDROM • *I/O User's I*, 3–8
  - RRD50 CDROM • *I/O User's I*, 3–8
  - RX02 • *I/O User's I*, 3–8
  - RX06 cartridge • *I/O User's I*, 3–7
  - RX07 cartridge • *I/O User's I*, 3–7
  - RX23 flexible • *I/O User's I*, 3–9
  - RX33 flexible • *I/O User's I*, 3–10
  - RX50 flexible • *I/O User's I*, 3–10
  - RZ22 • *I/O User's I*, 3–10
  - RZ23 • *I/O User's I*, 3–10
  - RZ55 • *I/O User's I*, 3–10
  - SDI • *I/O User's I*, 3–5
  - search function • *I/O User's I*, 3–31
  - sector translation • *I/O User's I*, 3–18
  - seek operations • *I/O User's I*, 3–16, 3–33

- Disk (Cont.)
  - sense mode function • *I/O User's I*, 3–31
  - set density function • *I/O User's I*, 3–31
  - set preferred path function • *I/O User's I*, 3–34
  - SII integral adapter • *I/O User's I*, 3–4
  - skip sectoring • *I/O User's I*, 3–17
  - status returns • *I/O User's I*, A–3
  - supported devices • *I/O User's I*, 3–1 to 3–11
  - SYS$GETDVI returns • *I/O User's I*, 3–22
  - TU58 magnetic tape • *I/O User's I*, 3–10, 3–16, 3–29, 3–30, 3–31, 3–33
  - UDA50 disk adapter • *I/O User's I*, 3–3
  - unload function • *I/O User's I*, 3–32
  - use with Verify Utility • *I/O User's I*, 3–19, 3–21
  - VAXstation 2000 and MicroVAX 2000 driver • *I/O User's I*, 3–21
  - write check function • *I/O User's I*, 3–33
  - write function • *I/O User's I*, 3–30
- Disk block • *File Applications*, 3–6
- Disk class driver
  - disabling the loading of • *I/O User's I*, 11–10; *Device Support (A)*, 17–31
- Disk cluster boundary
  - determining allocation quantity • *RMS*, 5–3
- Disk cylinder • *File Applications*, 3–6
- Disk drive
  - compatibility for volume shadowing • *I/O User's I*, 10–3
- Disk driver • *Device Support (A)*, 7–9, 8–2, 8–6, 9–5; *Device Support (B)*, 1–78, 1–79
  - See also MASSBUS
  - See also MBA
  - ECC correction routine for • *Device Support (B)*, 3–67
  - pack acknowledgment in • *Device Support (A)*, 11–2
  - recording disk geometry in • *Device Support (A)*, 11–3
  - removing a disk volume in • *Device Support (A)*, 9–8
  - using local disk UCB extension • *Device Support (B)*, 1–69, 1–82 to 1–84
  - waiting for disk unit spinup in • *Device Support (A)*, 11–3
- Disk file
  - opening • *System Services Intro*, 11–9
- Disk model • *File Def Language*, FDL–38
- Disk quota • *File Applications*, 3–5
- Disk quotas • *I/O User's I*, 1–33
- Disk space
  - efficiency
    - See /DATA qualifier
- Disk space (Cont.)
  - recovering
    - See /COMPRESS qualifier
- Disk volume • *File Applications*, 3–6
  - mounting • *System Services Intro*, 7–24
  - transfer • *File Def Language*, FDL–23
- DISMOUNT command • *I/O User's I*, 1–32
- Dispatcher
  - exception • *System Services Intro*, 10–7
- Displacement deferred mode • *MACRO*, 5–9
  - operand specifier formats • *MACRO*, 8–22
- Displacement mode • *MACRO*, 5–8
  - operand specifier formats • *MACRO*, 8–21
- Display
  - See also Instruction
  - See also Source display
  - See also Window
  - attribute • *Debugger*, 7–3, 7–20, CD–123, CD–244
  - canceling • *Debugger*, 7–12, CD–20
  - contracting • *Debugger*, 7–13, CD–98
  - creating • *Debugger*, 7–13, CD–69
  - current • *Debugger*, 7–3, 7–20, CD–123
  - default configuration • *Debugger*, 7–2, 7–4
  - defined • *Debugger*, 7–2
  - DO display • *Debugger*, 7–16, 11–24
  - expanding • *Debugger*, 7–13, CD–98
  - extracting • *Debugger*, 7–22, CD–101
  - hiding • *Debugger*, 7–12, CD–71
  - identifying • *Debugger*, 7–12, CD–217
  - instruction display (INST) • *Debugger*, 7–8, 7–17
  - kind • *Debugger*, 7–3, 7–15, C–1
  - list • *Debugger*, 7–3, CD–217, C–6
  - moving • *Debugger*, 7–13, CD–110
  - output display (OUT) • *Debugger*, 7–7, 7–18
  - pasteboard • *Debugger*, 7–3, CD–74
  - predefined • *Debugger*, 7–4, C–3
  - process specific • *Debugger*, 10–16
  - prompt display (PROMPT) • *Debugger*, 7–7
  - register display (REG) • *Debugger*, 7–10, 7–18, 11–24
  - removing • *Debugger*, 7–12, CD–73
  - saving • *Debugger*, 7–22, CD–116
  - scrolling • *Debugger*, 7–11, CD–118
  - selecting • *Debugger*, 7–20, CD–123
  - showing • *Debugger*, 7–12, CD–69
  - VAXTPU definition • *VAXTPU*, 4–16
  - window • *Debugger*, 7–2, 7–14, C–7
- DISPLAY command • *Debugger*, 7–12, 7–13, CD–69
- Displaying version number • *VAXTPU*, 4–2
- Display modes
  - See also Entry and display modes

Display modes (Cont.)
  how to set • *Delta/XDelta,* DELTA–16
/DISPLAY qualifier • *File Def Language,* FDL–42, FDL–49; *VAXTPU,* 5–8
  See also /NODISPLAY
Display service • *RMS,* RMS–25
  condition values • *RMS,* RMS–28
  control block input fields • *RMS,* RMS–26
  control block output fields • *RMS,* RMS–26
  requirements • *RMS,* RMS–26
"Display" string constant parameter to GET_INFO • *VAXTPU,* 7–177, 7–206
Display value
  fetching • *VAXTPU,* 7–222
  setting for window • *VAXTPU,* 7–370
  setting records • *VAXTPU,* 7–448
Display Value of Expression command • *Delta/XDelta,* DELTA–42
DISPLAY_VALUE parameter to SET built-in procedure • *VAXTPU,* 7–370
"display_value" string constant parameter to GET_ INFO • *VAXTPU,* 7–186, 7–222
DIVB2 (Divide Byte 2 Operand) instruction • *MACRO,* 9–18
DIVB3 (Divide Byte 3 Operand) instruction • *MACRO,* 9–18
DIVD2 (Divide D_floating 2 Operand) instruction • *MACRO,* 9–113
DIVD3 (Divide D_floating 3 Operand) instruction • *MACRO,* 9–113
DIVF2 (Divide F_floating 2 Operand) instruction • *MACRO,* 9–113
DIVF3 (Divide F_floating 3 Operand) instruction • *MACRO,* 9–113
DIVG2 (Divide G_floating 2 Operand) instruction • *MACRO,* 9–113
DIVG3 (Divide G_floating 3 Operand) instruction • *MACRO,* 9–113
DIVH2 (Divide H_floating 2 Operand) instruction • *MACRO,* 9–113
DIVH3 (Divide H_floating 3 Operand) instruction • *MACRO,* 9–113
Divide-by-zero trap • *MACRO,* 8–16
Division
  complex number • *RTL General Purpose,* OTS–40
  extended precision • *RTL Library,* LIB–126
  packed decimal • *RTL General Purpose,* OTS–44, OTS–47
Division operator (/) • *System Dump Analyzer,* SDA–13
DIVL2 (Divide Long 2 Operand) instruction • *MACRO,* 9–18
DIVL3 (Divide Long 3 Operand) instruction • *MACRO,* 9–18
DIVP (Divide Packed) instruction • *MACRO,* 9–163
DIVW2 (Divide Word 2 Operand) instruction • *MACRO,* 9–18
DIVW3 (Divide Word 3 Operand) instruction • *MACRO,* 9–18
DLDRIVER.MAR • *Device Support (A),* C–1 to C–29
DLT option • *File Def Language,* FDL–20
DMA transfer • *Device Support (A),* 1–22, 5–5
  See also Data path
  See also Map registers
  byte-aligned • *Device Support (A),* 14–11
  detecting memory error during • *Device Support (A),* 14–25
  flow • *Device Support (A),* 1–23 to 1–25, 14–8
  for modify operation • *Device Support (B),* 3–31 to 3–33, 3–34 to 3–36
  for read operation • *Device Support (B),* 3–40 to 3–42, 3–45 to 3–47
  for write operation • *Device Support (B),* 3–54 to 3–55, 3–58 to 3–60
  longword-aligned 32-bit random-access • *Device Support (A),* 14–12, 14–14 to 14–15
  on Q22 bus • *Device Support (A),* 14–15 to 14–16, 14–19 to 14–26
  on UNIBUS • *Device Support (A),* 14–15 to 14–26
  on VAXBI bus • *Device Support (A),* 16–18 to 16–22
  postprocessing • *Device Support (A),* 14–16, 14–24 to 14–26
  start I/O routine • *Device Support (A),* 8–1 to 8–7
  using direct data path in • *Device Support (A),* 14–10
  using direct I/O in • *Device Support (A),* 6–8
  using I/O adapter resources in • *Device Support (A),* 14–2 to 14–15
DMB32 asynchronous/synchronous multiplexer • *Device Support (A),* 16–20
DMB32 device • *I/O User's I,* 8–1
DMC11/DMR11 driver
  attention AST • *I/O User's II,* 1–9
    enabling • *I/O User's II,* 1–7
  data
    message size • *I/O User's II,* 1–3, 1–6, 1–9
  DDCMP (DIGITAL Data Communications Message Protocol) • *I/O User's II,* 1–1
  device characteristics • *I/O User's II,* 1–3, 1–8
  driver • *I/O User's II,* 1–1
    capabilities • *I/O User's II,* 1–2
  error summary bits • *I/O User's II,* 1–5
  function codes • *I/O User's II,* 1–5, A–1

DMC11/DMR11 driver (Cont.)
function modifiers • *I/O User's II*, 1–6, 1–8
I/O functions • *I/O User's II*, 1–5 to 1–7
I/O status block • *I/O User's II*, 1–9
mailbox
disabling • *I/O User's II*, 1–6
enabling • *I/O User's II*, 1–6
message • *I/O User's II*, 1–9
format • *I/O User's II*, 1–2
type • *I/O User's II*, 1–2
usage • *I/O User's II*, 1–2
programming example • *I/O User's II*, 1–10
quota • *I/O User's II*, 1–3, 1–9
read function • *I/O User's II*, 1–5
receive-message blocks • *I/O User's II*, 1–8, 1–9
set characteristics function • *I/O User's II*, 1–7
set mode and shut down unit • *I/O User's II*, 1–8
set mode and start unit • *I/O User's II*, 1–8
set mode function • *I/O User's II*, 1–6, 1–7
start unit • *I/O User's II*, 1–8
status returns • *I/O User's II*, A–1
supported DMC11 options • *I/O User's II*, 1–1
SYS$GETDVI • *I/O User's II*, 1–3
unit and line status • *I/O User's II*, 1–4
unit characteristics • *I/O User's II*, 1–4
write function • *I/O User's II*, 1–6
DMF32 device • *I/O User's I*, 8–1
DMP11/DMF32 driver
AST service routine address • *I/O User's II*, 2–19
attention AST • *I/O User's II*, 2–19
characteristics
controller • *I/O User's II*, 2–9, 2–19
device • *I/O User's II*, 2–3
extended • *I/O User's II*, 2–11 to 2–12, 2–16 to 2–17
modifying • *I/O User's II*, 2–9
tributary • *I/O User's II*, 2–16, 2–19
character-oriented protocol • *I/O User's II*, 2–3, 2–12, 2–13
controller
mode • *I/O User's II*, 2–12
starting • *I/O User's II*, 2–9
DDCMP (DIGITAL Data Communications Message Protocol) • *I/O User's II*, 2–1
DDCMP controller counter parameter IDs • *I/O User's II*, 2–22
device characteristics • *I/O User's II*, 2–3
diagnostic support • *I/O User's II*, 2–23
read device status slot • *I/O User's II*, 2–25
read line unit modem status • *I/O User's II*, 2–24

DMP11/DMF32 driver
diagnostic support (Cont.)
set line unit modem status • *I/O User's II*, 2–24
DMC11-compatible operating mode • *I/O User's II*, 2–1
DMF32 driver • *I/O User's II*, 2–1
control • *I/O User's II*, 2–12
transmitter interface • *I/O User's II*, 2–14
DMF32 driver transmitter interface • *I/O User's II*, 2–14
DMP11 driver • *I/O User's II*, 2–1
driver capabilities • *I/O User's II*, 2–1
duplex modes • *I/O User's II*, 2–1, 2–2, 2–11, 2–12
enable attention AST • *I/O User's II*, 2–19
enable modem • *I/O User's II*, 2–9
errors • *I/O User's II*, 2–5
error summary bits • *I/O User's II*, 2–5
extended characteristics • *I/O User's II*, 2–11 to 2–12, 2–12, 2–16 to 2–17
framing routine interface • *I/O User's II*, 2–13
function codes • *I/O User's II*, 2–6, A–2
function modifiers • *I/O User's II*, 2–8 to 2–9, 2–15, 2–18 to 2–19, 2–24 to 2–25
HDLC bit stuff mode • *I/O User's II*, 2–3, 2–12, 2–15
I/O functions • *I/O User's II*, 2–7 to 2–9, 2–15, 2–19
I/O status block • *I/O User's II*, 2–25
LAPB controller counter parameter IDs • *I/O User's II*, 2–22
message size • *I/O User's II*, 2–3, 2–8, 2–10
modem
disabling line • *I/O User's II*, 2–18
status • *I/O User's II*, 2–24
modifying characteristics • *I/O User's II*, 2–9
multipoint
configuration • *I/O User's II*, 2–1
control station • *I/O User's II*, 2–1
parameter ID • *I/O User's II*, 2–10, 2–11, 2–12
point-to-point
configuration • *I/O User's II*, 2–1
station • *I/O User's II*, 2–1
polling time • *I/O User's II*, 2–12, 2–17
privilege • *I/O User's II*, 2–7
programming example • *I/O User's II*, 2–26
protocol • *I/O User's II*, 2–1, 2–3, 2–11, 2–12, 2–13
starting • *I/O User's II*, 2–15
stopping • *I/O User's II*, 2–18
quotas • *I/O User's II*, 2–3
read device status slot • *I/O User's II*, 2–25
read function • *I/O User's II*, 2–7

DMP11/DMF32 driver (Cont.)
read internal counters • *I/O User's II*, 2–20
read line unit modem status • *I/O User's II*, 2–24
sense mode function • *I/O User's II*, 2–19
set controller mode • *I/O User's II*, 2–9
characteristics • *I/O User's II*, 2–10
extended characteristics • *I/O User's II*, 2–11 to 2–12
message size • *I/O User's II*, 2–10, 2–12, 2–13
P1 buffer • *I/O User's II*, 2–10
P2 buffer • *I/O User's II*, 2–11
parameter ID • *I/O User's II*, 2–10
receive message blocks • *I/O User's II*, 2–10
set line unit modem status • *I/O User's II*, 2–23, 2–24
set mode function • *I/O User's II*, 2–9
set tributary mode • *I/O User's II*, 2–15
characteristics • *I/O User's II*, 2–16
extended characteristics • *I/O User's II*, 2–16 to 2–17
P1 buffer • *I/O User's II*, 2–16
P2 buffer • *I/O User's II*, 2–16
parameter ID • *I/O User's II*, 2–16
shutdown controller mode • *I/O User's II*, 2–18
shutdown tributary mode • *I/O User's II*, 2–18
starting
controller • *I/O User's II*, 2–9
protocol • *I/O User's II*, 2–15
tributary • *I/O User's II*, 2–15
status, DMF32 driver • *I/O User's II*, 2–14
status returns • *I/O User's II*, A–3
stopping
controller • *I/O User's II*, 2–18
modem line • *I/O User's II*, 2–18
protocol • *I/O User's II*, 2–18
tributary • *I/O User's II*, 2–18
supported devices • *I/O User's II*, 2–1
sync characters • *I/O User's II*, 2–12, 2–13
SYS$GETDVI • *I/O User's II*, 2–3
timeout • *I/O User's II*, 2–13
tributary • *I/O User's II*, 2–1
address • *I/O User's II*, 2–1, 2–18
mode • *I/O User's II*, 2–1
starting • *I/O User's II*, 2–15
station • *I/O User's II*, 2–1
stopping • *I/O User's II*, 2–18
tributary counter parameter IDs • *I/O User's II*, 2–22
unit and line status • *I/O User's II*, 2–5
unit characteristics • *I/O User's II*, 2–4
write function • *I/O User's II*, 2–8

DMZ32 device • *I/O User's I*, 8–1
DNA (default name address) argument • *RMS*, B–5
DNM (default name) argument • *RMS*, B–3
DNM (default name) field • *RMS*, 4–3
DNM (default name) keyword
specifying FAB$L_DNA and FAB$B_DNS fields from VAX MACRO • *RMS*, 5–9
DNS (default name size) argument • *RMS*, B–5
DO clause
example • *Debugger*, 3–14
exiting • *Debugger*, CD–94, CD–112
format • *Debugger*, CD–4
DO command • *Debugger*, 10–6, 10–7, CD–76
Documentation
module description • *Modular Procedures*, 2–19, A–6
procedure description • *Modular Procedures*, 2–20, A–6
Documentation format
See System routine documentation
DO display • *Debugger*, 7–16, C–1
.DOUBLE directive • *MACRO*, 6–20
Double-precision value
converting • *RTL Math*, MTH–62
converting an array of • *RTL Math*, MTH–63
Double-width characters
See also Screen management
See also Virtual display
specifying • *Programming Resources*, 7–20
DOWN command • *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–24
/DOWN qualifier • *Debugger*, CD–98, CD–110, CD–118
DPT$V_NOUNLOAD • *Device Support (A)*, 12–7
DPT$V_NO_IDB_DISPATCH • *Device Support (A)*, 17–25
DPT$V_SMPMOD • *Device Support (A)*, 12–13, E–3
DPT$V_SUBCNTRL • *Device Support (A)*, 15–15
DPT$V_SVP • *Device Support (B)*, 1–79, 2–21, 3–79, 3–80
DPT$W_DEFUNITS • *Device Support (A)*, 12–21
DPT$W_DELIVER • *Device Support (B)*, 4–21
DPT$W_UNLOAD • *Device Support (B)*, 4–10
DPT (driver prologue table) • *System Dump Analyzer*, SDA–99; *Device Support (A)*, 1–2, 3–6, 11–1, 13–7; *Device Support (B)*, 1–31 to 1–35, 1–74, 1–76
creating • *Device Support (A)*, 6–1 to 6–3; *Device Support (B)*, 2–21 to 2–26
initialization table • *Device Support (A)*, 6–2, 12–4; *Device Support (B)*, 1–33, 2–25 to 2–26

- DPT (driver prologue table) (Cont.)
  - linked into system DPT list • *Device Support (A),* 12–3, 12–7, 12–8
  - of third-party SCSI class driver • *Device Support (A),* 17–25
  - reinitialization table • *Device Support (A),* 6–3, 12–4, 12–8; *Device Support (B),* 2–25, 2–25 to 2–26
- DPTAB macro • *Device Support (A),* 6–1, 11–1, 12–1, 16–11; *Device Support (B),* 1–69, 2–21 to 2–23
  - controlling autoconfiguration with • *Device Support (A),* 12–21
  - example • *Device Support (B),* 2–23
  - used by MASSBUS drivers • *Device Support (A),* 15–15
- DPT base address • *System Dump Analyzer,* SDA–24
- DPT_STORE macro • *Device Support (A),* 3–6, 6–2 to 6–3, 11–9; *Device Support (B),* 2–24 to 2–26
  - example • *Device Support (B),* 2–23
- DR11–W driver • *Device Support (A),* D–1 to D–26
- DR11–W/DRV11–WA driver
  - attention AST • *I/O User's II,* 3–14
  - BDP (buffered data path) • *I/O User's II,* 3–11, 3–15
  - block mode • *I/O User's II,* 3–4, 3–11, 3–15
  - CSR (control and status register)
    - ATTN bit • *I/O User's II,* 3–6, 3–11
    - bit assignment • *I/O User's II,* 3–16
    - CYCLE bit • *I/O User's II,* 3–5, 3–11
    - ERROR bit • *I/O User's II,* 3–6
    - FNCT and STATUS bits • *I/O User's II,* 3–5, 3–7, 3–11, 3–14
    - function • *I/O User's II,* 3–5
  - data registers • *I/O User's II,* 3–6
  - data transfer mode • *I/O User's II,* 3–4
  - data transfers
    - read and write • *I/O User's II,* 3–5
    - through BDP • *I/O User's II,* 3–15
  - DDP (direct data path) • *I/O User's II,* 3–11, 3–15
  - device characteristics • *I/O User's II,* 3–8
  - driver • *I/O User's II,* 3–1
  - EIR (error information register) • *I/O User's II,* 3–6
    - bit assignment • *I/O User's II,* 3–16
  - enable attention AST • *I/O User's II,* 3–14
  - error reporting • *I/O User's II,* 3–6
  - function codes • *I/O User's II,* 3–9, A–3
  - function modifiers • *I/O User's II,* 3–7, 3–11 to 3–12, 3–14 to 3–15
  - hardware errors • *I/O User's II,* 3–7, 3–8
  - I/O functions • *I/O User's II,* 3–13
  - I/O status block • *I/O User's II,* 3–15
    - byte count • *I/O User's II,* 3–15
- DR11–W/DRV11–WA driver (Cont.)
  - IDR (input data register) • *I/O User's II,* 3–6, 3–11, 3–14
  - interrupts • *I/O User's II,* 3–4, 3–6, 3–7, 3–8, 3–11, 3–14
  - link mode • *I/O User's II,* 3–6, 3–7, 3–11
  - NPR transfers • *I/O User's II,* 3–7
  - ODR (output data register) • *I/O User's II,* 3–6, 3–11
  - programming example • *I/O User's II,* 3–16
  - read function • *I/O User's II,* 3–13
  - set characteristics function • *I/O User's II,* 3–13
  - set mode function • *I/O User's II,* 3–13
  - SS$_BADPARAM • *I/O User's II,* 3–11
  - status returns • *I/O User's II,* A–3
  - SYS$CANCEL • *I/O User's II,* 3–14, 3–15
  - SYS$GETDVI • *I/O User's II,* 3–8
  - transfer mode • *I/O User's II,* 3–4
  - word mode • *I/O User's II,* 3–4, 3–11
  - write function • *I/O User's II,* 3–13
- DR32 device interconnect
  - See DDI
- DR32 driver
  - action routines • *I/O User's II,* 4–23, 4–28, 4–30, 4–34, 4–39
  - AST routine • *I/O User's II,* 4–15, 4–20, 4–21, 4–26, 4–33
  - buffer block • *I/O User's II,* 4–5, 4–13, 4–15, 4–21, 4–22, 4–25, 4–36
  - byte count field • *I/O User's II,* 4–15
  - command and data chaining • *I/O User's II,* 4–2
  - command block • *I/O User's II,* 4–5, 4–21, 4–22, 4–36
  - command chaining • *I/O User's II,* 4–2, 4–14, 4–29
  - command control • *I/O User's II,* 4–14
  - command packets • *I/O User's II,* 4–2, 4–4 to 4–7, 4–25 to 4–28, 4–31, 4–33 to 4–40
  - command sequences
    - device-initiated • *I/O User's II,* 4–7
    - initiating • *I/O User's II,* 4–7
  - control (command) messages • *I/O User's II,* 4–3, 4–7, 4–11, 4–12, 4–18, 4–29, 4–38
  - control select field • *I/O User's II,* 4–13
  - data chaining • *I/O User's II,* 4–2, 4–14, 4–29
  - data rate • *I/O User's II,* 4–4, 4–20, 4–22, 4–27
  - data transfer command table • *I/O User's II,* 4–21
  - data transfers • *I/O User's II,* 4–2, 4–3, 4–5, 4–11, 4–13, 4–14 to 4–16, 4–20, 4–25, 4–26, 4–29, 4–38
  - DDI (DR32 device interconnect) • *I/O User's II,* 4–2
  - device
    - characteristics • *I/O User's II,* 4–3

DR32 driver
device (Cont.)
control code • *I/O User's II*, 4–10, 4–28
message • *I/O User's II*, 4–7, 4–9, 4–11, 4–14, 4–18, 4–25, 4–27, 4–29, 4–32
diagnostic tests • *I/O User's II*, 4–10 to 4–13, 4–29, 4–39
DR device definition • *I/O User's II*, 4–2
driver • *I/O User's II*, 4–1
DSL (DR32 status longword) • *I/O User's II*, 4–9, 4–16, 4–24, 4–39
error checking • *I/O User's II*, 4–39
event flags • *I/O User's II*, 4–15, 4–20, 4–22, 4–26, 4–28, 4–30, 4–32, 4–33, 4–40
far-end DR device • *I/O User's II*, 4–2, 4–3, 4–5, 4–7, 4–11, 4–13, 4–18, 4–27
far-end DR device transfers • *I/O User's II*, 4–3
FREEQ (free queue) • *I/O User's II*, 4–5, 4–13, 4–18, 4–24, 4–27, 4–36
function codes • *I/O User's II*, A–4
function modifier • *I/O User's II*, 4–20
GO bit • *I/O User's II*, 4–7, 4–22
high-level language interface • *I/O User's II*, 4–4, 4–23
support routines • *I/O User's II*, 4–23
synchronization • *I/O User's II*, 4–33
I/O function codes • *I/O User's II*, 4–20
I/O status block • *I/O User's II*, 4–23, 4–32, 4–34, 4–39
INPTQ (input queue) • *I/O User's II*, 4–5, 4–11, 4–13, 4–22, 4–24, 4–28, 4–30, 4–38
INSQTI instruction • *I/O User's II*, 4–5
interrupt
See also DR32 driver, action routines
See also DR32 driver, event flags
AST • *I/O User's II*, 4–3, 4–28, 4–30, 4–32, 4–33, 4–34, 4–40
command packet • *I/O User's II*, 4–13, 4–20, 4–21, 4–22, 4–26, 4–28, 4–33, 4–38
reasons • *I/O User's II*, 4–3
interrupt, reasons for • *I/O User's II*, 4–3
interrupt control argument (XF$FREESET) • *I/O User's II*, 4–28
interrupt control field • *I/O User's II*, 4–15, 4–26, 4–40
length of device message field • *I/O User's II*, 4–9
length of log area field • *I/O User's II*, 4–10
load microcode function (IO$_LOADMCODE) • *I/O User's II*, 4–20
log area field • *I/O User's II*, 4–19
log message • *I/O User's II*, 4–30, 4–32
microcode loader (XFLOADER) • *I/O User's II*, 4–19

DR32 driver (Cont.)
NOP command packet • *I/O User's II*, 4–40
prefetch command packets • *I/O User's II*, 4–38
programming
examples • *I/O User's II*, 4–40
hints • *I/O User's II*, 4–37
interface • *I/O User's II*, 4–4
queue
headers • *I/O User's II*, 4–5, 4–21
processing • *I/O User's II*, 4–5
retry • *I/O User's II*, 4–6, 4–39, 4–47
random access • *I/O User's II*, 4–3, 4–13
REMQHI instruction • *I/O User's II*, 4–5
residual DDI byte count field • *I/O User's II*, 4–16
residual memory byte count field • *I/O User's II*, 4–16
start data transfer function (IO$_STARTDATA) • *I/O User's II*, 4–4, 4–7, 4–20
status returns • *I/O User's II*, 4–32, A–4
DDI status • *I/O User's II*, 4–37
device-dependent • *I/O User's II*, 4–36
suppress length error field • *I/O User's II*, 4–14
symbolic definitions • *I/O User's II*, 4–24
SYS$GETDVI • *I/O User's II*, 4–3
termination queue (TERMQ) • *I/O User's II*, 4–3, 4–5, 4–13
TERMQ (termination queue) • *I/O User's II*, 4–15 to 4–16, 4–21, 4–24, 4–30, 4–31, 4–33, 4–40
VAX FORTRAN programming • *I/O User's II*, 4–23, 4–24
VAX MACRO programming • *I/O User's II*, 4–23
virtual address of buffer field • *I/O User's II*, 4–15
XF$CLEANUP • *I/O User's II*, 4–33
XF$FREESET • *I/O User's II*, 4–27
XF$GETPKT • *I/O User's II*, 4–31
XF$PKTBLD • *I/O User's II*, 4–28
XF$STARTDEV • *I/O User's II*, 4–26
XFSETUP • *I/O User's II*, 4–24
DR32 status longword
See DSL
Drag operation
determining where started • *VAXTPU*, 7–188
Drawing characters • *RTL Screen Management*, 2–11
Drawing lines • *RTL Screen Management*, 2–11
Driver
See Device driver
asynchronous DDCMP • *I/O User's II*, 5–1
card reader • *I/O User's I*, 2–1
disk • *I/O User's I*, 3–1
DMC11/DMR11 • *I/O User's II*, 1–1
DMP11/DMF32 • *I/O User's II*, 2–1
DR11–W/DRV11–WA • *I/O User's II*, 3–1

Driver (Cont.)
DR32 • *I/O User's II*, 4–1
Ethernet/802 • *I/O User's II*, 6–1
LAT port • *I/O User's I*, 8–1
line printer • *I/O User's I*, 5–1
LPA11-K device • *I/O User's I*, 4–1
magnetic tape • *I/O User's I*, 6–1
mailbox • *I/O User's I*, 7–1
pseudoterminal • *I/O User's I*, 9–1
SCSI • *I/O User's I*, 3–22
shadow set virtual unit • *I/O User's I*, 10–1
terminal • *I/O User's I*, 8–1
VAXstation 2000 and MicroVAX 2000 disk • *I/O User's I*, 3–21
Driver dispatch table
See DDT
Driver prologue table
See DPT
Driver unloading routine • *Device Support (A)*, 6–3, 11–4, 12–7 to 12–8, 16–21; *Device Support (B)*, 2–22, 2–26
address • *Device Support (A)*, 6–2; *Device Support (B)*, 1–34, 4–10
context • *Device Support (B)*, 4–10
exit method • *Device Support (B)*, 4–10
functions • *Device Support (B)*, 4–10
input • *Device Support (B)*, 4–10
register usage • *Device Support (B)*, 4–10
synchronization requirements • *Device Support (B)*, 4–10
DRV11-WA driver • *Device Support (A)*, D–1 to D–26
See also DR11-W/DRV11-WA driver
DSA (DIGITAL Storage Architecture)
See DSA disk
DSA32 device • *I/O User's I*, 8–1
DSA disk • *I/O User's I*, 3–1, 3–14, 3–19
See also Disk
bad block • *I/O User's I*, 3–19, 3–21
bad block replacement • *I/O User's I*, 3–20, 3–21
forced error • *I/O User's I*, 3–20
forced error flag • *I/O User's I*, 3–21
use with Verify Utility • *I/O User's I*, 3–19, 3–21
DSBINT macro • *Device Support (A)*, 3–9, 3–10, 8–5, 8–6, E–4, E–9, E–10; *Device Support (B)*, 2–27
replacing with spin lock synchronization macro • *Device Support (A)*, E–13
DSC$K_DTYPE_BPV • *Modular Procedures*, 3–12
See also User-action routine
DSC$K_DTYPE_ZEM • *Modular Procedures*, 3–11
See also User-action routine
DST (debug symbol table)
creating • *Debugger*, 5–4
shareable image • *Debugger*, 5–14
source line correlation • *Debugger*, 6–1
DTK$ANSWER_PHONE • *RTL DECtalk*, 1–5, DTK–3
DTK$CHECK_HDWR_STATUS • *RTL DECtalk*, DTK–5
DTK$DIAL_PHONE • *RTL DECtalk*, 1–5, DTK–7
DTK$HANGUP_PHONE • *RTL DECtalk*, 1–5, DTK–9
DTK$INITIALIZE • *RTL DECtalk*, 1–1, DTK–10
DTK$LOAD_DICTIONARY • *RTL DECtalk*, 1–4, DTK–12
DTK$READ_KEYSTROKE • *RTL DECtalk*, 1–5, DTK–14
DTK$READ_STRING • *RTL DECtalk*, 1–5, DTK–16
DTK$RETURN_LAST_INDEX • *RTL DECtalk*, 1–4, DTK–18
DTK$SET_INDEX • *RTL DECtalk*, 1–4, DTK–19
DTK$SET_KEYPAD_MODE • *RTL DECtalk*, 1–5, DTK–20
DTK$SET_LOGGING_MODE • *RTL DECtalk*, 1–2 to 1–3, DTK–22
DTK$SET_MODE • *RTL DECtalk*, DTK–25
DTK$SET_SPEECH_MODE • *RTL DECtalk*, DTK–27
DTK$SET_TERMINAL_MODE • *RTL DECtalk*, 1–3, DTK–29
DTK$SET_VOICE • *RTL DECtalk*, DTK–31
DTK$SPEAK_FILE • *RTL DECtalk*, DTK–33
DTK$SPEAK_PHONEMIC_TEXT • *RTL DECtalk*, DTK–35
DTK$SPEAK_TEXT • *RTL DECtalk*, 1–4, DTK–37
DTK$SPELL_TEXT • *RTL DECtalk*, DTK–39
DTK$TERMINATE • *RTL DECtalk*, 1–4, DTK–41
$DTKDEF library • *RTL DECtalk*, 1–5
Dual host
definition of • *I/O User's I*, 3–4
Dual path
definition of • *I/O User's I*, 3–11
Dual-pathed disk • *I/O User's I*, 3–11
DSA disk • *I/O User's I*, 3–14
Dual path UCB extension • *Device Support (B)*, 1–69
Dual-ported device • *Device Support (B)*, 1–74
Dual-ported disk • *I/O User's I*, 3–12
DSA disk • *I/O User's I*, 3–14
HSC disk • *I/O User's I*, 3–15
restrictions for use • *I/O User's I*, 3–13
Dump
hexadecimal • *Analyze/RMS_File*, ARMS–25
DUMP
subset • *System Dump Analyzer*, SDA–4
DUMPBUG parameter • *System Dump Analyzer*, SDA–2, SDA–28

DUMP command • *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–25
Dump file
  See also SDA
  analyzing • *Programming Resources*, 1–21; *System Dump Analyzer*, SDA–32
  copying the contents • *System Dump Analyzer*, SDA–42
DUMPSTYLE parameter • *System Dump Analyzer*, SDA–4
DUP (duplicate) option
  in XAB$B_FLG field • *RMS*, B–21
Duplex mode
  See also Half-duplex mode
  terminal • *I/O User's I*, 8–10
Duplicate key • *File Def Language*, FDL–27
  examples • *RMS*, 7–8
  incompatibility between VMS RMS and RMS–11 • *RMS*, 13–9
  insertion order • *RMS*, RMS–72
  null key processing • *File Applications*, 3–19
  retrieving records • *RMS*, 7–8
Duplicate key values • *File Def Language*, FDL–5
DUPLICATES attribute • *File Def Language*, FDL–27
DUPLICATES_PER_SIDR attribute • *File Def Language*, FDL–5
DWBUA (VAXBI-to-UNIBUS adapter) • *Device Support (A)*, 1–13, 16–10, 19–4
  See also UNIBUS adapter
DWMBA (XMI-to-VAXBI adapter)
  See Memory interconnect to VAXBI adapter
DWMUA (VAXBI-to-UNIBUS adapter) • *Device Support (A)*, 1–13, 16–10
  See also UNIBUS adapter
DYN$C_BUFIO • *Device Support (B)*, 3–12, 3–22
DYN$C_IRP • *Device Support (B)*, 3–12
DYNAMIC attribute • *System Services Intro*, 3–5
Dynamic length string • *RTL String Manipulation*, 2–1, 2–2, 2–3, STR–68
  allocation of • *RTL String Manipulation*, STR–46
  deallocation of • *RTL String Manipulation*, STR–45
Dynamic memory allocation • *RTL Library*, 5–1
Dynamic mode • *Debugger*, CD–152
  image setting • *Debugger*, 5–14
  module setting • *Debugger*, 5–7
    with DECwindows • *Debugger*, 1–27
Dynamic process setting • *Debugger*, 10–8, CD–162
Dynamic prompt setting • *Debugger*, 10–2, CD–165
/DYNAMIC qualifier • *Debugger*, CD–71, CD–162, CD–236
Dynamic selection
  in EVE editor • *VAXTPU*, 4–16 to 4–17
Dynamic spin lock • *Device Support (A)*, 3–13
Dynamic string • *RTL General Purpose*, OTS–95
Dynamic string descriptor • *Routines Intro*, 2–24
DZ11 controller • *Device Support (B)*, 1–21
DZ11 device • *I/O User's I*, 8–1
DZ32 controller • *Device Support (B)*, 1–21
DZ32 device • *I/O User's I*, 8–1
DZV11 device • *I/O User's I*, 8–1
D_floating data type • *MACRO*, 8–4, 9–102
.D_FLOATING directive • *MACRO*, 6–20
/D_FLOAT qualifier • *Debugger*, CD–62, CD–86

E

ECC error correction • *Device Support (B)*, 1–78, 1–79, 1–83, 2–21, 3–67
ECC position register • *Device Support (B)*, 1–83
Echo
  terminal • *Programming Resources*, 7–40
  terminator • *Programming Resources*, 7–24
/ECHO qualifier • *Debugger*, CD–51; *System Dump Analyzer*, SDA–44
ECO level • *Patch*, PAT–2
  See also PATCH commands
  checking • *Patch*, PAT–45, PAT–46, PAT–47
  setting • *Patch*, PAT–33, PAT–35, PAT–75
;E command • *Delta/XDelta*, DELTA–38
ECRB (Ethernet controller data block) • *Device Support (B)*, 2–2
EDF$MAKE_FDL logical name • *File Applications*, 4–14
Edit
  instruction • *MACRO*, 9–169
    vector • *MACRO*, 10–83
  pattern operator • *MACRO*, 9–170, 9–172
EDIT/ACL command • *File Applications*, 4–22
EDIT built-in procedure • *VAXTPU*, 7–111 to 7–114
EDIT command • *Debugger*, CD–78
EDIT/FDL
  See Edit/FDL Utility
EDIT/FDL command • *Programming Resources*, 8–55
Edit/FDL Utility (EDIT/FDL) • *Programming Resources*, 1–39; *File Applications*, 1–14; *File Def Language*, FDL–39, FDL–40, FDL–42
  ANALYSIS_OF_KEY section • *File Def Language*, FDL–4
  calculating bucket size • *File Applications*, 3–13, 3–25

Edit/FDL Utility (EDIT/FDL) (Cont.)
- calculating extension size • *File Applications,* 3–5, 9–8
- commands • *File Applications,* 4–3; *File Def Language,* FDL–58
- contiguous files • *File Applications,* 3–4
- creating areas for index structures • *File Applications,* 3–23
- creating FDL files • *File Applications,* 4–2, 4–5; *File Def Language,* FDL–39
- default value • *File Applications,* 4–11
- editor • *Programming Resources,* 8–55
- exiting • *File Def Language,* FDL–43
- invoking • *File Def Language,* FDL–43
- invoking a script • *File Applications,* 4–5
- modifying a data file • *Programming Resources,* 8–58
- optimization algorithms • *File Applications,* A–1
- Optimize script • *File Applications,* 10–1, 10–25; *File Def Language,* FDL–39
- prompt • *File Applications,* 4–11
- restrictions • *File Def Language,* FDL–43
- scripts • *File Def Language,* FDL–63
- specifying run-time options • *File Applications,* 9–1 to 9–5

Editing commands
- adding lines • *SUMSLP,* SUM–7, SUM–9
- changing audit trail text • *SUMSLP,* SUM–12
- deleting lines • *SUMSLP,* SUM–9, SUM–10, SUM–11
- format of • *SUMSLP,* SUM–4
- replacing lines • *SUMSLP,* SUM–11
- specifying • *SUMSLP,* SUM–3
- using command parameters • *SUMSLP,* SUM–4
- using locator field parameters • *SUMSLP,* SUM–4
- using operators • *SUMSLP,* SUM–3

Editing context status
- built-in procedures
  - CURRENT_BUFFER • *VAXTPU,* 7–80
  - CURRENT_CHARACTER • *VAXTPU,* 7–81
  - CURRENT_COLUMN • *VAXTPU,* 7–83
  - CURRENT_DIRECTION • *VAXTPU,* 7–85
  - CURRENT_LINE • *VAXTPU,* 7–86
  - CURRENT_OFFSET • *VAXTPU,* 7–88
  - CURRENT_ROW • *VAXTPU,* 7–90
  - CURRENT_WINDOW • *VAXTPU,* 7–92
  - DEBUG_LINE • *VAXTPU,* 7–99
  - ERROR • *VAXTPU,* 7–123
  - ERROR_LINE • *VAXTPU,* 7–125
  - ERROR_TEXT • *VAXTPU,* 7–127
- built-in procedures for defining
  - SET • *VAXTPU,* 7–347

Editing context status
- built-in procedures for defining (Cont.)
  - SHOW • *VAXTPU,* 7–505

Editing interface
- See EVE editor

Editing point
- built-in procedures for moving
  - MARK • *VAXTPU,* 7–261
  - MOVE_HORIZONTAL • *VAXTPU,* 7–278
  - MOVE_VERTICAL • *VAXTPU,* 7–282
  - POSITION • *VAXTPU,* 7–287
- compared to cursor position • *VAXTPU,* 6–10
- effect of scrolling on • *VAXTPU,* 7–324

Editor
- See also Text processing
- EDT • *Programming Resources,* 1–3
- EVE • *Programming Resources,* 1–5
- FDL • *File Def Language,* FDL–42
- SUMSLP • *SUMSLP,* SUM–14
- text • *File Def Language,* FDL–42
- VAX Text Processing Utility • *Programming Resources,* 1–4

EDITPC (Edit Packed to Character String) instruction • *MACRO,* 9–170

/EDIT qualifier • *Debugger,* CD–28, CD–176, CD–246

EDIT/SUM command • *SUMSLP,* SUM–2, SUM–14

EDIT/TPU command • *VAXTPU,* 1–9, 5–1 to 5–20
- parameter • *VAXTPU,* 5–19
- qualifiers • *VAXTPU,* 1–9 to 1–10, 5–5 to 5–20
  - /COMMAND • *VAXTPU,* 5–6 to 5–7
  - /CREATE • *VAXTPU,* 5–7
  - /DEBUG • *VAXTPU,* 4–33, 5–8
  - /DISPLAY • *VAXTPU,* 5–8
  - /INITIALIZATION • *VAXTPU,* 5–9 to 5–10
  - /INTERFACE • *VAXTPU,* 5–10
  - /JOURNAL • *VAXTPU,* 5–10
  - /MODIFY • *VAXTPU,* 5–12
  - /OUTPUT • *VAXTPU,* 5–12
  - /READ_ONLY • *VAXTPU,* 5–13
  - /RECOVER • *VAXTPU,* 5–14, 7–408
  - /SECTION • *VAXTPU,* 5–16
  - /START_POSITION • *VAXTPU,* 5–17
  - /WRITE • *VAXTPU,* 5–17

“Edit_mode” string constant parameter to GET_INFO • *VAXTPU,* 7–198

EDIV (Extended Divide) instruction • *MACRO,* 9–19
- RTL routine to access • *RTL Library,* LIB–126

EDT$EDIT routine • *Utility Routines,* EDT–3

EDT argument • *RMS,* B–16

EDT editor
- mode
  - keypad • *Programming Resources,* 1–3

EDT editor
mode (Cont.)
line • *Programming Resources*, 1–3
nokeypad • *Programming Resources*, 1–4
EDT routines
examples • *Utility Routines*, EDT–1 to EDT–2
introduction • *Utility Routines*, EDT–1
user-written
FILEIO • *Utility Routines*, EDT–7
WORKIO • *Utility Routines*, EDT–11
XLATE • *Utility Routines*, EDT–13
EDT text editor
See EDT editor
ef_cluster_name data type • *Routines Intro*, A–5t
ef_number data type • *Routines Intro*, A–5t
EH? error message • *Delta/XDelta*, DELTA–13
“Eightbit” string constant parameter to GET_INFO • *VAXTPU*, 7–198
EIR (error information register) • *I/O User's II*, 3–6
bit assignment
• *I/O User's II*, 3–16
Elapsed time • *Convert*, CONV–24
Element
definition of • *RTL Parallel Processing*, 1–2
retrieving information about • *RTL Parallel Processing*, 4–1
synchronization • *RTL Parallel Processing*, 4–1
Element identifier
sharing • *RTL Parallel Processing*, 5–9
ELSE clause • *VAXTPU*, 3–22
%ELSE lexical keyword • *VAXTPU*, 3–36
EMB$C_DA • *Device Support (A)*, 11–10
EMB$C_DE • *Device Support (A)*, 11–10
EMB$C_DT • *Device Support (A)*, 11–10
EMB$L_DV_REGSAV • *Device Support (A)*, 11–9
EMB$W_DV_STS • *Device Support (B)*, 3–94
$EMBDEF macro • *Device Support (A)*, 11–9
EMB spin lock • *Device Support (A)*, 3–14; *Device Support (B)*, 3–8
EMODD (Extended Multiply and Integerize D_ floating) instruction • *MACRO*, 9–115
RTL routine to access • *RTL Library*, LIB–128
EMODF (Extended Multiply and Integerize F_floating) instruction • *MACRO*, 9–115
RTL routine to access • *RTL Library*, LIB–130
EMODG (Extended Multiply and Integerize G_ floating) instruction • *MACRO*, 9–115
RTL routine to access • *RTL Library*, LIB–132
EMODH (Extended Multiply and Integerize H_ floating) instruction • *MACRO*, 9–115
RTL routine to access • *RTL Library*, LIB–134
/EMPHASIS qualifier • *File Def Language*, FDL–42, FDL–50
EMUL (Extended Multiply) instruction • *MACRO*, 9–20
RTL routine to access • *RTL Library*, LIB–136
Emulated instructions
in device driver • *Device Support (A)*, 5–3
Enable assembler functions • *MACRO*, 6–22
ENABLE AST command • *Debugger*, 9–17, CD–80
Enable attention AST function
asynchronous DDCMP driver • *I/O User's II*, 5–9
DMC11/DMR11 driver • *I/O User's II*, 1–7
DMP11/DMF32 driver • *I/O User's II*, 2–19
DR11–W/DRV11–WA driver • *I/O User's II*, 3–14
Ethernet/802 drivers • *I/O User's II*, 6–36
.ENABLE directive • *MACRO*, 6–22, 6–34
ENBINT macro • *Device Support (A)*, 3–9, 3–10, E–4; *Device Support (B)*, 2–28
replacing with spin lock synchronization macro • *Device Support (A)*, E–13
Encryption key • *Device Support (B)*, 1–42
.ENDC directive • *MACRO*, 6–26
End conditional assembly directive (.END) • *MACRO*, 6–26
.END directive • *Programming Resources*, 9–8; *MACRO*, 6–25
in message source file • *Message*, MSG–17
%ENDIF lexical keyword • *VAXTPU*, 3–36
ENDIF statement • *VAXTPU*, 3–22 to 3–23
ENDLOOP statement • *VAXTPU*, 3–21 to 3–22
End macro definition directive (.ENDM) • *MACRO*, 6–27
.ENDM directive • *MACRO*, 6–27
ENDMODULE statement • *VAXTPU*, 3–14 to 3–15
End-of-file
See EOF
End-of-file field in XABFHC
See XAB$L_EBK field
End-of-file mark
positioning for user file open option • *RMS*, 5–18
End-of-file option
See RAB$V_EOF option
End-of-file positioning • *RMS*, RMS–7
End-of-tape
See EOT
End-of-volume
detection on magnetic tape • *I/O User's I*, 6–20
ENDON_ERROR statement • *VAXTPU*, 3–25 to 3–31
ENDPROCEDURE statement • *VAXTPU*, 3–15 to 3–21
.ENDR directive • *MACRO*, 6–28

END_OF built-in procedure • *VAXTPU,* 7–115 to 7–116
END_OF_FILE attribute • *File Def Language,* FDL–10
Engineering change order (ECO) level
  See ECO level
Entering control characters • *VAXTPU,* 3–2
Enter service • *RMS,* RMS–29
  condition values • *RMS,* RMS–31
  control block input fields • *RMS,* RMS–30
  control block output fields • *RMS,* RMS–30
  requirement for NAM block fields • *RMS,* RMS–30
Entry and display modes • *Patch,* PAT–14
  ASCII-NOASCII mode • *Patch,* PAT–16
  BYTE mode • *Patch,* PAT–16
  canceling • *Patch,* PAT–40
  DECIMAL mode • *Patch,* PAT–17
  displaying location contents • *Patch,* PAT–62
  displaying mode • *Patch,* PAT–85
  GLOBALS-NOGLOBALS mode • *Patch,* PAT–17
  HEXADECIMAL mode • *Patch,* PAT–17
  INSTRUCTION-NOINSTRUCTION mode • *Patch,* PAT–15
  length modes • *Patch,* PAT–16
  LONG mode • *Patch,* PAT–16
  mode qualifier, PATCH command • *Patch,* PAT–15
  OCTAL mode • *Patch,* PAT–17
  radix modes • *Patch,* PAT–17
  SCOPE-NOSCOPE mode • *Patch,* PAT–17
  setting the mode • *Patch,* PAT–76
  symbol search mode • *Patch,* PAT–17
  SYMBOLS-NOSYMBOLS mode • *Patch,* PAT–16
  WORD mode • *Patch,* PAT–16
.ENTRY directive • *MACRO,* 6–29
Entry mask • *MACRO,* 9–63
Entry mask procedure • *Routines Intro,* A–11t
Entry point • *RTL Intro,* 3–4
  See also JSB entry points
  CALL entry point • *RTL Intro,* 3–3; *RTL String Manipulation,* 2–9
  defining • *MACRO,* 6–29
  JSB entry point • *RTL Intro,* 3–5; *RTL String Manipulation,* 2–9
  specifying in driver tables • *Device Support (B),* 2–13
Entry point directive (.ENTRY) • *MACRO,* 6–29
Entry point name • *RTL Math,* 1–1
EO$ADJUST_INPUT (Adjust Input Length) pattern operator • *MACRO,* 9–175
EO$BLANK_ZERO (Blank Backwards when Zero) pattern operator • *MACRO,* 9–176
EO$CLEAR_SIGNIF (Clear Significance) pattern operator • *MACRO,* 9–185
EO$END (End Edit) pattern operator • *MACRO,* 9–177
EO$END_FLOAT (End Floating Sign) pattern operator • *MACRO,* 9–178
EO$FILL (Store Fill) pattern operator • *MACRO,* 9–179
EO$FLOAT (Float Sign) pattern operator • *MACRO,* 9–180
EO$INSERT (Insert Character) pattern operator • *MACRO,* 9–181
EO$LOAD_FILL (Load Fill Register) pattern operator • *MACRO,* 9–182
EO$LOAD_MINUS (Load Sign Register If Minus) pattern operator • *MACRO,* 9–182
EO$LOAD_PLUS (Load Sign Register If Plus) pattern operator • *MACRO,* 9–182
EO$LOAD_SIGN (Load Sign Register) pattern operator • *MACRO,* 9–182
EO$MOVE (Move Digits) pattern operator • *MACRO,* 9–183
EO$REPLACE_SIGN (Replace Sign when Zero) pattern operator • *MACRO,* 9–184
EO$SET_SIGNIF (Set Significance) pattern operator • *MACRO,* 9–185
EO$STORE_SIGN (Store Sign) pattern operator • *MACRO,* 9–186
EOB_TEXT keyword • *VAXTPU,* 7–374
"Eob_text" string constant parameter to GET_INFO • *VAXTPU,* 7–171
EOF (end of file) • *Programming Resources,* 7–5
  status
    card reader • *I/O User's I,* 2–2
    magnetic tape • *I/O User's I,* 6–17
  write mailbox message • *I/O User's I,* 7–9
EOF (end-of-file) option • *File Def Language,* FDL–10
EOJ command
  in card reader batch job • *I/O User's I,* 2–2
EOT (end-of-tape)
  status
    magnetic tape • *I/O User's I,* 6–17, 6–19, 6–21
EQUAL keyword
  with GSMATCH option • *Programming Resources,* 5–5
Equal-or-next key option • *File Applications,* 8–9
Equivalence name
  defining • *System Services Intro,* 6–1
  format convention • *System Services Intro,* 6–10
  specifying • *System Services,* SYS–68
EQUIVALENCE statement • *VAXTPU,* 3–33 to 3–34
Equivalence string • *File Applications,* 6–4

$EQULST macro • *Device Support (B),* 2–29 to 2–30
example • *Device Support (B),* 2–30, 2–103
ERASE built-in procedure • *VAXTPU,* 7–117 to 7–118
Erase service • *File Applications,* 5–9; *RMS,* RMS–32
alternative • *RMS,* RMS–33
condition values • *RMS,* RMS–34
See also Completion status code
control block input fields • *RMS,* RMS–33
control block output fields • *RMS,* RMS–33
requirements for using • *RMS,* RMS–33
use restriction • *RMS,* RMS–33
ERASE_CHARACTER built-in procedure • *VAXTPU,* 7–119 to 7–120
ERASE_LINE built-in procedure • *VAXTPU,* 7–121 to 7–122
ERASE_UNMODIFIABLE
keyword parameter to SET built-in procedure • *VAXTPU,* 7–375
ERASE_UNMODIFIABLE mode
and APPEND_LINE • *VAXTPU,* 7–376
and CHANGE_CASE • *VAXTPU,* 7–376
and COPY_TEXT • *VAXTPU,* 7–376
and EDIT • *VAXTPU,* 7–376
and ERASE (buffer) • *VAXTPU,* 7–376
and ERASE (range) • *VAXTPU,* 7–376
and ERASE_CHARACTER • *VAXTPU,* 7–376
and ERASE_LINE • *VAXTPU,* 7–376
and FILL • *VAXTPU,* 7–376
and MOVE_TEXT • *VAXTPU,* 7–376
and SPLIT_LINE • *VAXTPU,* 7–376
and TRANSLATE • *VAXTPU,* 7–377
“erase_unmodifiable” string constant parameter
GET_INFO built-in • *VAXTPU,* 7–169
“Erase_unmodifiable” string constant parameter to GET_INFO • *VAXTPU,* 7–171
Erasing unmodifiable records • *VAXTPU,* 7–375
Erasure operations • *RTL Screen Management,* 2–7
ERL$DEVICEATTN • *Device Support (A),* 11–10; *Device Support (B),* 3–8 to 3–9, 4–15
ERL$DEVICERR • *Device Support (A),* 11–10; *Device Support (B),* 1–30, 1–80, 1–81, 3–8 to 3–9, 4–15
ERL$DEVICTMO • *Device Support (A),* 10–6, 11–10; *Device Support (B),* 1–30, 1–80, 1–81, 3–8 to 3–9, 4–15
ERL$RELEASEMB • *Device Support (A),* 10–3; *Device Support (B),* 3–95
Error • *RTL Intro,* 3–14
See also Error logging
associated with I/O request • *Device Support (A),* 11–10
Error (Cont.)
in file structure • *Analyze/RMS_File,* ARMS–13
not associated with I/O request • *Device Support (A),* 11–10
recommended method for signaling • *RMS,* 2–6
resulting from exceeding virtual address space • *VAXTPU,* 5–1
returning condition value • *RTL Intro,* 3–15
servicing within driver • *Device Support (A),* 1–4, 8–5; *Device Support (B),* 3–82 to 3–83
signaling condition value • *RTL Intro,* 3–15
signaling of • *RTL Library,* 4–3
Error check • *System Services Intro,* 2–16; *File Applications,* 10–1
in FOLR routines • *RTL Math,* 2–7
Error completion routine • *RMS,* 2–5
Error condition • *Analyze/RMS_File,* ARMS–7
Error creating shared memory
reasons for • *RTL Parallel Processing,* 3–2
.ERROR directive • *MACRO,* 6–31
Error handler
case-style • *VAXTPU,* 3–28 to 3–31
procedural • *VAXTPU,* 3–26 to 3–28
Error handling • *Programming Resources,* 9–1; *VAXTPU,* 3–25 to 3–31, 4–38
See also Condition handling
Error information register
See EIR
ERROR lexical element • *VAXTPU,* 3–25
ERRORLOG.EXE • *System Dump Analyzer,* SDA–60
Error log allocation buffer • *Device Support (A),* 11–10; *Device Support (B),* 3–8
ERRORLOGBUFFERS parameter • *System Dump Analyzer,* SDA–3
Error log entry
examining the contents of • *Device Support (A),* 17–33 to 17–43
Error logger
sending message to • *System Services,* SYS–441
Error logging • *Device Support (B),* 1–79 to 1–80, 3–8 to 3–9
driver prerequisites • *Device Support (A),* 11–9
enabling • *Device Support (B),* 1–75
error log sequence number • *Device Support (B),* 1–42
final error count • *Device Support (A),* 10–3
inhibiting • *Device Support (B),* 3–8
in progress • *Device Support (B),* 1–77
performed by IOC$REQCOM • *Device Support (B),* 3–95
Error logging enable bit
See UCB$V_ERLOGIP

Error logging routine • *Device Support (A)*, 1–4, 11–9 to 11–10; *Device Support (B)*, 1–30
  See also Register dumping routine
  address • *Device Support (A)*, 11–1
  in SCSI third-party class driver • *Device Support (A)*, 17–20 to 17–22
Error logging routines
  global symbols • *System Dump Analyzer*, SDA–60
Error log in progress bit
  See UCB$V_ERLOGIP
Error log UCB extension • *Device Support (B)*, 1–69, 1–80 to 1–81
Error message
  warning • *Convert*, CONV–3
Error message buffer • *Device Support (A)*, 3–14, 10–3; *Device Support (B)*, 1–81, 1–83, 3–82
  allocating • *Device Support (A)*, 11–10; *Device Support (B)*, 3–8
  filling • *Device Support (B)*, 3–9
  initializing • *Device Support (A)*, 11–10
  of third-party SCSI device driver • *Device Support (A)*, 17–20 to 17–21
  releasing • *Device Support (A)*, 10–3; *Device Support (B)*, 3–95
  size • *Device Support (B)*, 3–8
  specifying size • *Device Support (A)*, 6–4, 11–9, 11–10; *Device Support (B)*, 1–30
  written into by IOC$REQCOM • *Device Support (B)*, 3–95
Error PPL$_INSVIRMEM
  reasons for • *RTL Parallel Processing*, PPL–11
/ERROR qualifier • *Debugger*, 7–20, CD–123
  in message definition • *Message*, MSG–23
Error recovery • *System Services Intro*, 7–13
  disk • *I/O User's I*, 3–17
  line printer • *I/O User's I*, 5–3
  magnetic tape • *I/O User's I*, 6–9
  shadow set virtual unit driver • *I/O User's I*, 10–9
ERROR statement • *VAXTPU*, 7–123 to 7–124
Error status
  clearing • *Device Support (A)*, 11–2
Error status code • *RMS*, 2–6
  from invalid control blocks • *RMS*, 2–6
ERROR_LINE lexical element • *VAXTPU*, 3–26
ERROR_LINE statement • *VAXTPU*, 7–125 to 7–126
ERROR_TEXT lexical element • *VAXTPU*, 3–26
ERROR_TEXT statement • *VAXTPU*, 7–127 to 7–128
ESA (expanded string area address)
  program example • *RMS*, 4–12
Escape sequence
  ANSI • *I/O User's I*, B–9
Escape sequence (Cont.)
  Digital-private • *I/O User's I*, B–9
  terminal • *I/O User's I*, 8–7, 8–21
Escape sequences
  read • *Programming Resources*, 7–53
  using from terminal devices • *RMS*, RMS–49
ESC command • *Delta/XDelta*, DELTA–23
ESC key equivalent • *Delta/XDelta*, DELTA–23
ESP symbol • *System Dump Analyzer*, SDA–13
Ethernet
  device drivers • *I/O User's II*, 6–1
Ethernet/802 drivers
  address
    destination • *I/O User's II*, 6–17, 6–20
    Ethernet • *I/O User's II*, 6–2 to 6–5
    hardware • *I/O User's II*, 6–38
    loopback assistance • *I/O User's II*, 6–4
    multicast • *I/O User's II*, 6–4, 6–17, 6–29, 6–30
    node • *I/O User's II*, 6–2
    physical • *I/O User's II*, 6–2, 6–4, 6–17, 6–31, 6–38
    port • *I/O User's II*, 6–31
    shared protocol destination • *I/O User's II*, 6–26
    source • *I/O User's II*, 6–17
  AST access mode • *I/O User's II*, 6–36
  AST service routine address • *I/O User's II*, 6–36
  attention AST • *I/O User's II*, 6–36
  buffer
    hardware • *I/O User's II*, 6–23
    receive • *I/O User's II*, 6–17, 6–23
  channel assignment • *I/O User's II*, 6–2
  characteristics
    device • *I/O User's II*, 6–14, 6–37
    extended • *I/O User's II*, 6–23 to 6–34, 6–38
  controller mode • *I/O User's II*, 6–24
  CRC generation • *I/O User's II*, 6–25
  data chaining • *I/O User's II*, 6–26
  device characteristics • *I/O User's II*, 6–14, 6–37
    See also Ethernet/802 drivers, extended characteristics
  drivers • *I/O User's II*, 6–1
    initializing • *I/O User's II*, 6–2
    operating • *I/O User's II*, 6–2
  driver service (802 format) • *I/O User's II*, 6–34
  echo mode (DEUNA only) • *I/O User's II*, 6–27
  error summary bits • *I/O User's II*, 6–15
  Ethernet • *I/O User's II*, 6–1, 6–2, 6–7
  Ethernet addresses • *I/O User's II*, 6–2
  Ethernet packet format • *I/O User's II*, 6–6
  Ethernet packet padding • *I/O User's II*, 6–8

- Ethernet/802 drivers (Cont.)
  - Ethernet programming example • *I/O User's II*, 6–41
  - exclusive mode • *I/O User's II*, 6–9
  - extended characteristics • *I/O User's II*, 6–23 to 6–34, 6–37
  - function codes • *I/O User's II*, 6–16, A–6
  - function modifiers • *I/O User's II*, 6–19, 6–21, 6–22, 6–36 to 6–37
  - hardware buffer size • *I/O User's II*, 6–23
  - hardware interface • *I/O User's II*, 6–2
  - I/O functions • *I/O User's II*, 6–17, 6–19, 6–21, 6–37
  - I/O status block • *I/O User's II*, 6–39
  - IEEE 802
    - Class I service packet format • *I/O User's II.* 6–10, 6–27
    - driver service parameter • *I/O User's II*, 6–34
    - extended packet format • *I/O User's II*, 6–13, 6–27
    - 802 format SAP parameter • *I/O User's II*, 6–33
    - group SAP parameter • *I/O User's II*, 6–28
    - programming example • *I/O User's II*, 6–47
    - read function • *I/O User's II*, 6–17
    - SAP use and restrictions • *I/O User's II*, 6–12
    - support • *I/O User's II*, 6–5
    - user-supplied service packet format • *I/O User's II*, 6–11, 6–27
    - write function • *I/O User's II*, 6–19
  - internal loopback mode (DELUA only) • *I/O User's II*, 6–29
  - loopback mode • *I/O User's II*, 6–24
  - message size • *I/O User's II*, 6–15, 6–17, 6–19, 6–20, 6–24
  - modify characteristics • *I/O User's II*, 6–22
  - multicast address state • *I/O User's II*, 6–30
  - packet format • *I/O User's II*, 6–6
    - Ethernet • *I/O User's II*, 6–6
    - extended 802 • *I/O User's II*, 6–13
    - IEEE 802 • *I/O User's II*, 6–10
    - set mode parameters • *I/O User's II*, 6–34
    - SNAP SAP value • *I/O User's II*, 6–14
    - user-supplied service • *I/O User's II*, 6–11
  - padding
    - message size • *I/O User's II*, 6–15, 6–19
    - transmit messages • *I/O User's II*, 6–30
  - parameter ID • *I/O User's II*, 6–22
    - packet format • *I/O User's II*, 6–34
  - parameter validation • *I/O User's II*, 6–35
  - port • *I/O User's II*, 6–1
    - address • *I/O User's II*, 6–23

- Ethernet/802 drivers
  - port (Cont.)
    - start • *I/O User's II*, 6–22
  - privilege • *I/O User's II*, 6–17
  - programming example • *I/O User's II*, 6–41, 6–47
  - programming notes • *I/O User's II*, 6–40
  - promiscuous mode • *I/O User's II*, 6–32, 6–40
    - rules for • *I/O User's II*, 6–41
  - protocol type • *I/O User's II*, 6–1, 6–17, 6–20, 6–32
    - access mode • *I/O User's II*, 6–23
    - cross-company • *I/O User's II*, 6–7
    - DIGITAL • *I/O User's II*, 6–7
    - Ethernet • *I/O User's II*, 6–7
    - sharing • *I/O User's II*, 6–9
  - protocol type sharing • *I/O User's II*, 6–9
  - read function • *I/O User's II*, 6–17
  - restart • *I/O User's II*, 6–33
  - sense mode function • *I/O User's II*, 6–37
  - Service Access Point (SAP) • *I/O User's II*, 6–12
  - set controller mode • *I/O User's II*, 6–22
    - extended characteristics • *I/O User's II*, 6–23 to 6–34
    - P2 buffer • *I/O User's II*, 6–22
    - parameter ID • *I/O User's II*, 6–22
    - protocol type sharing • *I/O User's II*, 6–9
  - set mode function • *I/O User's II*, 6–21
  - shared default mode • *I/O User's II*, 6–9
  - shared with destination mode • *I/O User's II*, 6–9
  - shutdown controller mode • *I/O User's II*, 6–36
  - shutdown port • *I/O User's II*, 6–36
  - software interface • *I/O User's II*, 6–2
  - status returns • *I/O User's II*, A–6
  - supported devices • *I/O User's II*, 6–1
  - SYS$ASSIGN • *I/O User's II*, 6–2
  - SYS$DASSGN • *I/O User's II*, 6–2
  - SYS$GETDVI • *I/O User's II*, 6–14
  - transmit/receive buffer size • *I/O User's II*, 6–23
  - unit and line status • *I/O User's II*, 6–15
  - write function • *I/O User's II*, 6–19
- ETO (extended terminal operation) option • *RMS*, RMS–49
  - See also RAB$V_ETO option
- ETYPE • *MACRO*, 10–6, 10–69
- Euclidean norm
  - of a vector • *RTL Math*, MTH–170
- Evaluate
  - memory address • *Debugger*, 4–13, CD–83
    - with DECwindows • *Debugger*, 1–26
- EVALUATE/ADDRESS command • *Debugger*, 3–13, 3–19, 4–13, CD–83

- EVALUATE command • *Debugger*, 4–5, CD–81; *Patch*, PAT–59 to PAT–61; *System Dump Analyzer*, SDA–48
- EVALUATE/PSL command • *System Dump Analyzer*, SDA–22
- Evaluation precedence • *Delta/XDelta*, DELTA–9
- EVE editor
  - building applications on • *VAXTPU*, G–1 to G–12
  - command window • *VAXTPU*, 4–16
  - $DEFAULTS$ buffer • *VAXTPU*, 4–32
  - initialization files • *VAXTPU*, 4–31 to 4–33
    - during a session • *VAXTPU*, 4–32
    - effects on buffer settings • *VAXTPU*, 4–32
  - Initialization files • *VAXTPU*, 5–10
  - input files • *VAXTPU*, 5–20
  - keypad emulation
    - EDT • *Programming Resources*, 1–5
    - numeric • *Programming Resources*, 1–5
    - VT100 • *Programming Resources*, 1–5
    - WPS • *Programming Resources*, 1–5
  - message buffer • *VAXTPU*, 4–18
  - message window • *VAXTPU*, 4–16
  - order of initialization • *VAXTPU*, G–4
  - output file • *VAXTPU*, 5–13, 5–20
  - restriction on defining GOLD key • *VAXTPU*, 7–472
  - sample procedures • *VAXTPU*, B–1 to B–33
  - source files • *VAXTPU*, 4–3
  - status line • *VAXTPU*, G–7
  - use of EDIT/TPU command qualifiers • *VAXTPU*, 5–18
  - user window • *VAXTPU*, 4–16
  - wildcard characters in file specifications • *VAXTPU*, 5–20
  - wildcards in file names • *VAXTPU*, 5–20
- EVE editor$BUILD • *VAXTPU*, G–1 to G–12
  - exit and quit handlers • *VAXTPU*, G–8
  - initialization modules • *VAXTPU*, G–4 to G–5
  - invoking • *VAXTPU*, G–10 to G–11
  - output • *VAXTPU*, G–11 to G–12
  - status line field • *VAXTPU*, G–7 to G–8
  - synonym creation • *VAXTPU*, G–5 to G–7
  - using parsing routines with • *VAXTPU*, G–3 to G–4
- EVE editor$GET_STATUS_FIELDS procedure • *VAXTPU*, G–8
- EVE editor$INIT logical name • *VAXTPU*, 4–31
- EVE editor$PARSER_DISPATCH procedure • *VAXTPU*, G–3
- EVE editor$SELECTION procedure
  - using to obtain EVE's current selection • *VAXTPU*, 4–17
- EVE editor default settings • *VAXTPU*, 4–32 to 4–33
- .EVEN directive • *MACRO*, 6–33
- Event
  - awaiting • *RTL Parallel Processing*, 4–7
  - creating • *RTL Parallel Processing*, 4–5
  - definition of • *RTL Parallel Processing*, 4–5
  - deleting • *RTL Parallel Processing*, 4–6
  - disabling • *RTL Parallel Processing*, 4–7
  - notification for abnormal exit • *RTL Parallel Processing*, 4–9
  - notification for normal exit • *RTL Parallel Processing*, 4–9
  - predefined • *RTL Parallel Prócessing*, 4–9
  - reading • *RTL Parallel Processing*, 4–8
  - resetting • *RTL Parallel Processing*, 4–8
  - triggering • *RTL Parallel Processing*, 4–8
- Event facility, setting • *Debugger*, CD–141, CD–220
- Event flag • *Programming Resources*, 4–1; *Modular Procedures*, 2–16; *Device Support (B)*, 1–39
  - See also Synchronization
  - allocation of • *RTL Library*, 2–17
  - clearing • *System Services Intro*, 4–4; *System Services*, SYS–63
  - cluster • *Programming Resources*, 4–1; *Routines Intro*, A–5t
  - common • *Programming Resources*, 4–1
  - for interprocess communication • *System Services Intro*, 8–10
  - for synchronous operations • *RMS*, 2–7
  - getting current status • *System Services*, SYS–385
  - handling for aborted I/O request • *Device Support (B)*, 3–11
  - local • *Programming Resources*, 3–2, 4–1
  - number • *Routines Intro*, A–5t
  - posting • *Device Support (A)*, 4–20
  - RTL routine to free • *RTL Library*, LIB–174
  - setting • *System Services Intro*, 4–4; *System Services*, SYS–401; *Device Support (A)*, 2–7
  - specifying • *System Services Intro*, 4–2
  - wait • *System Services Intro*, 4–3
  - waiting for entire set of • *System Services*, SYS–540
  - waiting for one of set • *System Services*, SYS–542
  - waiting for setting of • *System Services*, SYS–537
- Event flag cluster • *System Services Intro*, 4–2
  - associating with a process • *System Services*, SYS–15
  - deleting • *System Services Intro*, 4–5; *System Services*, SYS–146
  - disassociating • *System Services Intro*, 4–5; *System Services*, SYS–116
  - getting current status • *System Services*, SYS–385
  - number • *System Services Intro*, 4–2

Event flag cluster (Cont.)
  specifying name for • *System Services Intro,* 4–7
Event flag number • *System Services Intro,* 4–2
Event flag routines
  global symbols • *System Dump Analyzer,* SDA–60
Event flag service
  example using • *System Services Intro,* 4–8
Event notification
  pseudoterminal • *I/O User's I,* 9–6
Eventpoint
  See Breakpoint
  See Tracepoint
  See Watchpoint
/EVENT qualifier • *Debugger,* 3–15, CD–17, CD–30, CD–131, CD–188
Event synchronization
  See also Synchronization
  advantages and disadvantages • *RTL Parallel Processing,* 5–7
  PPL$ routines for • *RTL Parallel Processing,* 4–5 to 4–8
EVENT_FLAGS_AND_ASTS.EXE
  global symbols • *System Dump Analyzer,* SDA–60
EVE source files • *VAXTPU,* 1–11
Exact key match • *File Applications,* 8–11
EXACT keyword
  with LEARN_BEGIN • *VAXTPU,* 7–244
  with SEARCH • *VAXTPU,* 7–328
  with SEARCH_QUIETLY • *VAXTPU,* 7–333
EXACT_POSITIONING attribute • *File Def Language,* FDL–7
EXACT_POSITIONING secondary attribute • *File Applications,* 4–31
Examine
  address • *Debugger,* 4–25
    with DECwindows • *Debugger,* 1–26
  EXAMINE command • *Debugger,* 4–2, CD–85
  instruction • *Debugger,* 4–20, 11–9
    with DECwindows • *Debugger,* 1–26
  register • *Debugger,* 4–23, 11–4
    with DECwindows • *Debugger,* 1–26
  using vector mask • *Debugger,* 11–13
  variable • *Debugger,* 4–2, 4–15
    with DECwindows • *Debugger,* 1–25
  vector address expression • *Debugger,* 11–17
  vector instruction • *Debugger,* 11–9
  vector register • *Debugger,* 11–4
Examine button
  with DECwindows • *Debugger,* 1–9
EXAMINE command • *Debugger,* 4–2, CD–85; *Patch,* PAT–62 to PAT–64; *System Dump Analyzer,* SDA–16, SDA–24, SDA–51
EXAMINE/INSTRUCTION command • *Debugger,* 4–20, 7–9, C–5; *System Dump Analyzer,* SDA–23
EXAMINE/OPERANDS command • *Debugger,* 4–20, 11–9
EXAMINE/SOURCE command • *Debugger,* 6–4, 7–6, C–4
"Examine" string constant parameter to GET_INFO • *VAXTPU,* 7–179
Example program
  in VAX BLISS-32 • *RTL Parallel Processing,* 6–4
  in VAX C • *RTL Parallel Processing,* 6–14
  in VAX FORTRAN • *RTL Parallel Processing,* 6–9
Examples • *SUMSLP,* SUM–21
  See also PATCH command, qualifiers
  See also PATCH commands
  See also Using symbols
  adding lines • *SUMSLP,* SUM–8, SUM–9
  analyzing a file interactively • *Analyze/RMS_File,* ARMS–36
  analyzing a remote file • *Analyze/RMS_File,* ARMS–36
  appending a remote file • *Convert,* CONV–30
  audit trail text • *SUMSLP,* SUM–12
  converting a carriage control file to stream • *Convert,* CONV–30
  converting a carriage control file to variable length • *Convert,* CONV–30
  converting a remote file • *Convert,* CONV–29
  converting fixed format to variable length • *Convert,* CONV–30
  converting record formats • *Convert,* CONV–29
  creating an FDL file • *Analyze/RMS_File,* ARMS–36
  creating an FDL file from a remote file • *Analyze/RMS_File,* ARMS–36
  deleting lines • *SUMSLP,* SUM–9
  improving a file's performance • *Convert,* CONV–29
  interactive patch session • *Patch,* PAT–92
  listing file • *SUMSLP,* SUM–6
  modifying an FDL file • *File Def Language,* FDL–68
  modifying an FDL file noninteractively • *File Def Language,* FDL–68
  reclaiming buckets • *Convert,* CONV–29
  reorganizing a remote file • *Convert,* CONV–29
  tuning a file • *File Def Language,* FDL–68
Examples of DECwindows VAXTPU built-in procedures • *VAXTPU,* B–1 to B–33

Examples of VAXTPU procedures
ADJUST_HELP • *VAXTPU*, 7–23
ANCHOR • *VAXTPU*, 7–25
ANY • *VAXTPU*, 7–27
APPEND_LINE • *VAXTPU*, 7–29
ARB • *VAXTPU*, 7–31
ASCII • *VAXTPU*, 7–33, 7–34
BEGINNING_OF • *VAXTPU*, 7–38
BREAK • *VAXTPU*, 7–39
CALL_USER • *VAXTPU*, 7–42
CHANGE_CASE • *VAXTPU*, 7–46
COPY_TEXT • *VAXTPU*, 7–54
CREATE BUFFER • *VAXTPU*, 7–62
CREATE_KEY_MAP • *VAXTPU*, 7–64
CREATE_KEY_MAP_LIST • *VAXTPU*, 7–66
CREATE_PROCESS • *VAXTPU*, 7–68
CREATE_RANGE • *VAXTPU*, 7–71
CREATE_WINDOW • *VAXTPU*, 7–79
CURRENT_BUFFER • *VAXTPU*, 7–80
CURRENT_CHARCTER • *VAXTPU*, 7–82
CURRENT_COLUMN • *VAXTPU*, 7–84
CURRENT_DIRECTION • *VAXTPU*, 7–85
CURRENT_LINE • *VAXTPU*, 7–87
CURRENT_OFFSET • *VAXTPU*, 7–89
CURRENT_ROW • *VAXTPU*, 7–91
CURRENT_WINDOW • *VAXTPU*, 7–93
CURRSOR_HORIZONTAL • *VAXTPU*, 7–95
CURSOR_VERTICAL • *VAXTPU*, 7–98
DEFINE_KEY • *VAXTPU*, 7–103
DELETE • *VAXTPU*, 7–109
EDIT • *VAXTPU*, 7–114
END_OF • *VAXTPU*, 7–116
ERASE • *VAXTPU*, 7–118
ERASE_CHARACTER • *VAXTPU*, 7–120
ERROR • *VAXTPU*, 7–124
ERROR_LINE • *VAXTPU*, 7–126
ERROR_TEXT • *VAXTPU*, 7–128
EXECUTE • *VAXTPU*, 7–131, 7–132
EXPAND_NAME • *VAXTPU*, 7–137
FAO • *VAXTPU*, 7–139
FILE_PARSE • *VAXTPU*, 7–142
FILE_SEARCH • *VAXTPU*, 7–145
GET_INFO • *VAXTPU*, 7–160 to 7–161
HELP_TEXT • *VAXTPU*, 7–229
INDEX • *VAXTPU*, 7–231
INT • *VAXTPU*, 7–233
KEY_NAME • *VAXTPU*, 7–240
LENGTH • *VAXTPU*, 7–248
LINE_BEGIN • *VAXTPU*, 7–250
LINE_END • *VAXTPU*, 7–251
LOCATE_MOUSE • *VAXTPU*, 7–253

Examples of VAXTPU procedures (Cont.)
LOOKUP_KEY • *VAXTPU*, 7–256 to 7–257
MAP • *VAXTPU*, 7–260
MARK • *VAXTPU*, 7–263
MATCH • *VAXTPU*, 7–265
MESSAGE • *VAXTPU*, 7–269
MOVE_HORIZONTAL • *VAXTPU*, 7–279
MOVE_TEXT • *VAXTPU*, 7–281
MOVE_VERTICAL • *VAXTPU*, 7–283
NOTANY • *VAXTPU*, 7–285
PAGE_BREAK • *VAXTPU*, 7–286
POSITION • *VAXTPU*, 7–290
QUIT • *VAXTPU*, 7–292
READ_CHAR • *VAXTPU*, 7–294
READ_FILE • *VAXTPU*, 7–298
READ_KEY • *VAXTPU*, 7–302
REFRESH • *VAXTPU*, 7–311
REMAIN • *VAXTPU*, 7–312
RETURN • *VAXTPU*, 7–315
SAVE • *VAXTPU*, 7–318
SCAN • *VAXTPU*, 7–320 to 7–321
SCANL • *VAXTPU*, 7–323
SCROLL • *VAXTPU*, 7–326
SEARCH • *VAXTPU*, 7–330 to 7–331
SEARCH_QUIETLY • *VAXTPU*, 7–335 to 7–336
SELECT • *VAXTPU*, 7–339
SELECT_RANGE • *VAXTPU*, 7–341
SEND • *VAXTPU*, 7–343
SET (AUTO_REPEAT) • *VAXTPU*, 7–354
SET (BELL) • *VAXTPU*, 7–356
SET (DEBUG) • *VAXTPU*, 7–365
SET (LINE_NUMBER) • *VAXTPU*, 7–417
SET (SELF_INSERT) • *VAXTPU*, 7–471
SET (TEXT) • *VAXTPU*, 7–485
SET (TRACEBACK) • *VAXTPU*, 7–489
SLEEP • *VAXTPU*, 7–509
SPANL • *VAXTPU*, 7–514
SPLIT_LINE • *VAXTPU*, 7–519
STR • *VAXTPU*, 7–522
SUBSTR • *VAXTPU*, 7–524
TRANSLATE • *VAXTPU*, 7–528
UNANCHOR • *VAXTPU*, 7–531
UNDEFINE_KEY • *VAXTPU*, 7–533
UNMAP • *VAXTPU*, 7–537
UPDATE • *VAXTPU*, 7–539
WRITE_FILE • *VAXTPU*, 7–545
Exception • *MACRO*, E–1
See also Vector exception
access control violation • *MACRO*, E–4
arithmetic • *MACRO*, E–1
arithmetic type code • *MACRO*, E–1

- Exception (Cont.)
  - breakpoint • *MACRO*, E–8
  - change mode • *MACRO*, E–8
  - compatibility mode • *MACRO*, E–7
    - type code • *MACRO*, E–7
  - control • *MACRO*, 8–14
  - customer reserved opcode • *MACRO*, E–6
  - debugging • *Debugger*, 9–10
  - decimal
    - string overflow • *MACRO*, E–3
  - definition • *RTL Library*, 4–2
  - dispatcher • *System Services Intro*, 10–7
  - fatal • *System Dump Analyzer*, SDA–16
  - floating
    - divide-by-zero • *MACRO*, E–2, E–3
    - overflow • *MACRO*, E–2, E–3
    - underflow • *MACRO*, E–3, E–4
  - floating-point underflow • *RTL Library*, 4–31
  - generating • *Device Support (A)*, 5–4
  - generating on system service failure • *System Services*, SYS–423
  - how handled by Run-Time Library • *RTL Library*, 4–30
  - identifying causes of • *System Dump Analyzer*, SDA–21
  - instruction
    - emulation • *MACRO*, E–6
    - execution • *MACRO*, E–6
  - integer
    - divide-by-zero • *MACRO*, E–2
    - overflow • *MACRO*, E–2
  - kernel stack not valid • *MACRO*, E–10
  - machine check • *MACRO*, E–11
  - memory managment • *MACRO*, E–4
  - multiple • *System Services Intro*, 10–17
  - operand reference • *MACRO*, E–4
  - recovering from • *RTL Math*, 2–8
  - reserved
    - addressing mode • *MACRO*, E–4
    - operand • *MACRO*, E–4
  - subscript-range • *MACRO*, E–3
  - trace • *MACRO*, E–8
  - trace operation • *MACRO*, E–9
  - translation not valid • *MACRO*, E–4
  - type • *System Services Intro*, 10–1
  - vector processor • *MACRO*, 10–12, 10–28, 10–35
    - arithmetic • *MACRO*, 10–6, 10–22, 10–28, 10–30, 10–68
    - floating-point • *MACRO*, 10–68
    - memory management • *MACRO*, 10–28
- EXCEPTION.EXE
  - global symbols • *System Dump Analyzer*, SDA–60
- Exception breakpoint or tracepoint
  - canceling • *Debugger*, 9–12, CD–17, CD–30
  - qualifying • *Debugger*, 9–16, D–10
  - resuming execution at • *Debugger*, 9–12
  - setting • *Debugger*, 9–11, CD–131, CD–188
- Exception condition • *Routines Intro*, 1–12, 2–3, 2–44; *System Services Intro*, 10–1; *RTL Library*, 4–2, 4–4; *Convert*, CONV–3
  - handler • *Routines Intro*, 1–12, 2–45
  - indicating occurrence of • *Routines Intro*, 2–47
  - returning condition value • *RTL Library*, 4–4
  - signaling • *RTL Library*, 4–3, 4–5, 4–7, 4–16, 4–18, 4–23, 4–31
  - signaling an • *Routines Intro*, 2–47
- Exception Condition Type
  - See ETYPE
- Exception handler
  - debugger as • *Debugger*, 3–22
  - debugging • *Debugger*, 9–10
- Exception handling routines
  - global symbols • *System Dump Analyzer*, SDA–60
- /EXCEPTION qualifier • *Debugger*, 9–10, CD–17, CD–30, CD–131, CD–188, CD–265
- Exception record • *Convert*, CONV–3
- Exceptions file • *Convert*, CONV–3
- /EXCEPTIONS_FILE qualifier • *Convert*, CONV–9, CONV–26
- Exception vector
  - setting • *System Services*, SYS–402
- EXC file type • *Convert*, CONV–3
- Exclamation point (!)
  - as comment delimiter • *File Def Language*, FDL–40
  - comment delimiter • *Debugger*, CD–4
  - log file • *Debugger*, 8–5
- Exclusive OR operator • *MACRO*, 3–16
- %EXC_FACILITY • *Debugger*, 9–16, D–10
- %EXC_NAME • *Debugger*, 9–16, D–10
- %EXC_NUMBER • *Debugger*, 9–16, D–10
- %EXC_SEVERITY • *Debugger*, 9–16, D–10
- EXE$ABORTIO • *Device Support (A)*, 7–5, 18–14; *Device Support (B)*, 1–40, 3–7, 3–10 to 3–11, 3–33, 3–42, 3–44, 3–46, 3–50, 3–51, 3–55, 3–57, 3–59, 4–12
- EXE$ALLOCBUF • *Device Support (A)*, 7–6, 16–19; *Device Support (B)*, 3–12 to 3–13
- EXE$ALLOCIRP • *Device Support (B)*, 1–42, 1–44, 3–12 to 3–13
- EXE$ALONONPAGED • *Device Support (B)*, 3–13, 3–14, 3–61
- EXE$ALONPAGVAR • *Device Support (B)*, 3–15

EXE$ALOPHYCNTG • *Device Support (A)*, 16–21; *Device Support (B)*, 3–16
EXE$ALTQUEPKT • *Device Support (A)*, 7–5; *Device Support (B)*, 1–30, 3–5, 3–17, 4–2, 4–12
EXE$ASSIGN • *Device Support (A)*, 11–12; *Device Support (B)*, 1–11, 1–12, 4–6
EXE$BUFFRQUOTA
  replaced in VMS Version 5.0 • *Device Support (A)*, E–5
EXE$BUFQUOPRC
  replaced in VMS Version 5.0 • *Device Support (A)*, E–5
EXE$CANCEL • *Device Support (A)*, 11–7 to 11–8; *Device Support (B)*, 3–68
EXE$CREDIT_BYTCNT • *Device Support (A)*, 7–8, E–5; *Device Support (B)*, 3–18
EXE$CREDIT_BYTCNT_BYTLM • *Device Support (A)*, E–5; *Device Support (B)*, 3–18
EXE$DASSGN • *Device Support (B)*, 1–12
EXE$DEANONPAGED • *Device Support (B)*, 3–3, 3–13, 3–19
EXE$DEBIT_BYTCNT • *Device Support (A)*, E–5; *Device Support (B)*, 3–20 to 3–21
EXE$DEBIT_BYTCNT_ALO • *Device Support (A)*, 7–6, 16–19, E–6; *Device Support (B)*, 3–22 to 3–23
EXE$DEBIT_BYTCNT_BYTLM • *Device Support (A)*, 7–6, E–5; *Device Support (B)*, 3–20 to 3–21
EXE$DEBIT_BYTCNT_BYTLM_ALO • *Device Support (A)*, 7–6, 16–19, E–6; *Device Support (B)*, 3–22 to 3–23
EXE$DEBIT_BYTCNT_BYTLM_NW • *Device Support (A)*, E–6; *Device Support (B)*, 3–20 to 3–21
EXE$DEBIT_BYTCNT_NW • *Device Support (A)*, E–5; *Device Support (B)*, 3–20 to 3–21
EXE$FINISHIO • *Device Support (A)*, 7–4, 7–9, 18–14; *Device Support (B)*, 1–41, 3–24 to 3–25, 3–49, 3–50, 3–51, 4–12
EXE$FINISHIOC • *Device Support (A)*, 7–4; *Device Support (B)*, 1–41, 3–24 to 3–25, 4–12
EXE$FORK • *Device Support (A)*, 11–6; *Device Support (B)*, 1–21, 2–32, 3–26
EXE$FORKDSPTH • *Device Support (A)*, 3–5, 3–24; *Device Support (B)*, 1–73
EXE$GB_CPUTYPE • *Device Support (B)*, 2–10
EXE$GL_ABSTIM • *Device Support (B)*, 1–22
EXE$GL_CONFREGL • *Device Support (A)*, 16–7
EXE$GL_INTSTK
  replaced by CPU$L_INTSTK • *Device Support (B)*, 1–12
EXE$GQ_1ST_TIME • *Device Support (A)*, 3–8, 3–9, 3–13, 3–14; *Device Support (B)*, 3–29
EXE$GQ_SYSTIME • *Device Support (A)*, 3–8, 3–9, 3–14; *Device Support (B)*, 2–52, 3–69
  reading • *Device Support (A)*, E–15
EXE$HWCLKINT • *Device Support (A)*, 3–8
EXE$INSERTIRP • *Device Support (A)*, 4–13; *Device Support (B)*, 1–38, 1–39, 1–76, 3–27, 3–28, 3–38
EXE$INSIOQ • *Device Support (A)*, 3–23, 4–13, 7–4, 8–1; *Device Support (B)*, 1–77, 3–28, 3–38
  returning control to • *Device Support (A)*, 4–16
EXE$INSIOQC • *Device Support (B)*, 3–28
EXE$INSTIMQ • *Device Support (B)*, 3–29
EXE$IOFORK • *Device Support (A)*, 9–4, 10–1 to 10–2, 14–24; *Device Support (B)*, 1–72, 1–73, 3–30
EXE$MODIFY • *Device Support (A)*, 7–9; *Device Support (B)*, 3–31 to 3–33
EXE$MODIFYLOCK • *Device Support (B)*, 3–32, 3–34 to 3–36
EXE$MODIFYLOCKR • *Device Support (B)*, 1–43, 3–32, 3–34 to 3–36, 3–109
EXE$ONEPARM • *Device Support (A)*, 7–9; *Device Support (B)*, 1–41, 3–37
EXE$QIO • *Device Support (A)*, 4–1 to 4–13; *Device Support (B)*, 1–12, 1–30, 1–37 to 1–40, 1–42
EXE$QIOACPPKT • *Device Support (B)*, 1–74
EXE$QIODRVPKT • *Device Support (A)*, 4–13, 7–4, 7–9, 8–1; *Device Support (B)*, 3–32, 3–33, 3–37, 3–38, 3–41, 3–51, 3–55, 3–62, 4–12
EXE$QIORETURN • *Device Support (A)*, 18–14; *Device Support (B)*, 3–39
EXE$READ • *Device Support (A)*, 7–9; *Device Support (B)*, 1–41, 3–40 to 3–42
EXE$READCHK • *Device Support (A)*, 7–6; *Device Support (B)*, 3–43 to 3–44
EXE$READCHKR • *Device Support (B)*, 3–32, 3–35, 3–41, 3–43 to 3–44, 3–46
EXE$READLOCK • *Device Support (B)*, 3–41, 3–45 to 3–47
EXE$READLOCKR • *Device Support (B)*, 1–43, 3–41, 3–45 to 3–47, 3–109
EXE$RMVTIMQ • *Device Support (B)*, 3–48
EXE$SENSEMODE • *Device Support (A)*, 7–9; *Device Support (B)*, 3–49
EXE$SETCHAR • *Device Support (A)*, 7–9; *Device Support (B)*, 3–50 to 3–51
EXE$SETMODE • *Device Support (A)*, 7–9; *Device Support (B)*, 3–50 to 3–51
EXE$SNDEVMSG • *Device Support (A)*, 9–7 to 9–8, 10–7, E–7; *Device Support (B)*, 3–52 to 3–53
EXE$SWTIMINT • *Device Support (A)*, 3–8

EXE$TIMEOUT • *Device Support (B)*, 1–74, 1–77, 1–79
EXE$WRITE • *Device Support (A)*, 7–9; *Device Support (B)*, 1–41, 3–54 to 3–55
EXE$WRITECHK • *Device Support (A)*, 7–6; *Device Support (B)*, 3–56 to 3–57
EXE$WRITECHKR • *Device Support (B)*, 3–55, 3–56 to 3–57, 3–59
EXE$WRITELOCK • *Device Support (B)*, 3–55, 3–58 to 3–60
EXE$WRITELOCKR • *Device Support (B)*, 1–43, 3–55, 3–58 to 3–60, 3–109
EXE$WRTMAILBOX • *Device Support (B)*, 3–52, 3–61
EXE$ZEROPARM • *Device Support (A)*, 7–9; *Device Support (B)*, 1–41, 3–62
Executable image • *Linker*, 6–1; *Patch*, PAT–3
  linker parameters for creating • *Linker*, 1–1
  output of linker • *Linker*, 2–5
/EXECUTABLE qualifier • *Linker*, 1–5, 2–5, LINK–7
EXECUTE access • *File Def Language*, FDL–23
EXECUTE built-in procedure • *VAXTPU*, 4–19
Execute Command String command • *Delta/XDelta*, DELTA–38
Execute procedure • *System Dump Analyzer*, SDA–40
Execution
  as controlled by debugger • *Debugger*, 3–22
  discrepancies caused by debugger • *Debugger*, 3–23
  interrupting with Ctrl/C • *Debugger*, 2–7
  interrupting with Ctrl/Y • *Debugger*, 3–3
    with DECwindows • *Debugger*, 1–34
  interrupting with Stop button
    with DECwindows • *Debugger*, 1–9, 1–21
  monitoring with SHOW CALLS command • *Debugger*, 2–14, CD–214
  monitoring with tracepoint • *Debugger*, 3–10, CD–187
    with DECwindows • *Debugger*, 1–24
  multiprocess program • *Debugger*, 10–6, CD–152
  resuming after exception break • *Debugger*, 9–12
  starting or resuming with CALL command • *Debugger*, 8–11, 11–23, CD–10
  starting or resuming with GO command • *Debugger*, 2–13, CD–105
    with DECwindows • *Debugger*, 1–24
  starting or resuming with STEP command • *Debugger*, 3–7, CD–265
    with DECwindows • *Debugger*, 1–24
  suspending with breakpoint • *Debugger*, 3–9, CD–130
    with DECwindows • *Debugger*, 1–24
Execution (Cont.)
  suspending with exception breakpoint • *Debugger*, 9–11, CD–131
  suspending with watchpoint • *Debugger*, 3–17, 10–17, CD–200
    with DECwindows • *Debugger*, 1–25
  vectorized program • *Debugger*, 11–2
Execution context • *System Services Intro*, 8–2
Execution model
  vector processor • *MACRO*, 10–18
Executive image
  contents • *System Dump Analyzer*, SDA–60, SDA–104
  listing names and addresses • *Delta/XDelta*, DELTA–44
Executive mode
  changing to • *System Services*, SYS–64
Executive-mode (PSL$C_EXEC) constant
  for FAB$V_CHAN_MODE • *RMS*, 5–5
/EXECUTIVE qualifier • *System Dump Analyzer*, SDA–59, SDA–157
Executive stack pointer • *System Dump Analyzer*, SDA–13
Exit
  See also Exit handler
  abnormal • *RTL Parallel Processing*, 4–9
  forced • *System Services Intro*, 8–16
  image • *Programming Resources*, 9–26; *System Services Intro*, 8–14
  normal • *RTL Parallel Processing*, 4–9
$EXIT • *Debugger*, 9–16
EXIT built-in procedure • *VAXTPU*, 7–133 to 7–134
EXIT command • *Debugger*, 3–5, 9–16, CD–94; *Patch*, PAT–2, PAT–65; *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–26; *Delta/XDelta*, DELTA–45; *System Dump Analyzer*, SDA–55
  EDIT/FDL • *File Def Language*, FDL–61
  multiprocess program • *Debugger*, 10–9, 10–10
  with DECwindows • *Debugger*, 1–21
Exit handler • *Programming Resources*, 7–53, 9–26; *System Services Intro*, 8–15; *RTL Screen Management*, 4–3
  canceling • *System Services*, SYS–41
  control block • *System Services*, SYS–125
    deleting • *System Services*, SYS–41
  debugging • *Programming Resources*, 9–30; *Debugger*, 9–16, CD–94
  declaring • *System Services*, SYS–125
  establishing • *Programming Resources*, 9–27
  executing • *Debugger*, 3–5, CD–94
    with DECwindows • *Debugger*, 1–21
  execution sequence of • *Debugger*, 9–16

Exit handler (Cont.)
  identifying • *Debugger*, 9–17, CD–221
  writing • *Programming Resources*, 9–29
EXITIF statement • *VAXTPU*, 3–21 to 3–22
Exiting
  from ANALYZE/RMS_FILE • *Analyze/RMS_File*, ARMS–10
  from CONVERT • *Convert*, CONV–5
  from CONVERT/RECLAIM • *Convert*, CONV–5
  from CREATE/FDL • *File Def Language*, FDL–43
  from DELTA • *Delta/XDelta*, DELTA–2, DELTA–45
  from EDIT/FDL • *File Def Language*, FDL–43
  from SDA • *System Dump Analyzer*, SDA–55
  from XDELTA • *Delta/XDelta*, DELTA–8
EXITLOOP command • *Debugger*, 8–11, CD–97
/EXIT qualifier • *Debugger*, CD–78; *Convert*, CONV–10
exit_handler_block data type • *Routines Intro*, A–5t
EXPAND command • *Debugger*, 7–13, CD–98
Expanded string • *File Applications*, 6–4, 6–5
  requesting • *RMS*, 6–2
Expanded string area address
  See ESA
Expanded string area address field
  See NAM$L_ESA field
Expanded string length field
  See NAM$B_ESL field
Expanded string size field
  See NAM$B_ESS field
EXPAND keyword
  for /DATA qualifier • *National Char Set*, NCS–26
EXPAND_NAME built-in procedure • *VAXTPU*, 7–135 to 7–137
Expected interrupt
  See Device interrupt
EXPIRATION attribute • *File Def Language*, FDL–16
Expiration date field
  See also XAB$Q_EDT field
Explanatory text • *Routines Intro*, 1–4, 1–11
Exponential • *RTL Math*, MTH–65, MTH–90
  of complex number • *RTL Math*, MTH–31, MTH–33
Exponentiation
  complex base to complex exponent • *RTL General Purpose*, OTS–56
  complex base to signed integer exponent • *RTL General Purpose*, OTS–59
  D-floating base • *RTL General Purpose*, OTS–61, OTS–63, OTS–65
  F-floating base • *RTL General Purpose*, OTS–81, OTS–84, OTS–86

Exponentiation (Cont.)
  G-floating base • *RTL General Purpose*, OTS–67, OTS–70
  H-floating base • *RTL General Purpose*, OTS–72, OTS–74
  signed longword base • *RTL General Purpose*, OTS–77
  word base to word exponent • *RTL General Purpose*, OTS–76
Expression • *System Dump Analyzer*, SDA–11 to SDA–14; *MACRO*, 3–9; *VAXTPU*, 3–8 to 3–12
  See Address expression
  See also Numeric expression
  See Language expression
  absolute • *MACRO*, 3–9
  arithmetic • *VAXTPU*, 3–9
  Boolean • *VAXTPU*, 3–11
  evaluating • *System Dump Analyzer*, SDA–48
  evaluation by compiler • *VAXTPU*, 3–9
  evaluation of • *MACRO*, 3–9
  example of • *MACRO*, 3–10
  external • *MACRO*, 3–9
  global • *MACRO*, 3–9
  in message source file • *Message*, MSG–7
  pattern • *VAXTPU*, 3–11
  precedence in • *Delta/XDelta*, DELTA–9
  relational • *VAXTPU*, 3–10
  relocatable • *MACRO*, 3–9, 3–18
  types of • *VAXTPU*, 3–9
Extended attribute block
  See XAB
  See XAB block
Extended attribute block address field
  See FAB$L_XAB field
  See RAB$L_XAB field
Extended terminal operation option
  See ETO option
Extend service • *RMS*, RMS–35
  condition values • *RMS*, RMS–37
  control block input fields • *RMS*, RMS–36
  control block output fields • *RMS*, RMS–36
  invoking • *RMS*, 5–11
  requirements • *RMS*, RMS–36
  use restriction • *RMS*, RMS–36
  XAB overrides • *RMS*, RMS–36
Extend subfunction • *I/O User's I*, 1–11
/EXTEND_QUANTITY qualifier • *File Applications*, 9–8
Extensible VAX Editor
  See EVE editor

EXTENSION attribute • *File Def Language*, FDL–7, FDL–20
EXTENSION secondary attribute • *File Applications*, 4–31
Extension size • *File Applications*, A–1
  calculating • *File Applications*, 9–8
  performance • *File Applications*, 9–8, 9–9
Extent • *File Applications*, 1–4, 9–8
  defining section • *System Services Intro*, 11–10
  syntax • *MACRO*, 7–1
.EXTERNAL directive • *MACRO*, 6–34
External expression • *MACRO*, 3–9
External register base
  See MBA$L_ERB
External symbol • *MACRO*, 6–101
  attribute directive (.EXTERNAL) • *MACRO*, 6–34
  defining • *MACRO*, 6–22, 6–34
EXTRACT command • *Debugger*, 7–22, CD–101
%EXTRACT operator • *MACRO*, 4–10
/EXTRACT qualifier • *Librarian*, LIB–12, LIB–22
  for extracting definition modules from NCS library • *National Char Set*, NCS–28
  LIBRARY command • *Programming Resources*, 5–2
  using with /OUTPUT • *Librarian*, LIB–36
EXTV (Extract Field) instruction • *MACRO*, 9–39
EXTZV (Extract Zero Extended Field) instruction • *MACRO*, 9–39

F

F$SEARCH lexical function • *Device Support (A)*, 13–24
FAB$B_ACMODES
  See FAB$V_CHAN_MODE option and FAB$V_LNM_MODE option
FAB$B_BID field • *RMS*, 5–3
FAB$B_BKS field • *File Applications*, 3–24, 4–28, 7–19, 7–20; *File Def Language*, FDL–18; *RMS*, 5–3
  considerations for calculating • *RMS*, 5–4
  default logic • *RMS*, 5–4
  limitation for RMS–11 • *RMS*, 5–3
  performance considerations • *RMS*, 5–4
  requirements for RMS–11 compatibility • *RMS*, 5–5
  selecting default size for indexed files • *RMS*, 5–4
  variations for XABs • *RMS*, 5–4
FAB$B_BLN field • *RMS*, 5–4
FAB$B_BLS field • *File Applications*, 4–28
FAB$B_DEQ field • *File Applications*, 9–8
FAB$B_DNS field • *File Applications*, 9–7; *File Def Language*, FDL–19; *RMS*, 5–9, B–3
  specifying default file specification • *RMS*, 5–2
FAB$B_FAC field • *File Applications*, 9–6; *File Def Language*, FDL–2, FDL–3; *RMS*, 5–9
  comparing with FAB$B_SHR field • *RMS*, 5–9
  for specifying sharing options • *RMS*, 4–1
  interdependency with FAB$B_SHR field • *RMS*, 5–27
  list of options • *File Applications*, 7–3; *RMS*, 5–10
  use with FAB$B_SHR • *RMS*, 5–10
FAB$B_FNS field • *File Applications*, 6–5, 9–7; *File Def Language*, FDL–22; *RMS*, 5–12
  specifying primary file specification • *RMS*, 5–2
FAB$B_FSZ field • *File Applications*, 4–29; *File Def Language*, FDL–34; *RMS*, 5–18
FAB$B_ORG field • *File Applications*, 4–28; *File Def Language*, FDL–22; *RMS*, 5–23
FAB$B_RAT field • *File Applications*, 4–29; *File Def Language*, FDL–33, FDL–34; *RMS*, 5–23
FAB$B_RFM field • *File Applications*, 4–30; *File Def Language*, FDL–35; *RMS*, 5–25
FAB$B_RTV field • *File Applications*, 9–8, 9–10; *File Def Language*, FDL–25; *RMS*, 5–26
FAB$B_SHR field • *File Applications*, 9–6; *File Def Language*, FDL–37; *RMS*, 5–27
  comparing option names with file access option names • *RMS*, 5–27
  conflict with FAB$B_FAC field • *RMS*, 5–27
  default logic • *RMS*, 5–27
  FAB$V_MSE option • *File Applications*, 7–22
  FAB$V_SHRGET option • *File Applications*, 7–22
  FAB$V_UPI option • *File Applications*, 7–7
  for specifying sharing options • *RMS*, 4–1
  interdependency with FAB$B_FAC field • *RMS*, 5–27
  list of options • *File Applications*, 7–4; *RMS*, 5–28
  option naming convention • *RMS*, 5–27
FAB$C_FIX option • *RMS*, 5–25
FAB$C_STMCR option • *RMS*, 5–25
FAB$C_STMLF option • *RMS*, 5–26
FAB$C_STM option • *RMS*, 5–25
FAB$C_UDF option • *RMS*, 5–26
FAB$C_VAR option • *RMS*, 5–26
FAB$C_VFC option • *RMS*, 5–26
FAB$L_ALQ field • *File Applications*, 4–30; *File Def Language*, FDL–17; *RMS*, 5–3
  as output field • *RMS*, 5–3
  functional variations for XABs • *RMS*, 5–3
  setting at run time • *RMS*, 3–5
  use with Create service • *RMS*, 5–3
  use with Extend service • *RMS*, 5–3

- FAB$L_ALQ field (Cont.)
  - use with Open service • *RMS*, 5–3
- FAB$L_CTX field • *File Def Language*, FDL–18; *RMS*, 5–6
- FAB$L_DEV field • *RMS*, 5–7
  - bits listed • *RMS*, 5–7
- FAB$L_DNA field • *File Applications*, 6–4, 9–7; *File Def Language*, FDL–19; *RMS*, 5–8, 5–9, B–3
  - components listed • *RMS*, 5–9
  - specifying default file specification • *RMS*, 5–2
- FAB$L_FNA field • *File Applications*, 6–4, 6–5, 9–7; *File Def Language*, FDL–22; *RMS*, 5–11
  - specifying primary file specification • *RMS*, 5–2
- FAB$L_FOP field • *File Applications*, 4–27; *File Def Language*, FDL–18, FDL–19, FDL–20, FDL–21, FDL–22, FDL–23, FDL–24, FDL–25; *RMS*, 5–12
  - FAB$V_CBT option • *File Applications*, 4–31
  - FAB$V_CTG option • *File Applications*, 4–30
  - FAB$V_DFW option • *File Applications*, 3–14, 3–15, 3–27, 7–19, 7–20, 9–9
  - FAB$V_MXV option • *File Applications*, 4–27
  - FAB$V_NAM option • *File Applications*, 6–5
  - FAB$V_NEF option • *File Applications*, 8–15, 8–16
  - FAB$V_OFP option • *File Applications*, 6–9, 6–10
  - FAB$V_PPF option • *File Applications*, 6–20
  - FAB$V_RCK option • *File Applications*, 9–11
  - FAB$V_SQO option • *File Applications*, 9–10
  - FAB$V_TMP option • *File Applications*, 4–28
  - FAB$V_UFO option • *File Applications*, 7–4, 9–14
  - FAB$V_WCK option • *File Applications*, 9–11
  - list of options • *File Applications*, 9–14; *RMS*, 5–13
- FAB$L_MRN field • *File Applications*, 4–29; *File Def Language*, FDL–20; *RMS*, 5–21
- FAB$L_MRS field • *File Applications*, 4–29
- FAB$L_NAM field • *File Applications*, 6–9, 9–7; *RMS*, 5–23
- FAB$L_SDC field • *RMS*, 5–27
- FAB$L_STS field • *RMS*, 5–29
  - handling for ACL error status • *RMS*, 14–3
- FAB$L_STV field • *File Applications*, 9–14; *RMS*, 5–29
  - examples of using • *RMS*, 3–12
  - for invoking SYS$QIO • *RMS*, 5–18
  - for total number of blocks allocated • *RMS*, RMS–36
  - with I/O channel • *RMS*, RMS–16
- FAB$L_XAB field • *RMS*, 5–29
- FAB$V_ASY option • *RMS*, 5–14
- FAB$V_BIO option • *RMS*, 5–10
  - how used to specify I/O type • *RMS*, 4–24
- FAB$V_BLK option • *RMS*, 5–23
- FAB$V_BRO option
  - use for sharing files • *RMS*, 5–28
- FAB$V_CBT option • *RMS*, 5–13
  - precedence over FAB$V_CTG option • *RMS*, 5–14
- FAB$V_CHAN_MODE option
  - list of values • *RMS*, 5–5
  - setting from MACRO • *RMS*, 5–6
- FAB$V_CIF option • *RMS*, 5–15
  - precedence over FAB$V_SUP option • *RMS*, 5–15
- FAB$V_CR option • *RMS*, 5–24
  - restriction against use with FAB$V_FTN and FAB$V_PRN options • *RMS*, 5–23
- FAB$V_CTG option • *RMS*, 5–14
- FAB$V_DEL option • *RMS*, 5–10
  - for enabling Delete service • *RMS*, 4–20
- FAB$V_DFW option • *RMS*, 5–14
  - exception to use of global buffers • *RMS*, 5–19
- FAB$V_DLT option • *RMS*, 5–16
  - qualified use by Close service • *RMS*, RMS–4
- FAB$V_FTN option • *RMS*, 5–24
  - restriction against use with FAB$V_CR and FAB$V_PRN options • *RMS*, 5–23
- FAB$V_GET option • *RMS*, 5–10, 5–28
  - use with block I/O operations • *RMS*, 5–10, 5–11
- FAB$V_LNM_MODE option
  - values listed • *RMS*, 5–20
- FAB$V_LNM_MODE subfield • *RMS*, 5–20
- FAB$V_MSE option • *RMS*, 5–28
  - enabling multiple RABs • *RMS*, RMS–7
  - for overriding the FAB$V_UPI option • *RMS*, 5–29
  - requirement for read-only buffer cache • *RMS*, 5–20, 5–28
  - use with other options • *RMS*, 5–28
- FAB$V_MXV option • *RMS*, 5–15
- FAB$V_NAM option • *RMS*, 5–16
- FAB$V_NEF option • *RMS*, 5–17
- FAB$V_NFS option • *RMS*, 5–18
  - relationship to CHAN_MODE subfield • *RMS*, 5–5
- FAB$V_NIL option • *RMS*, 5–28
  - effect on specifying user file open option • *RMS*, 5–18
  - requirement for block I/O • *RMS*, 4–23
- FAB$V_OFP option • *RMS*, 5–16
- FAB$V_POS option • *RMS*, 5–17
  - subordinate to FAB$V_RWO option • *RMS*, 5–17, 5–18
- FAB$V_PRN option • *RMS*, 5–24
  - restriction against use with FAB$V_FTN and FAB$V_CR options • *RMS*, 5–23
- FAB$V_PUT option • *RMS*, 5–11, 5–28
  - use with block I/O operations • *RMS*, 5–10

FAB$V_RCK option • *RMS*, 5–15
FAB$V_RWC option • *RMS*, 5–17
FAB$V_RWO option • *RMS*, 5–17
  precedence over FAB$V_POS option • *RMS*, 5–17, 5–18
FAB$V_SCF option • *RMS*, 5–16
  qualified use by Close service • *RMS*, RMS–4
FAB$V_SHRDEL option • *RMS*, 5–28
FAB$V_SHRGET option
  requirement for read-only buffer cache • *RMS*, 5–20, 5–28
FAB$V_SPL option • *RMS*, 5–16
  qualified use by Close service • *RMS*, RMS–4
FAB$V_SQO option • *RMS*, 5–14
  prohibiting random access • *RMS*, RMS–48
FAB$V_SUP option • *RMS*, 5–16
  subordinate to FAB$V_CIF option • *RMS*, 5–15
FAB$V_SYNCSTS option • *RMS*, 5–15
FAB$V_TEF option • *RMS*, 5–14
FAB$V_TMD option • *RMS*, 5–17
  inhibiting automatic Create • *RMS*, RMS–29
FAB$V_TMP option • *RMS*, 5–17
  inhibiting automatic Create • *RMS*, RMS–29
FAB$V_TRN option
  in file access field • *RMS*, 5–11
  requirement for truncate-on-put operation • *RMS*, 7–17
FAB$V_UFO option • *RMS*, 5–18
  effect on internal structures • *RMS*, 5–20
  relationship to CHAN_MODE subfield • *RMS*, 5–5
FAB$V_UPD option • *RMS*, 5–11, 5–28
  requirement for implementing update-if option • *RMS*, 7–17
  requirement for Update service • *RMS*, 4–22
FAB$V_UPI option • *RMS*, 5–28
  reqirement for setting • *RMS*, 5–29
  requirement for block I/O • *RMS*, 4–23
  requirement for user file open option • *RMS*, 5–18
FAB$V_WCK option • *RMS*, 5–15
FAB$W_BLS field • *File Def Language*, FDL–21; *RMS*, 5–5
FAB$W_DEQ field • *File Applications*, 4–31, 9–9; *File Def Language*, FDL–20; *RMS*, 5–4, 5–6
  default logic • *RMS*, 5–6
  overriding default • *RMS*, 5–7
FAB$W_GBC field • *File Applications*, 7–17, 7–22, 9–9; *File Def Language*, FDL–20; *RMS*, 5–19
FAB$W_IFI field • *RMS*, 5–20
FAB$W_MRS field • *File Def Language*, FDL–35; *RMS*, 5–21
  as output • *RMS*, 5–22
  program example • *RMS*, 4–4
FAB$W_MRS field (Cont.)
  summary • *RMS*, 5–22
  use with fixed-length records • *RMS*, 5–21
  use with variable-length records • *RMS*, 5–21
FAB (file access block) • *Programming Resources*, 1–36, 8–58; *File Applications*, 1–11, 4–1; *RMS*, 5–1
  argument categories • *RMS*, 1–2
  description • *RMS*, 1–2
  requirements for • *RMS*, 5–2
  summary of fields • *RMS*, 5–1
FAB (file attributes block) • *System Dump Analyzer*, SDA–76
fab data type • *Routines Intro*, A–5t
$FABDEF • *File Applications*, 5–10
$FAB macro • *RMS*, B–2
  argument categories • *RMS*, B–3
$FAB_STORE macro • *RMS*, B–4
  argument categories • *RMS*, B–5
  FAB argument requirement • *RMS*, B–5
  run-time arguments • *RMS*, B–5
FAC field
  See FAB$B_FAC field
Facility
  creation • *Modular Procedures*, 5–1
  library • *Modular Procedures*, 3–2
  naming • *Modular Procedures*, 5–1
  naming conventions • *Modular Procedures*, 3–2
  number • *Modular Procedures*, 3–3
  prefix • *Modular Procedures*, 3–2, 5–1
.FACILITY directive • *Programming Resources*, 9–7
  in message source file • *Message*, MSG–18
  qualifiers • *Message*, MSG–18
Facility name
  in .FACILITY directive • *Message*, MSG–18
Facility number
  in .FACILITY directive • *Message*, MSG–18
Facility object module • *Message*, MSG–4
Facility-specific data type code • *Routines Intro*, 2–19
Facility-specific descriptor class codes • *Routines Intro*, 2–43
FACILITY_NAME keyword • *VAXTPU*, 7–378
“Facility_name” string constant parameter to GET_INFO • *VAXTPU*, 7–206
Failure exception mode • *System Services Intro*, 2–14
FALSE logical value • *File Def Language*, FDL–2
$FAO • *RTL Library*, 4–13, 4–16, 4–27
  RTL jacket routine for • *RTL Library*, LIB–404
FAO argument • *Message*, MSG–1, MSG–22, MSG–23
  signaling • *Programming Resources*, 9–12
FAO built-in procedure • *VAXTPU*, 7–138 to 7–139

FAO directives
- with MESSAGE • *VAXTPU*, 7–267
- with MESSAGE_TEXT • *VAXTPU*, 7–270

FAO parameter
- specifying • *Programming Resources*, 9–12

/FAO_COUNT qualifier
- in message definition • *Message*, MSG–22
- Message Utility • *Programming Resources*, 9–9

Fast-delete option • *File Applications*, 8–5, 9–9
- See also RAB$V_FDL option

FAST_DELETE attribute • *File Def Language*, FDL–10

/FAST_LOAD option
- compared with /NOFAST_LOAD option • *Convert*, CONV–11

/FAST_LOAD qualifier • *Convert*, CONV–11

Fatal exception • *System Dump Analyzer*, SDA–16

FATALEXCPT bugcheck • *System Dump Analyzer*, SDA–16

Fatal internal error
- resulting from exceeding virtual address space • *VAXTPU*, 5–1

/FATAL qualifier
- in message definition • *Message*, MSG–23

Fault
- access control violation • *MACRO*, E–4
- arithmetic • *MACRO*, E–1
- arithmetic type code • *MACRO*, E–1
- breakpoint • *MACRO*, E–8
- customer reserved opcode • *MACRO*, E–6
- fix floating reserved operand • *RTL Library*, LIB–165
- floating
  - divide-by-zero • *MACRO*, E–3
  - overflow • *MACRO*, E–2, E–3
  - underflow • *MACRO*, E–4
- instruction execution • *MACRO*, E–6
- memory management • *MACRO*, E–4
- privileged instruction • *MACRO*, E–6
- reserved
  - addressing mode • *MACRO*, E–4
  - opcode • *MACRO*, E–6
- trace • *MACRO*, E–8
- translation not valid • *MACRO*, E–4

FCB (file control block) • *System Dump Analyzer*, SDA–76

FDL$CREATE • *Programming Resources*, 8–57; *File Def Language*, FDL–41

FDL$CREATE routine • *Utility Routines*, FDL–6

FDL$GENERATE • *Programming Resources*, 8–55; *File Def Language*, FDL–41

FDL$GENERATE routine • *Utility Routines*, FDL–11

FDL$PARSE • *File Applications*, 9–1; *File Def Language*, FDL–41; *RMS*, 4–9

FDL$PARSE routine • *Utility Routines*, FDL–14

FDL$RELEASE • *RMS*, 4–9

FDL$RELEASE routine • *Utility Routines*, FDL–17

FDL (File Definition Language) • *Programming Resources*, 1–39, 8–54; *File Applications*, 1–11, 3–13, 4–2; *File Def Language*, FDL–1, FDL–42
- See also FDL file
- ACCESS attribute • *File Def Language*, FDL–2
- applying source • *Programming Resources*, 8–57
- attributes • *File Applications*, 4–2; *File Def Language*, FDL–1, FDL–46
- editor • *Programming Resources*, 8–55; *File Def Language*, FDL–42
- file type • *Analyze/RMS_File*, ARMS–16
- generating source • *Programming Resources*, 8–55
- library routine • *File Def Language*, FDL–41
- scripts • *File Applications*, 4–2
- syntax • *File Applications*, 4–2; *File Def Language*, FDL–39

FDL attribute
- predefined
  - using FDL$PARSE routine • *File Applications*, 9–1

FDL Editor • *File Applications*, 1–14
- See also Edit/FDL Utility
- as alternative to multiple XABs in example • *RMS*, 4–9

FDL file • *Programming Resources*, 1–39, 8–55; *File Def Language*, FDL–41, FDL–42, FDL–54
- ANALYSIS_OF_AREA section • *File Def Language*, FDL–3
- ANALYZE/RMS_FILE • *Analyze/RMS_File*, ARMS–14
- comment in • *File Def Language*, FDL–40
- created with ANALYZE/RMS_FILE • *File Def Language*, FDL–39
- creating • *Programming Resources*, 8–55; *File Applications*, 4–2; *Analyze/RMS_File*, ARMS–10, ARMS–14; *File Def Language*, FDL–39
- creating data files • *File Applications*, 4–17
- creating with FDL$GENERATE routine • *File Applications*, 4–15
- designing • *File Applications*, 4–11
- examining with ANALYZE/RMS_FILE • *File Applications*, 10–1
- generating from a data file • *File Applications*, 10–24
- using existing • *Programming Resources*, 8–55

- FDL file (Cont.)
  - with CONVERT • *Convert,* CONV–1
  - with EDIT/FDL • *File Def Language,* FDL–42, FDL–47
- FDL option • *File Def Language,* FDL–10
- /FDL qualifier • *File Applications,* 10–24
  - limitation • *Analyze/RMS_File,* ARMS–10, ARMS–13, ARMS–20
  - overview • *Analyze/RMS_File,* ARMS–14
  - using with /OUTPUT qualifier • *Analyze/RMS_File,* ARMS–16
  - with CONVERT • *Convert,* CONV–1, CONV–13
- FDL routine
  - creating data files • *File Def Language,* FDL–41
  - FDL$CREATE routine • *File Applications,* 4–15, 4–18, 6–3
  - FDL$GENERATE routine • *File Applications,* 4–15
  - FDL$PARSE routine • *File Applications,* 4–15, 6–3, 9–1
    - example • *File Applications,* 9–20 to 9–22
  - FDL$RELEASE routine • *File Applications,* 4–15, 6–3, 9–1
    - example • *File Applications,* 9–20 to 9–22
- FDL routines
  - See also FDL specification
  - examples • *Utility Routines,* FDL–1 to FDL–5
  - introduction • *Utility Routines,* FDL–1
- FDL specification
  - See also FDL routines
  - creating • *Utility Routines,* FDL–6
  - default attributes • *Utility Routines,* FDL–15
  - generating • *Utility Routines,* FDL–11
  - in character string • *Utility Routines,* FDL–8
    - use of semicolons as delimiters • *Utility Routines,* FDL–1
  - parsing • *Utility Routines,* FDL–14
  - with CONV routines • *Utility Routines,* CONV–15
- FDT (function decision table) • *Device Support (A),* 1–2, 4–10
  - address • *Device Support (A),* 4–8, 6–4; *Device Support (B),* 1–30
  - as used by EXE$QIO • *Device Support (A),* 4–8
  - creating • *Device Support (A),* 6–4 to 6–8, 11–4; *Device Support (B),* 2–37 to 2–38
  - dispatching to FDT routines from • *Device Support (A),* 4–13
  - relocating addresses specified in • *Device Support (A),* 11–4
  - size • *Device Support (B),* 1–31
  - specifying buffered functions in • *Device Support (A),* 4–11
  - specifying legal functions in • *Device Support (A),* 4–11
- FDT routine • *Device Support (A),* 1–3, 1–22 to 1–23, 2–3 to 2–4
  - adjusting process quotas in • *Device Support (B),* 3–12
  - allocating IRPE in • *Device Support (B),* 1–42
  - allocating system buffer in • *Device Support (A),* 7–6 to 7–7
  - calling sequence • *Device Support (A),* 7–2
  - completing an I/O operation in • *Device Support (B),* 3–24 to 3–25
  - context • *Device Support (A),* 4–13, 7–1; *Device Support (B),* 4–11
  - creating • *Device Support (A),* 7–1 to 7–5
  - dispatched to from EXE$QIO • *Device Support (A),* 4–12
  - ensuring an even byte count in • *Device Support (A),* 14–23
  - entry point • *Device Support (B),* 4–11
  - exit method • *Device Support (A),* 7–2 to 7–5; *Device Support (B),* 4–12
  - for buffered I/O • *Device Support (A),* 7–6 to 7–8
  - for direct I/O • *Device Support (A),* 7–6, 7–9; *Device Support (B),* 3–31 to 3–33, 3–40 to 3–42, 3–54 to 3–55
  - provided by VMS • *Device Support (A),* 7–8 to 7–9
  - register usage • *Device Support (A),* 5–3, 7–1; *Device Support (B),* 4–11
  - returning to the system service dispatcher • *Device Support (B),* 3–39
  - setting attention ASTs in • *Device Support (B),* 3–6
  - specifying • *Device Support (B),* 4–11
  - synchronization requirements • *Device Support (B),* 4–11
  - unlocking process buffers in • *Device Support (B),* 3–109
- FFC (Find First Clear) instruction • *MACRO,* 9–40
- F-floating conversion • *RTL Math,* 1–5
- FFS (Find First Set) instruction • *MACRO,* 9–40
- FFx instruction
  - RTL routine to access • *RTL Library,* LIB–147
- FIB (file information block) • *I/O User's I,* 1–3
  - See also ACP function
  - access control • *I/O User's I,* 1–10
  - contents • *I/O User's I,* 1–5 to 1–7
  - descriptor • *I/O User's I,* 1–2, 1–3
  - directory lookup • *I/O User's I,* 1–8
  - disk quota • *I/O User's I,* 1–33 to 1–34
  - extend control • *I/O User's I,* 1–11
  - format • *I/O User's I,* 1–5
  - IO$_ACCESS • *I/O User's I,* 1–26
  - IO$_ACPCONTROL • *I/O User's I,* 1–31 to 1–34

FIB (file information block) (Cont.)
- IO$_CREATE • *I/O User's I*, 1–23
- IO$_DEACCESS • *I/O User's I*, 1–28
- IO$_DELETE • *I/O User's I*, 1–30
- IO$_MODIFY • *I/O User's I*, 1–29
- truncate control • *I/O User's I*, 1–13

Field • *File Applications*, 1–1; *MACRO*, 2–1
- comment • *MACRO*, 2–1, 2–3
- label • *MACRO*, 2–1, 2–2
- must be zero (MBZ) • *MACRO*, 7–1
- operand • *MACRO*, 2–3
- operator • *MACRO*, 2–3
- read as zero (RAZ) • *MACRO*, 7–2
- should be zero (SBZ) • *MACRO*, 7–2
- variable-length bit • *MACRO*, 8–6

Field length
- identifier in symbolic name • *RMS*, 2–3

File • *File Applications*, 1–1
- See also File characteristic
- See also File protection
- See also File sharing
- See also File structure
- See Command procedure
- See Initialization file
- See Log file
- See Source file
- access in a VAXcluster • *File Applications*, 3–29
- access strategies • *Programming Resources*, 8–1
- adding lines to a • *SUMSLP*, SUM–7
- aligning • *File Applications*, 3–13
- analysis • *Analyze/RMS_File*, ARMS–10
- attributes • *Programming Resources*, 8–1, 8–3; *File Def Language*, FDL–1
- characteristics argument for FAB • *RMS*, 1–2
- compressing • *Programming Resources*, 8–26
- contiguity • *File Applications*, 3–4, 3–24
- corrupted • *File Applications*, 10–1; *Analyze/RMS_File*, ARMS–14
- creating • *File Def Language*, FDL–39
- creating FDL • *Analyze/RMS_File*, ARMS–10
- default name for journaling • *VAXTPU*, 1–12
- exceptions • *Convert*, CONV–3
- expanding • *Programming Resources*, 8–32
- extension • *File Applications*, 3–23
- extension size • *File Applications*, 3–5
- FDL • *File Applications*, 4–2, 4–17, 10–1, 10–24; *File Def Language*, FDL–42
- getting information about
  - asynchronously • *System Services*, SYS–257
  - synchronously • *System Services*, SYS–297
- header • *File Applications*, 3–9, 3–12, 3–15, 10–11

File (Cont.)
- how CONVERT processes • *Convert*, CONV–11
- indexed • *File Applications*, 10–28, 10–30
- initial allocation • *File Applications*, 3–4
- input source • *SUMSLP*, SUM–1
- insertion of • *Librarian*, LIB–27
- integrity • *Analyze/RMS_File*, ARMS–13
- internal structure • *File Applications*, 10–1; *Analyze/RMS_File*, ARMS–1
- listing • *SUMSLP*, SUM–3, SUM–6
- locking in a VAXcluster • *File Applications*, 3–29
- magnetic tape • *File Applications*, 1–9
- mapping • *Programming Resources*, 8–4
- merging • *Programming Resources*, 8–19
- modifying • *Programming Resources*, 8–58
- organization • *Analyze/RMS_File*, ARMS–1; *Convert*, CONV–1; *RMS*, 1–1
- output • *SUMSLP*, SUM–3
- reorganization • *Convert*, CONV–4
- sequential • *Programming Resources*, 8–10
- sorting • *Programming Resources*, 8–15
- specification argument for FAB • *RMS*, 1–2
- specifying one or many • *File Applications*, 5–16
- structure of • *Analyze/RMS_File*, ARMS–1
- temporary • *Convert*, CONV–27; *File Def Language*, FDL–19
- transferring to and from remote node • *Convert*, CONV–3
- update • *SUMSLP*, SUM–1
  - produced by DIFFERENCES/SLP DCL command • *SUMSLP*, SUM–3

File access
- category summary • *File Applications*, 4–21
- controlling through access control lists • *Utility Routines*, ACL–1
- defaults • *File Applications*, 7–5
- options • *File Applications*, 4–21
- protection • *Routines Intro*, A–5t

File access block • *Routines Intro*, A–5t
- See FAB

File access block address field
- See RAB$L_FAB field

File access field
- See FAB$B_FAC field

FILE attribute • *File Def Language*, FDL–2, FDL–16

FILE ATTRIBUTES structure • *File Applications*, 10–12, 10–16, 10–19

File characteristic • *File Applications*, 4–14, 4–27, 4–28

File characteristics
- ACP-QIO attributes • *I/O User's I*, 1–18

File component descriptor
  address field • *RMS*, 6–3
  example • *RMS*, 6–4
  field value logic • *RMS*, 6–3
  list of • *RMS*, 6–3
  size field • *RMS*, 6–3
  suggested use of • *RMS*, 6–4
File control block
  See FCB
File Definition Language
  See FDL
File Definition Language Editor
  See FDL Editor
File Definition Language routines
  See FDL routines
File design
  attributes • *File Applications*, 3–4
File disposition • *File Applications*, 9–12
File extension
  using Extend service • *RMS*, RMS–36
File header • *File Applications*, 1–7; *Analyze/RMS_File*, ARMS–1
File header characteristic extended address block
  See XABFHC block
FILE HEADER structure • *File Applications*, 10–12, 10–16, 10–19
File identification field
  See NAM$W_FID field
File management • *Programming Resources*, 1–23
File name address descriptor
  See NAM$L_NAME descriptor
File name address field
  See NAM$L_NAME field
File name length field
  See NAM$B_NAME field
File name size descriptor
  See NAM$B_NAME descriptor
File name status field
  See NAM$L_FNB field
File name string
  component parts • *RMS*, 4–9
File name string address (FAB$L_FNA) field
  how used to specify file name string • *RMS*, 4–9
File name string size (FAB$B_FNS) field
  how used to specify file name size • *RMS*, 4–9
File-opening option
  See also Creation-time option
  adding records • *File Applications*, 9–10 to 9–11
  data reliability • *File Applications*, 9–11
  file access and sharing • *File Applications*, 9–6 to 9–7
File-opening option (Cont.)
  file disposition • *File Applications*, 9–12
  file performance • *File Applications*, 9–7 to 9–10
  file specification • *File Applications*, 9–7
  for indexed files • *File Applications*, 9–12 to 9–13
  for magnetic tape processing • *File Applications*, 9–13 to 9–14
  for nonstandard file processing • *File Applications*, 9–14
  record access • *File Applications*, 9–10
File organization • *File Applications*, 1–2, 2–13; *VAXTPU*, F–1
  changing with CONV routines • *Utility Routines*, CONV–1
  selecting • *File Applications*, 2–1
File organization and record format field
  See XAB$B_RFO field
File organization field
  See FAB$B_ORG field
File organization option • *File Applications*, 4–28
File owner group number field
  See also XAB$W_GRP field
  in XABPRO field • *RMS*, 14–4
File owner member number field
  See XAB$W_MBM field
File positioning • *File Applications*, 4–30
  effect on shared files • *RMS*, RMS–7
FILE primary attribute
  ALLOCATION secondary attribute • *File Applications*, 3–4, 3–24, 4–30
  BEST_TRY_CONTIGUOUS secondary attribute • *File Applications*, 3–4, 4–31
  BUCKET_SIZE secondary attribute • *File Applications*, 3–13, 3–24, 4–28, 7–19, 7–20
  CONTIGUOUS secondary attribute • *File Applications*, 3–4, 3–24, 4–30
  CONTROL_FIELD_SIZE secondary attribute • *File Applications*, 4–29
  CREATE_IF secondary attribute • *File Applications*, 4–27
  DEFAULT_NAME secondary attribute • *File Applications*, 6–4, 9–7
  DEFERRED_WRITE secondary attribute • *File Applications*, 3–14, 3–27, 7–19, 7–20, 9–9
  DIRECTORY_ENTRY secondary attribute • *File Applications*, 4–28
  EXTENSION secondary attribute • *File Applications*, 3–5, 4–31, 9–8, 9–9
  GLOBAL_BUFFER_COUNT secondary attribute • *File Applications*, 3–9, 7–17, 7–22
  MAXIMIZE_VERSION secondary attribute • *File Applications*, 4–27

FILE primary attribute (Cont.)

MAX_RECORD_NUMBER secondary attribute • *File Applications*, 4–29

MT_BLOCK_SIZE secondary attribute • *File Applications*, 4–28

MT_CLOSE_REWIND secondary attribute • *File Applications*, 9–14

MT_CURRENT_POSITION secondary attribute • *File Applications*, 9–14

MT_NOT_EOF secondary attribute • *File Applications*, 9–14

MT_OPEN_REWIND secondary attribute • *File Applications*, 9–14

MT_PROTECTION secondary attribute • *File Applications*, 4–28

NAME secondary attribute • *File Applications*, 6–4, 9–7

NON_FILE_STRUCTURED secondary attribute • *File Applications*, 9–14

ORGANIZATION secondary attribute • *File Applications*, 4–28

OWNER secondary attribute • *File Applications*, 4–28

PRINT_ON_CLOSE secondary attribute • *File Applications*, 9–12

PROTECTION secondary attribute • *File Applications*, 4–28

READ_CHECK secondary attribute • *File Applications*, 9–11

REVISION secondary attribute • *File Applications*, 4–28

SEQUENTIAL_ONLY secondary attribute • *File Applications*, 9–10

SUBMIT_ON_CLOSE secondary attribute • *File Applications*, 9–12

SUPERSEDE secondary attribute • *File Applications*, 4–27

TEMPORARY secondary attribute • *File Applications*, 4–27

USER_FILE_OPEN secondary attribute • *File Applications*, 7–4, 9–14

WINDOW_SIZE secondary attribute • *File Applications*, 9–8, 9–10

WRITE_CHECK secondary attribute • *File Applications*, 9–11

File processing • *SUMSLP*, SUM–7

many files • *File Applications*, 5–15 to 5–16

nonstandard file • *File Applications*, 9–14

services listed • *RMS*, 3–3

single file • *File Applications*, 5–14 to 5–15

File-processing option

as service output • *RMS*, 5–12

categories listed • *RMS*, 5–12

File-processing option (Cont.)

naming convention • *RMS*, 5–12

File-processing options field

See FAB$L_FOP field

File protection • *File Applications*, 4–28; *File Def Language*, FDL–23

File protection extended address block

See XABPRO block

File protection field

See XAB$W_PRO field

File protection option field

See XAB$B_PROT_OPT field

File qualifier

/OPTIONS • *Linker*, 1–5

Files

Prolog 3 indexed files • *Utility Routines*, CONV–1, CONV–18

Files–11 On-Disk Structure • *File Applications*, 1–3

file headers • *File Applications*, 1–7

home block • *File Applications*, 1–7

index file • *File Applications*, 1–7

File section

defining in context of multiple volumes • *RMS*, RMS–56

File sharing • *File Applications*, 3–8, 9–6

compatibility with subsequent record access • *File Applications*, 7–5 to 7–6

defaults • *File Applications*, 7–5

features • *RMS*, 1–1

interlocked interprocess • *File Applications*, 7–2, 7–5 to 7–6

multistreaming • *File Applications*, 7–2, 7–4

no-access function • *File Applications*, 7–4

options • *File Applications*, 7–4

user-interlocked interprocess • *File Applications*, 7–2, 7–4, 7–7

File-sharing field

See FAB$B_SHR field

File specification • *File Applications*, 6–3; *Convert*, CONV–5; *File Def Language*, FDL–19

See also Default file specification

applicable services and routines • *File Applications*, 5–8 to 5–14

component descriptors • *RMS*, 6–2

components • *File Applications*, 5–1 to 5–2

default

See also Default file specification

default requirements • *RMS*, 4–9

directory • *File Applications*, 6–12 to 6–20

for a command procedure • *Patch*, PAT–48

format • *File Applications*, 5–1 to 5–4, 6–5 to 6–7

File specification (Cont.)
  for remote files • *File Applications,* 5–2 to 5–4, 5–8
  how handled by Search service • *RMS,* 4–9
  input • *File Applications,* 6–10
  maximum length • *File Applications,* 5–2
  output • *File Applications,* 6–10
  parsing • *System Services,* SYS–179; *RMS,* RMS–66
  partial • *File Def Language,* FDL–19
  preprocessing • *File Applications,* 5–8
  primary • *File Applications,* 5–4, 6–1 to 6–4, 9–7
  process default • *File Applications,* 5–4
  program-supplied • *File Applications,* 5–4, 6–1 to 6–4
  related • *File Applications,* 5–4, 6–1 to 6–4, 6–9, 9–7
  searching string for • *System Services,* SYS–179
  using • *File Applications,* 5–1
  using logical name • *File Applications,* 6–5 to 6–7
  using name block • *File Applications,* 5–8
  using search lists • *File Applications,* 5–8 to 5–16, 6–7 to 6–8
  using SYS$DISK • *File Applications,* 6–2
  using wildcard characters • *File Applications,* 5–8 to 5–16
File specification address
  See FAB$L_FNA field
File specification parsing • *File Applications,* 5–7 to 5–8, 6–4 to 6–12
  conventions used by VMS RMS • *File Applications,* 6–4 to 6–12
  for input file • *File Applications,* 6–10
  for output file • *File Applications,* 6–10
  for related file • *File Applications,* 6–9
  logical name • *File Applications,* 6–5 to 6–7
  search list • *File Applications,* 6–7 to 6–8
File specifications
  with CONV routines • *Utility Routines,* CONV–12
File specification size
  See FAB$B_FNS field
File specification string address • *RMS,* 4–9
File specification string size • *RMS,* 4–9
File structure • *File Applications,* 10–11; *Analyze/RMS_File,* ARMS–1
  analyzing interactively • *Analyze/RMS_File,* ARMS–1
  examining • *Analyze/RMS_File,* ARMS–15
File structured device • *Device Support (B),* 1–74
File system
  synchronizing access to • *Device Support (A),* 3–13
File system ACP • *I/O User's I,* 1–1
File terminator • *Programming Resources,* 7–54
File tuning
  See Tuning
File type • *Convert,* CONV–5
  ANL • *File Applications,* 10–5; *Analyze/RMS_File,* ARMS–16
  DAT • *Analyze/RMS_File,* ARMS–10
  default for command definition file • *Command Def,* CDU–4
  default for input files • *National Char Set,* NCS–21
  EXC • *Convert,* CONV–3
  FDL • *Analyze/RMS_File,* ARMS–16
  used for linker input • *Linker,* 1–4
File type address descriptor
  See NAM$L_TYPE descriptor
File type address field
  See NAM$L_TYPE field
File type length field
  See NAM$B_TYPE field
File type size descriptor
  See NAM$B_TYPE descriptor
File version address descriptor
  See NAM$L_VER descriptor
File version address field
  See NAM$L_VER field
File version length field
  See NAM$B_VER field
File version limit field
  See XAB$W_VERLIMIT field
File version size descriptor
  See NAM$B_VER descriptor
File work area
  See FWA
FILE_ID option • *File Applications,* 4–31
FILE_MONITORING attribute • *File Def Language,* FDL–20
FILE_NAME option • *File Applications,* 4–31
/FILE_NAME qualifier • *Message,* MSG–10
"File_name" string constant parameter to GET_ INFO • *VAXTPU,* 7–171, 7–177
FILE_PARSE built-in procedure • *VAXTPU,* 7–140 to 7–142
file_protection data type • *Routines Intro,* A–5t
FILE_SEARCH built-in procedure • *VAXTPU,* 7–143 to 7–145
FILL built-in procedure • *VAXTPU,* 7–146 to 7–148
Fill factor • *File Applications,* 3–26; *File Def Language,* FDL–5, FDL–28

Fill level
  comparing primary key and alternate keys • *RMS,* 13–10
FILL_BUCKETS attribute • *File Def Language,* FDL–10
/FILL_BUCKETS qualifier • *Convert,* CONV–14; *File Def Language,* FDL–27, FDL–28
FILSYS spin lock • *Device Support (A),* 3–13
Final handler • *Debugger,* 9–13
Find service • *File Applications,* 8–1, 8–2 to 8–3; *RMS,* RMS–38
  and key matches • *File Applications,* 8–10
  capabilities • *RMS,* RMS–39
  compared with Get service • *File Applications,* 8–2
  condition values • *RMS,* RMS–41
  control block input fields • *RMS,* RMS–39
  control block output fields • *RMS,* RMS–41
  effect on next-record position • *File Applications,* 8–16
  high-level language equivalents • *File Applications,* 8–1
  improved performance • *File Applications,* 8–3
  requirement for end-of-file test • *File Applications,* 8–3
  run-time options • *File Applications,* 9–14 to 9–17
"Find_buffer" string constant parameter to GET_INFO • *VAXTPU,* 7–169
FIND_CPU_DATA macro • *Device Support (A),* E–6; *Device Support (B),* 2–31
  example • *Device Support (B),* 2–31
Fine granularity • *RTL Parallel Processing,* 5–2
"first" string parameter to ADD_KEY_MAP • *VAXTPU,* 7–17
FIRST command • *File Applications,* 10–12; *Analyze/RMS_File,* ARMS–27
First data bucket start virtual block number field
  See XAB$L_DVB field
First free byte field
  See XAB$W_FFB field
First in first out (FIFO) queue • *RTL Parallel Processing,* 4–16, 4–18
First Order Linear Recurrence • *RTL Math,* MTH–192, MTH–197, MTH–201, MTH–205
"First" string constant parameter to GET_INFO • *VAXTPU,* 7–166, 7–167, 7–169, 7–181, 7–183, 7–184, 7–191, 7–218
First-time flag
  testing and setting • *Modular Procedures,* 3–14
"First_marker" string constant parameter to GET_INFO • *VAXTPU,* 7–172
"First_range" string constant parameter to GET_INFO • *VAXTPU,* 7–172
Fixed control • *File Def Language,* FDL–34, FDL–35
FIXED format • *File Def Language,* FDL–35
Fixed-length cell • *File Applications,* 3–12
Fixed-length control area size field
  See FAB$B_FSZ field
Fixed-length control field • *File Applications,* 3–12
  size option • *File Applications,* 4–28
Fixed-length descriptor • *Routines Intro,* 2–23
Fixed-length header control size field
  See XAB$B_HSZ field
Fixed-length record • *Convert,* CONV–18, CONV–26; *File Def Language,* FDL–35
Fixed-length record format option
  See FAB$C_FIX option
Fixed-length string • *RTL String Manipulation,* 2–1
/FIXED_CONTROL qualifier • *Convert,* CONV–15
Fix-up image section
  condition for insertion of • *Linker,* 6–20
  creation of • *Linker,* 6–20
  in relation to code reference • *Linker,* 6–21
  purpose of • *Linker,* 6–20, 6–21
Flag
  See also Event flag
Flag word • *Routines Intro,* A–10t
FLG=CHG option • *File Def Language,* FDL–26
FLG=DUP option • *File Def Language,* FDL–28
FLG=NUL option • *File Def Language,* FDL–29
.FLOAT directive • *MACRO,* 6–35
Floating address • *Device Support (A),* 12–14
Floating CSR space
  assigning to device • *Device Support (A),* 12–22
  current base • *Device Support (A),* 12–22
Floating overflow fault • *MACRO,* 8–16
Floating-point
  accuracy • *MACRO,* 9–103
  rounding • *MACRO,* 9–104
  zero • *MACRO,* 9–102
Floating-point constants (.D_FLOATING) • *MACRO,* 6–20
Floating-point conversion to nearest value • *RTL Math,* 1–8
Floating-point data type • *MACRO,* 8–3, 9–101
  D_floating • *MACRO,* 8–4
  G_floating • *MACRO,* 8–4
  H_floating • *MACRO,* 8–5
Floating-point emulation code
  base address • *System Dump Analyzer,* SDA–13
Floating-point instructions • *MACRO,* 9–101
  in device driver • *Device Support (A),* 5–3
  vector • *MACRO,* 10–68
Floating-point multiplication • *RTL Math,* 1–8
Floating-point number • *MACRO,* 9–101

Floating-point number (Cont.)
  D_floating complex • *Routines Intro,* A–3t
  D_floating standard • *Routines Intro,* A–6t
  format • *MACRO,* 3–3
  .F_FLOATING • *MACRO,* 6–35
  F_floating complex • *Routines Intro,* A–3t
  F_floating standard • *Routines Intro,* A–6t
  .G_FLOATING • *MACRO,* 6–36
  G_floating complex • *Routines Intro,* A–4t
  G_floating standard • *Routines Intro,* A–7t
  .H_FLOATING • *MACRO,* 6–38
  H_floating standard • *Routines Intro,* A–7t
  in source statement • *MACRO,* 3–3
  rounding • *MACRO,* 6–23
  storage • *MACRO,* 6–20
  storing • *MACRO,* 6–35, 6–36, 6–38
  truncating • *MACRO,* 6–23
Floating-point operator • *MACRO,* 3–14
Floating-point positive difference • *RTL Math,* 1–5
Floating-point sign function • *RTL Math,* 1–9
Floating-point storage directive
  .D_FLOATING • *MACRO,* 6–20
  (.F_FLOATING) • *MACRO,* 6–35
  (.G_FLOATING) • *MACRO,* 6–36
Floating-point underflow • *RTL Library,* 4–31
Floating underflow enable (FU) • *MACRO,* 8–16
Floating vector space
  assigning to device • *Device Support (A),* 12–22
  current base • *Device Support (A),* 12–22
floating_point data type • *Routines Intro,* A–6t
/FLOAT qualifier • *Debugger,* CD–62, CD–86
Floppy disk
  See Diskette
Flush service • *File Applications,* 7–7, 8–5; *RMS,* RMS–43, RMS–44
  condition values • *RMS,* RMS–44
    See also Completion status code
  control block input fields • *RMS,* RMS–44
  control block output fields • *RMS,* RMS–44
/FMASK qualifier • *Debugger,* 11–13, CD–88
FNA argument • *RMS,* B–5
FNM argument • *RMS,* B–3
FNM keyword
  for specifying FAB$L_FNA and FAB$B_FNS fields from VAX MACRO • *RMS,* 5–11
FNS argument • *RMS,* B–5
FOLR
  definition of • *RTL Math,* 2–7
FOLR routine • *RTL Math,* MTH–192, MTH–197, MTH–201, MTH–205
FOLR routines
  error checking • *RTL Math,* 2–7
  naming conventions • *RTL Math,* 2–7
Forced exit • *System Services Intro,* 8–16
FOR command • *Debugger,* 8–10, CD–103
Foreign command • *RTL Library,* 2–3
Foreign command name
  use of dollar sign • *RTL Library,* 2–4
Foreign device • *System Services Intro,* 7–7
Foreign terminal
  definition • *RTL Screen Management,* 5–1
  input support • *RTL Screen Management,* 5–23
Foreign volume • *System Services Intro,* 7–4, 7–5, 7–7
Fork block • *Device Support (A),* 1–5, 1–8, 3–24, 3–27, 4–16, 8–7, 10–1; *Device Support (B),* 2–104, 3–26, 3–30, 3–104 to 3–106
  dequeuing • *Device Support (A),* 3–5
  in CRB • *Device Support (A),* 12–7; *Device Support (B),* 1–21
  in extended UCB • *Device Support (A),* 11–6
  in UCB • *Device Support (B),* 1–72 to 1–73
Fork context • *Device Support (A),* 1–8, 3–22 to 3–23, 4–16
Fork database • *Device Support (A),* 3–5
  accessing • *Device Support (B),* 2–33 to 2–34
  synchronizing access to • *Device Support (A),* 3–22 to 3–25
Fork dispatcher • *Device Support (A),* 2–6, 3–3, 3–5, 3–8, 3–24; *Device Support (B),* 2–33
  functions • *Device Support (A),* 4–18
Forking • *Device Support (A),* 3–16, 3–23, E–9; *Device Support (B),* 2–32, 2–43, 3–26, 3–30
  avoiding multiple • *Device Support (A),* 11–6
  from controller initialization routine • *Device Support (A),* 11–6; *Device Support (B),* 4–8
  from driver unloading routine • *Device Support (B),* 4–10
  from interrupt service routine • *Device Support (A),* 9–5
  from unit initialization routine • *Device Support (A),* 11–6; *Device Support (B),* 4–22
  in terminal port driver • *Device Support (A),* 18–14, 18–20
Fork IPL • *Device Support (A),* 2–4, 3–2, 3–5, 3–16, 3–22, 4–18; *Device Support (B),* 1–73, 2–33 to 2–34
Fork lock • *Device Support (A),* 2–4, 3–6, 3–8, 3–13, 3–16, 3–22, 11–7, 14–16; *Device Support (B),* 1–21, 1–68
  See also Spin lock
  acquisition IPL • *Device Support (B),* 3–111

Fork lock (Cont.)
- multiple acquisition of • *Device Support (B)*, 2–35, 3–116
- obtained by fork dispatcher • *Device Support (A)*, 3–5
- obtaining • *Device Support (A)*, 3–10; *Device Support (B)*, 2–33 to 2–34, 3–111 to 3–112
- ownership • *Device Support (A)*, 13–30
- rank • *Device Support (A)*, 3–13 to 3–14
- releasing • *Device Support (A)*, 3–10; *Device Support (B)*, 2–35 to 2–36, 3–114
- restoring • *Device Support (B)*, 2–35, 3–116

Fork lock index • *Device Support (A)*, 3–13 to 3–14; *Device Support (B)*, 1–73
- list • *Device Support (A)*, E–8
- placing in UCB$B_FLCK • *Device Support (A)*, 6–2, E–8; *Device Support (B)*, 2–25

FORKLOCK macro • *Device Support (A)*, 3–9, 3–10, E–4; *Device Support (B)*, 2–33 to 2–34, 3–111
- example • *Device Support (B)*, 2–34

FORK macro • *Device Support (A)*, 3–12, 3–24, 14–18, 14–20; *Device Support (B)*, 2–32, 3–26
- See also IOFORK macro

Fork process • *Device Support (A)*, 1–8, 3–22 to 3–25, 8–1
- context • *Device Support (A)*, 4–15, 4–16, 4–17, 8–1 to 8–2
- creating • *Device Support (B)*, 2–32, 2–43, 3–26, 3–30
- creation by driver • *Device Support (A)*, 2–6, 4–17, 10–1 to 10–2
- creation by IOC$INITIATE • *Device Support (A)*, 4–13 to 4–15, 8–1, 10–3; *Device Support (B)*, 3–70 to 3–71
- reactivating • *Device Support (A)*, 4–18
- rules • *Device Support (A)*, 3–24
- suspending • *Device Support (A)*, 4–16, 8–6 to 8–7; *Device Support (B)*, 2–104, 3–104 to 3–106

Fork queue • *Device Support (A)*, 3–24, 4–17, 4–18, E–14; *Device Support (B)*, 1–17, 1–72, 3–26, 3–30

FORKUNLOCK macro • *Device Support (A)*, 3–10, E–4; *Device Support (B)*, 2–35 to 2–36, 3–114, 3–116
- example • *Device Support (B)*, 2–34

Form
- getting information about
  - asynchronously • *System Services*, SYS–257
  - synchronously • *System Services*, SYS–297

Formal argument • *MACRO*, 4–1

Format
- for DEFINE SYNTAX statement • *Command Def*, CDU–5
- for DEFINE TYPE statement • *Command Def*, CDU–7
- for DEFINE VERB statement • *Command Def*, CDU–8
- for definition path • *Command Def*, CDU–12
- for DISALLOW verb clause • *Command Def*, CDU–9
- for IDENT statement • *Command Def*, CDU–14
- for LINK command • *Linker*, 1–2
- for MODULE statement • *Command Def*, CDU–14
- for SET COMMAND command • *Command Def*, CDU–18
- of fixed-length record • *Convert*, CONV–18
- of hexadecimal dump • *Analyze/RMS_File*, ARMS–25
- of LIBRARY command • *Librarian*, LIB–11
- of message source file statements • *Message*, MSG–3

FORMAT attribute • *File Def Language*, FDL–35

FORMAT command • *System Dump Analyzer*, SDA–26, SDA–56, SDA–64

Format heading • *Routines Intro*, 1–2
- See also System routine documentation

/FORMAT qualifier • *National Char Set*, NCS–29

FORMAT secondary attribute • *File Applications*, 4–30

Form feed
- line printer • *I/O User's I*, 5–4
- mechanical • *I/O User's I*, 5–4
- terminal • *I/O User's I*, 8–21

FORTRAN
- See VAX FORTRAN
- /BLAS qualifier • *RTL Math*, 2–1
- example in • *RTL Parallel Processing*, 6–9
- special considerations • *RTL Parallel Processing*, 5–6

FORTRAN carriage control • *Convert*, CONV–2

FORTRAN carriage control option
- See FAB$V_FTN option

FORTRAN carriage control option list • *RMS*, 5–24

FORTRAN data type declaration • *Routines Intro*, A–31

FORTRAN implementation table • *Routines Intro*, A–31

Forward indexing • *RTL Math*, 2–6

FORWARD keyword • *VAXTPU*, 7–85, 7–379
- with SEARCH • *VAXTPU*, 7–328
- with SEARCH_QUIETLY • *VAXTPU*, 7–333

Found range selection
- in EVE editor • *VAXTPU*, 4–18

%FP • *Debugger,* 4–23, D–3
FPEMUL symbol • *System Dump Analyzer,* SDA–13
FP symbol • *System Dump Analyzer,* SDA–13
Frame
  call • *MACRO,* 9–64
  stack • *MACRO,* 9–64
Frame pointer • *System Dump Analyzer,* SDA–13
Free bucket list • *Convert,* CONV–4
Free cursor movement • *VAXTPU,* 7–95, 7–96
Free marker • *VAXTPU,* 2–9 to 2–10, 7–70
Free page list
  displaying • *System Dump Analyzer,* SDA–115
/FREE qualifier • *System Dump Analyzer,* SDA–115, SDA–118
Free queue
  See DR32 driver, FREEQ
Free service • *File Applications,* 8–5; *RMS,* RMS–45
  condition values • *RMS,* RMS–46
  control block input and output fields • *RMS,* RMS–46
FREE_CURSOR keyword
  with MARK • *VAXTPU,* 7–261
Full callable interface
  See VAXTPU routines
Full-checking synchronization image • *Device Support (A),* 13–28, E–17 to E–18
  loading • *Device Support (A),* E–2
Full duplex device driver • *Device Support (A),* 7–5; *Device Support (B),* 4–2
  I/O completion for • *Device Support (B),* 3–5
Full-duplex mode • *I/O User's I,* 8–10
Full image map • *Linker,* 1–12
Full map • *Linker,* 5–1, LINK–8
  module information in • *Linker,* 5–2, 5–3
  sections in • *Linker,* 5–2
  symbols cross-referenced in • *Linker,* LINK–5
FULL prompt • *File Def Language,* FDL–55
/FULL qualifier • *Debugger,* CD–236, CD–253; *Librarian,* LIB–23; *Linker,* LINK–8; *National Char Set,* NCS–30
  used with /LIST and /HISTORY qualifiers • *National Char Set,* NCS–31
  using with /HISTORY • *Librarian,* LIB–26
Full-reentrancy • *Modular Procedures,* 3–19
FUNCTAB macro • *Device Support (A),* 6–7; *Device Support (B),* 2–37 to 2–38
  example • *Device Support (B),* 2–38
Function
  definition of • *Routines Intro,* 2–3; *RTL Intro,* 1–1
Function code • *System Services Intro,* 7–12; *I/O User's II,* A–1 to A–6
  See also I/O function

Function code (Cont.)
  IO$_ACCESS • *I/O User's I,* 1–26
  IO$_ACPCONTROL • *I/O User's I,* 1–30, 6–15
  IO$_ADDSHAD • *I/O User's I,* 10–5
  IO$_AVAILABLE • *I/O User's I,* 3–33, 6–27, 10–8
  IO$_COPYSHAD • *I/O User's I,* 10–6
  IO$_CREATE • *I/O User's I,* 1–22
  IO$_CRESHAD • *I/O User's I,* 10–4
  IO$_DEACCESS • *I/O User's I,* 1–28
  IO$_DELETE • *I/O User's I,* 1–29
  IO$_DSE • *I/O User's I,* 6–27
  IO$_FORMAT • *I/O User's I,* 3–31
  IO$_INITIALIZE • *I/O User's I,* 4–9
  IO$_LOADMCODE • *I/O User's I,* 4–8; *I/O User's II,* 4–20
  IO$_MODIFY • *I/O User's I,* 1–28
  IO$_PACKACK • *I/O User's I,* 3–32
  IO$_READLBLK • *I/O User's I,* 2–6, 3–29, 6–17, 7–5, 8–26; *I/O User's II,* 1–5, 2–7, 3–13, 5–5, 6–17
  IO$_READPBLK • *I/O User's I,* 2–6, 3–29, 6–17, 7–5; *I/O User's II,* 1–5, 2–7, 3–13, 5–5, 6–17
  IO$_READPROMPT • *I/O User's I,* 8–26
  IO$_READVBLK • *I/O User's I,* 2–6, 3–29, 6–17, 7–5, 8–26; *I/O User's II,* 1–5, 2–7, 3–13, 5–5, 6–17
  IO$_REMSHAD • *I/O User's I,* 10–7
  IO$_REWIND • *I/O User's I,* 6–19
  IO$_REWINDOFF • *I/O User's I,* 6–21
  IO$_SEARCH • *I/O User's I,* 3–31
  IO$_SEEK • *I/O User's I,* 3–33
  IO$_SENSECHAR • *I/O User's I,* 3–31, 8–53, 10–8
  IO$_SENSEMODE • *I/O User's I,* 2–7, 3–31, 5–9, 6–22, 8–53; *I/O User's II,* 2–19, 5–10, 6–37
  IO$_SETCHAR • *I/O User's I,* 2–10, 5–9, 6–23, 8–38; *I/O User's II,* 1–7, 2–9, 3–13, 5–6, 6–21
  IO$_SETCLOCK • *I/O User's I,* 4–10
  IO$_SETMODE • *I/O User's I,* 2–8, 5–9, 6–23, 8–38; *I/O User's II,* 1–7, 2–9, 3–13, 5–6, 6–21
  IO$_SETPRFPTH • *I/O User's I,* 3–34
  IO$_SKIPFILE • *I/O User's I,* 6–19
  IO$_SKIPRECORD • *I/O User's I,* 6–20
  IO$_STARTDATA • *I/O User's I,* 4–11; *I/O User's II,* 4–4, 4–7, 4–20
  IO$_UNLOAD • *I/O User's I,* 3–32, 6–22
  IO$_WRITECHECK • *I/O User's I,* 3–33
  IO$_WRITELBLK • *I/O User's I,* 3–30, 5–5, 6–18, 7–6, 8–34; *I/O User's II,* 1–6, 2–8, 3–13, 5–5, 6–19
  IO$_WRITEOF • *I/O User's I,* 6–21

- Function code (Cont.)
  - IO$_WRITEPBLK • *I/O User's I*, 3–30, 5–5, 6–18, 7–6, 8–34; *I/O User's II*, 1–6, 2–8, 3–13, 5–5, 6–19
  - IO$_WRITEVBLK • *I/O User's I*, 3–30, 5–5, 6–18, 7–6, 8–34; *I/O User's II*, 1–6, 2–8, 3–13, 5–5, 6–19
  - list of • *I/O User's I*, A–1 to A–9
- Function decision table
  - See FDT
- Function key
  - control code • *VAXTPU*, 7–241
  - control sequence • *VAXTPU*, 7–241
- Function modifier • *System Services Intro*, 7–13; *I/O User's II*, A–1 to A–6
  - for DR11–W/DRV11–WA driver • *I/O User's II*, 3–11, 4–20
  - for asynchronous DDCMP driver • *I/O User's II*, 5–5
  - for DMC11/DMR11 driver • *I/O User's II*, 1–6
  - for DMP11/DMF32 driver • *I/O User's II*, 2–8
  - for Ethernet/802 driver • *I/O User's II*, 6–19
  - IO$M_ACCESS • *I/O User's I*, 1–23, 1–26, 6–13
  - IO$M_ATTNAST • *I/O User's II*, 1–8, 2–19, 3–14, 5–10, 6–36
  - IO$M_BINARY • *I/O User's I*, 2–6
  - IO$M_BRDCST • *I/O User's I*, 8–46, 8–55
  - IO$M_BREAKTHRU • *I/O User's I*, 8–10, 8–35
  - IO$M_CANCTRLO • *I/O User's I*, 8–5, 8–35
  - IO$M_CLR_COUNTS • *I/O User's II*, 2–20, 5–11
  - IO$M_CREATE • *I/O User's I*, 1–23, 1–26, 6–13
  - IO$M_CTRL • *I/O User's II*, 2–9, 2–18 to 2–20, 2–25, 5–6, 5–9 to 5–11, 6–22, 6–36, 6–37
  - IO$M_CTRLCAST • *I/O User's I*, 8–42
  - IO$M_CTRLYAST • *I/O User's I*, 8–5, 8–42
  - IO$M_CVTLOW • *I/O User's I*, 8–27
  - IO$M_CYCLE • *I/O User's II*, 3–5, 3–11
  - IO$M_DATACHECK • *I/O User's I*, 3–15, 3–29, 3–30, 6–8, 6–17, 6–18
  - IO$M_DATAPATH • *I/O User's II*, 3–15
  - IO$M_DELDATA • *I/O User's I*, 3–30
  - IO$M_DELETE • *I/O User's I*, 1–23, 1–30
  - IO$M_DMOUNT • *I/O User's I*, 1–31
  - IO$M_DSABLMBX • *I/O User's I*, 8–27; *I/O User's II*, 1–6
  - IO$M_ENABLMBX • *I/O User's I*, 8–35; *I/O User's II*, 1–6
  - IO$M_ERASE • *I/O User's I*, 3–27, 3–31, 6–18
  - IO$M_ESCAPE • *I/O User's I*, 8–7, 8–27
  - IO$M_EXTEND • *I/O User's I*, 8–27, 8–29
  - IO$M_HANGUP • *I/O User's I*, 8–42
  - IO$M_INCLUDE • *I/O User's I*, 8–43, 8–46
- Function modifier (Cont.)
  - IO$M_INHEXTGAP • *I/O User's I*, 6–10
  - IO$M_INHRETRY • *I/O User's I*, 3–29, 6–9
  - IO$M_MAINT • *I/O User's I*, 8–44, 8–45
  - IO$M_NOECHO • *I/O User's I*, 8–10, 8–24, 8–27
  - IO$M_NOFILTR • *I/O User's I*, 8–27
  - IO$M_NOFORMAT • *I/O User's I*, 8–11, 8–35
  - IO$M_NORSWAIT • *I/O User's I*, 7–7
  - IO$M_NOW • *I/O User's I*, 7–6, 7–7; *I/O User's II*, 1–6, 2–8, 5–5, 6–19
  - IO$M_NOWAIT • *I/O User's I*, 6–19, 6–21, 6–22
  - IO$M_OUTBAND • *I/O User's I*, 8–46
  - IO$M_PACKED • *I/O User's I*, 2–6
  - IO$M_PURGE • *I/O User's I*, 8–27
  - IO$M_RD_COUNTS • *I/O User's II*, 2–20, 5–11
  - IO$M_RD_MEM • *I/O User's II*, 2–25
  - IO$M_RD_MODEM • *I/O User's I*, 8–54; *I/O User's II*, 2–24
  - IO$M_READATTN • *I/O User's I*, 7–9
  - IO$M_REFRESH • *I/O User's I*, 8–36
  - IO$M_RESET • *I/O User's II*, 3–12
  - IO$M_RESPONSE • *I/O User's II*, 6–21
  - IO$M_REVERSE • *I/O User's I*, 6–17
  - IO$M_SETEVF • *I/O User's I*, 4–11; *I/O User's II*, 4–20, 4–22
  - IO$M_SETFNCT • *I/O User's II*, 3–5, 3–11
  - IO$M_SETPROT • *I/O User's I*, 7–11
  - IO$M_SET_MODEM • *I/O User's I*, 8–44; *I/O User's II*, 2–24
  - IO$M_SHUTDOWN • *I/O User's II*, 1–8, 2–18, 5–9, 6–36
  - IO$M_STARTUP • *I/O User's II*, 1–8, 2–9, 2–15, 5–6, 5–8, 6–22
  - IO$M_TIMED • *I/O User's I*, 8–27; *I/O User's II*, 3–11
  - IO$M_TRMNOECHO • *I/O User's I*, 8–28
  - IO$M_TT_ABORT • *I/O User's I*, 8–46
  - IO$M_TYPEAHDCNT • *I/O User's I*, 8–54
  - IO$M_UNLOOP • *I/O User's I*, 8–45
  - IO$M_WORD • *I/O User's II*, 3–11
  - list of • *I/O User's I*, A–1 to A–9
  - types of
    - IO$M_DATACHECK • *System Services Intro*, 7–13
    - IO$M_INHERLOG • *System Services Intro*, 7–7
    - IO$M_INHRETRY • *System Services Intro*, 7–13
- Function procedures • *VAXTPU*, 3–19
- Function return value • *RTL Intro*, 3–5; *RTL String Manipulation*, 2–6

Function return value (Cont.)
returned in output argument • *RTL String Manipulation*, 2–6
returned in R0/R1 • *RTL String Manipulation*, 2–6
Function value • *Routines Intro*, 2–7
registers • *Routines Intro*, 2–12
Function value returned
in registers • *Routines Intro*, 2–7
function_code data type • *Routines Intro*, A–7t
FWA (file work area) • *System Dump Analyzer*, SDA–77
F_floating data type • *MACRO*, 8–3, 9–102
.F_FLOATING directive • *MACRO*, 6–35

G

Gadget • *VAXTPU*, 2–25
GBD (global buffer descriptor) • *System Dump Analyzer*, SDA–77
GBD (global buffer descriptor) summary page • *System Dump Analyzer*, SDA–77
GBH (global buffer header) • *System Dump Analyzer*, SDA–77
GBLPAGES system parameter • *File Applications*, 1–16
GBLPAGFIL system parameter • *File Applications*, 1–16
GBLSECTIONS system parameter • *File Applications*, 1–16
GBSB (global buffer synchronization block) • *System Dump Analyzer*, SDA–77
;G command • *Delta/XDelta*, DELTA–33
General mode • *MACRO*, 5–15
General purpose registers
rules for using in driver code • *Device Support (A)*, 5–3
General register
See also Register
General register mode • *MACRO*, 5–1
summary • *MACRO*, 8–28
General register symbol • *Delta/XDelta*, DELTA–9, DELTA–13
/GENERATE qualifier • *Debugger*, CD–71
Generic key match • *File Applications*, 8–11
Generic SCSI class driver • *I/O User's I*, 11–1 to 11–16
assigning a channel to • *I/O User's I*, 11–10
flow of • *I/O User's I*, 11–4 to 11–6
I/O status block returned by • *I/O User's I*, 11–11
loading • *I/O User's I*, 11–9
Generic SCSI class driver (Cont.)
obtaining device information from • *I/O User's I*, 11–14
programming example • *I/O User's I*, 11–15 to 11–16
$QIO system service format for • *I/O User's I*, 11–11 to 11–14
security considerations • *I/O User's I*, 11–6
Generic SCSI descriptor
format of • *I/O User's I*, 11–12 to 11–14
Generic VAXBI device • *Device Support (A)*, 11–2, 16–1 to 16–30
See also VAXBI node
initialized by driver • *Device Support (A)*, 16–11 to 16–18
initialized by VMS • *Device Support (A)*, 16–7 to 16–11
interrupt destination • *Device Support (A)*, 16–10
Geometric model of performance • *RTL Parallel Processing*, 5–10 to 5–13
GET attribute • *File Def Language*, FDL–3, FDL–37
$GET macro
program example • *RMS*, 4–16
$GETMSG • *RTL Library*, 4–16
GET option • *File Def Language*, FDL–3, FDL–37
See also FAB$V_GET option
GET secondary attribute • *File Applications*, 7–4, 7–22
Get service • *File Applications*, 8–1, 8–2; *RMS*, RMS–47, RMS–53
and current record • *File Applications*, 8–15
applicable access modes • *RMS*, RMS–48
compared with Find service • *File Applications*, 8–2
condition values • *RMS*, RMS–53
See also Completion status code
control block input fields • *RMS*, RMS–50
control block output fields • *RMS*, RMS–53
effect on next-record position • *File Applications*, 8–16
high-level language equivalents • *File Applications*, 8–1
requirement for end-of-file test • *File Applications*, 8–3
requirement for user record area • *RMS*, RMS–50
returning terminator character for terminal input • *RMS*, RMS–49
return status for various file access methods • *RMS*, RMS–7
run-time options • *File Applications*, 9–14 to 9–17
using input from mailbox devices • *RMS*, RMS–50
using stream input • *RMS*, RMS–48
using terminal input • *RMS*, RMS–48

Get service (Cont.)
using the RAB$L_STV field for additional status information • *RMS*, RMS–50
Get sharing option
See FAB$V_GET option
GET_CLIPBOARD built-in procedure • *VAXTPU*, 7–149
example of use • *VAXTPU*, B–11 to B–13
GET_DEFAULT built-in procedure • *VAXTPU*, 7–151
GET_GLOBAL_SELECT built-in procedure • *VAXTPU*, 7–153
example of use • *VAXTPU*, B–13 to B–15
GET_INFO built-in procedure • *VAXTPU*, 7–156 to 7–161
buffer variable parameter
"read_routine" • *VAXTPU*, 7–174, 7–201
COMMAND_LINE keyword parameter
"line" • *VAXTPU*, 7–176, 7–177
key_name parameter
"key_modifiers" • *VAXTPU*, 7–162
marker_variable parameter
"record_number" • *VAXTPU*, 7–186
mouse_event_keyword parameter
"mouse_button" • *VAXTPU*, 7–188
"window" • *VAXTPU*, 7–188
SCREEN keyword parameter
"active_area" • *VAXTPU*, 7–196
"decwindows" • *VAXTPU*, 7–197
"event" • *VAXTPU*, 7–199
"global_select" • *VAXTPU*, 7–199
"grab_routine" • *VAXTPU*, 7–199
"icon_name" • *VAXTPU*, 7–199
"input_focus" • *VAXTPU*, 7–199
"length" • *VAXTPU*, 7–199
"new_length" • *VAXTPU*, 7–200
"new_width" • *VAXTPU*, 7–200
"old_length" • *VAXTPU*, 7–200
"old_width" • *VAXTPU*, 7–200
"original_length" • *VAXTPU*, 7–200
"read_routine" • *VAXTPU*, 7–201
"screen_limits" • *VAXTPU*, 7–201
"time" • *VAXTPU*, 7–202
"ungrab_routine" • *VAXTPU*, 7–202
string constant parameter
"active_area" • *VAXTPU*, 7–196
“Ansi_crt” • *VAXTPU*, 7–196
“auto_repeat” • *VAXTPU*, 7–196
“bell” • *VAXTPU*, 7–205
“beyond_eob” • *VAXTPU*, 7–185
“beyond_eol” • *VAXTPU*, 7–185, 7–220
“blink_status” • *VAXTPU*, 7–221
“blink_video” • *VAXTPU*, 7–221

GET_INFO built-in procedure
string constant parameter (Cont.)
“bold_status” • *VAXTPU*, 7–221
“bold_video” • *VAXTPU*, 7–221
"bottom" • *VAXTPU*, 7–222
“bound” • *VAXTPU*, 7–171, 7–185, 7–221
“breakpoint” • *VAXTPU*, 7–179
“buffer” • *VAXTPU*, 7–185, 7–193, 7–222
"callback_parameters" • *VAXTPU*, 7–209
"callback_routine" • *VAXTPU*, 7–214
“character” • *VAXTPU*, 7–171
“children” • *VAXTPU*, 7–210
“class” • *VAXTPU*, 7–214
“client_message” • *VAXTPU*, 7–197
“client_message_routine” • *VAXTPU*, 7–197
“column_move_vertical” • *VAXTPU*, 7–206
“command” • *VAXTPU*, 7–176
“command_file” • *VAXTPU*, 7–176
“create” • *VAXTPU*, 7–177
“cross_window_bounds” • *VAXTPU*, 7–197
“current” • *VAXTPU*, 7–166, 7–167, 7–169, 7–184, 7–191, 7–218
“current_column” • *VAXTPU*, 7–197, 7–222
“current_row” • *VAXTPU*, 7–197, 7–222
"decwindows" • *VAXTPU*, 7–197
“dec_crt2” • *VAXTPU*, 7–197
“dec_crt” • *VAXTPU*, 7–197
“default_directory” • *VAXTPU*, 7–206
“defined” • *VAXTPU*, 7–190
“detached_action” • *VAXTPU*, 7–197
“detached_reason” • *VAXTPU*, 7–198
“direction” • *VAXTPU*, 7–171
“display” • *VAXTPU*, 7–177, 7–206
“display_value” • *VAXTPU*, 7–186, 7–222
“edit_mode” • *VAXTPU*, 7–198
“eightbit” • *VAXTPU*, 7–198
"enable_resize" • *VAXTPU*, 7–206
“eob_text” • *VAXTPU*, 7–171
“erase_unmodifiable” • *VAXTPU*, 7–169, 7–171
"event" • *VAXTPU*, 7–199
“examine” • *VAXTPU*, 7–179
“facility_name” • *VAXTPU*, 7–206
“file_name” • *VAXTPU*, 7–171, 7–177
“find_buffer” • *VAXTPU*, 7–169
“first” • *VAXTPU*, 7–166, 7–167, 7–169, 7–181, 7–183, 7–184, 7–191, 7–218
“first_marker” • *VAXTPU*, 7–172
“first_range” • *VAXTPU*, 7–172
"global_select" • *VAXTPU*, 7–199
"grab_routine" • *VAXTPU*, 7–199
“high_index” • *VAXTPU*, 7–167

GET_INFO built-in procedure
string constant parameter (Cont.)
"icon_name" • *VAXTPU*, 7–199
"informational" • *VAXTPU*, 7–206
"initialization" • *VAXTPU*, 7–177
"initialization_file" • *VAXTPU*, 7–177
"init_file" • *VAXTPU*, 7–177
"input_focus" • *VAXTPU*, 7–199
"is_managed" • *VAXTPU*, 7–214
"is_subclass" • *VAXTPU*, 7–214
"journaling" • *VAXTPU*, 1–12, 5–10, 7–172
"journaling_frequency" • *VAXTPU*, 7–206
"journal" • *VAXTPU*, 7–177, 7–203
"journal_file" • *VAXTPU*, 1–12, 5–11, 7–172, 7–177, 7–206
"journal_name" • *VAXTPU*, 7–172
"key_map_list" • *VAXTPU*, 7–222
"key_map_list" • *VAXTPU*, 7–172
"key_modifiers" • *VAXTPU*, 7–162
"key_type" • *VAXTPU*, 7–162
"last" • *VAXTPU*, 7–166, 7–167, 7–169, 7–181, 7–183, 7–184, 7–191, 7–218
"left" • *VAXTPU*, 7–222
"left_margin" • *VAXTPU*, 7–172, 7–186
"left_margin_action" • *VAXTPU*, 7–172
"length" • *VAXTPU*, 7–199, 7–223
"line" • *VAXTPU*, 7–176, 7–177
"line" • *VAXTPU*, 7–172
"line_editing" • *VAXTPU*, 7–199
"line_number" • *VAXTPU*, 7–179, 7–206
"local" • *VAXTPU*, 7–179
"map_count" • *VAXTPU*, 7–173
"maximum_parameters" • *VAXTPU*, 7–190
"max_lines" • *VAXTPU*, 7–173
"menu_position" • *VAXTPU*, 7–210
"message_action_level" • *VAXTPU*, 7–206
"message_action_type" • *VAXTPU*, 7–206
"message_flags" • *VAXTPU*, 7–207
"middle_of_tab" • *VAXTPU*, 7–223
"minimum_parameters" • *VAXTPU*, 7–190
"mode" • *VAXTPU*, 7–173
"modifiable" • *VAXTPU*, 7–173
"modified" • *VAXTPU*, 7–173
"modify" • *VAXTPU*, 7–177
"mouse" • *VAXTPU*, 7–200
"mouse_button" • *VAXTPU*, 7–188
"name" • *VAXTPU*, 7–215
"name" • *VAXTPU*, 7–164, 7–173, 7–182
"new_length" • *VAXTPU*, 7–200
"new_width" • *VAXTPU*, 7–200

GET_INFO built-in procedure
string constant parameter (Cont.)
"next" • *VAXTPU*, 7–166, 7–168, 7–169, 7–180, 7–181, 7–183, 7–184, 7–191, 7–218, 7–223
"next_marker" • *VAXTPU*, 7–173
"next_range" • *VAXTPU*, 7–173
"nomodify" • *VAXTPU*, 7–177
"no_video" • *VAXTPU*, 7–223
"no_video_status" • *VAXTPU*, 7–223
"no_write" • *VAXTPU*, 7–174
"offset" • *VAXTPU*, 7–174, 7–186
"offset_column" • *VAXTPU*, 7–174, 7–186
"old_length" • *VAXTPU*, 7–200
"old_width" • *VAXTPU*, 7–200
"original_bottom" • *VAXTPU*, 7–223
"original_length" • *VAXTPU*, 7–200
"original_length" • *VAXTPU*, 7–223
"original_top" • *VAXTPU*, 7–223
"original_width" • *VAXTPU*, 7–200
"output" • *VAXTPU*, 7–177
"output_file" • *VAXTPU*, 7–174, 7–178
"pad" • *VAXTPU*, 7–223
"pad_overstruck_tabs" • *VAXTPU*, 7–207
"parameter" • *VAXTPU*, 7–180
"parent" • *VAXTPU*, 7–215
"permanent" • *VAXTPU*, 7–174
"pid" • *VAXTPU*, 7–192
"post_key_procedure" • *VAXTPU*, 7–204
"previous" • *VAXTPU*, 7–166, 7–168, 7–169, 7–180, 7–181, 7–183, 7–184, 7–191, 7–218, 7–223
"pre_key_procedure" • *VAXTPU*, 7–204
"procedure" • *VAXTPU*, 7–180
"prompt_length" • *VAXTPU*, 7–200
"prompt_row" • *VAXTPU*, 7–201
"read_only" • *VAXTPU*, 7–178
"read_routine" • *VAXTPU*, 7–174, 7–201
"record_count" • *VAXTPU*, 7–175
"record_number" • *VAXTPU*, 7–186
"record_number" • *VAXTPU*, 7–175
"record_size" • *VAXTPU*, 7–175
"recover" • *VAXTPU*, 7–207
"recover" • *VAXTPU*, 7–178
"resize_action" • *VAXTPU*, 7–207
"resources" • *VAXTPU*, 7–215
"reverse_status" • *VAXTPU*, 7–224
"reverse_video" • *VAXTPU*, 7–224
"right" • *VAXTPU*, 7–224
"right_margin" • *VAXTPU*, 7–175, 7–186
"right_margin_action" • *VAXTPU*, 7–175
"safe_for_journaling" • *VAXTPU*, 7–175

GET_INFO built-in procedure
  string constant parameter (Cont.)
    "screen_limits" • *VAXTPU*, 7–201
    “screen_update” • *VAXTPU*, 7–201
    “scroll” • *VAXTPU*, 7–201, 7–224
    “scroll_amount” • *VAXTPU*, 7–224
    "scroll_bar" • *VAXTPU*, 7–224
    "scroll_bar_auto_thumb" • *VAXTPU*, 7–224
    “scroll_bottom” • *VAXTPU*, 7–224
    “scroll_top” • *VAXTPU*, 7–225
    “section” • *VAXTPU*, 7–178
    “section_file” • *VAXTPU*, 7–178, 7–207
    “self_insert” • *VAXTPU*, 7–204
    “shift_amount” • *VAXTPU*, 7–225
    “shift_key” • *VAXTPU*, 7–204, 7–207
    “special_graphics_status” • *VAXTPU*, 7–225
    “start_character” • *VAXTPU*, 7–178
    “start_record” • *VAXTPU*, 7–178
    “status_line” • *VAXTPU*, 7–225
    “status_video” • *VAXTPU*, 7–225
    “success” • *VAXTPU*, 7–207
    “system” • *VAXTPU*, 7–175
    “tab_stops” • *VAXTPU*, 7–175
    "text" • *VAXTPU*, 7–215
    “text” • *VAXTPU*, 7–225
    "time" • *VAXTPU*, 7–202
    “timed_message” • *VAXTPU*, 7–207
    "timer" • *VAXTPU*, 7–207
    "top" • *VAXTPU*, 7–225
    “traceback” • *VAXTPU*, 7–207
    “type” • *VAXTPU*, 7–165
    “undefined_key” • *VAXTPU*, 7–204
    “underline_status” • *VAXTPU*, 7–225
    “underline_video” • *VAXTPU*, 7–225
    "ungrab_routine" • *VAXTPU*, 7–202
    “unmodifiable_records” • *VAXTPU*, 7–175, 7–186, 7–193
    “update” • *VAXTPU*, 7–208
    “version” • *VAXTPU*, 7–208
    “video” • *VAXTPU*, 7–187, 7–193, 7–226
    “visible” • *VAXTPU*, 7–226
    “visible_bottom” • *VAXTPU*, 7–226
    “visible_length” • *VAXTPU*, 7–202, 7–226
    “visible_top” • *VAXTPU*, 7–226
    “vk100” • *VAXTPU*, 7–202
    “vt100” • *VAXTPU*, 7–202
    “vt200” • *VAXTPU*, 7–202
    “vt300” • *VAXTPU*, 7–202
    "widget_id" • *VAXTPU*, 7–209
    "widget_info" • *VAXTPU*, 7–216
    "width" • *VAXTPU*, 7–226
    “width” • *VAXTPU*, 7–202

GET_INFO built-in procedure
  string constant parameter (Cont.)
    "window" • *VAXTPU*, 7–188
    “within_range” • *VAXTPU*, 7–187
    “write” • *VAXTPU*, 7–178
  SYSTEM keyword parameter
    "enable_resize" • *VAXTPU*, 7–206
    "recover" • *VAXTPU*, 7–207
    "resize_action" • *VAXTPU*, 7–207
    "timer" • *VAXTPU*, 7–207
  WIDGET keyword parameter
    "callback_parameters" • *VAXTPU*, 4–11, 7–209
    "widget_id" • *VAXTPU*, 7–209
  widget variable parameter
    "name" • *VAXTPU*, 7–215
    "text" • *VAXTPU*, 7–215
    "widget_info" • *VAXTPU*, 7–216
  widget_variable parameter
    "callback_routine" • *VAXTPU*, 7–214
  window variable parameter
    "left" • *VAXTPU*, 7–222
    "length" • *VAXTPU*, 7–223
    "right" • *VAXTPU*, 7–224
    "scroll_bar" • *VAXTPU*, 7–224
    "scroll_bar_auto_thumb" • *VAXTPU*, 7–224
    "top" • *VAXTPU*, 7–225
    "width" • *VAXTPU*, 7–226
  window_variable parameter
    "bottom" • *VAXTPU*, 7–222
    example of use • *VAXTPU*, B–16 to B–19, B–19 to B–22
    "key_map_list" • *VAXTPU*, 7–222
Givens plane rotation
  applying to a vector • *RTL Math*, MTH–173
  generating the elements for • *RTL Math*, MTH–178
Global buffer • *File Applications*, 1–16, 3–8, 3–27; *File Def Language*, FDL–20; *RMS*, 5–19
  determining number of • *RMS*, 5–20
  number • *File Applications*, 7–17
  performance • *File Applications*, 9–9
  restricted use • *File Applications*, 7–21
  with deferred-write option • *File Applications*, 3–9
  with indexed file • *File Applications*, 7–21
  with relative file • *File Applications*, 7–21
  with shared file • *File Applications*, 7–20 to 7–22
  with shared sequential file • *File Applications*, 3–12
Global buffer count
  example of run-time specification • *File Applications*, 5–10 to 5–12
Global buffer count field
  See FAB$W_GBC field

- Global buffer descriptor
  - See GBD
- Global buffer header
  - See GBH
- Global buffer synchronization block
  - See GBSB
- GLOBAL clause
  - for PLACEMENT clause • *Command Def*, CDU–25, CDU–34
- .GLOBAL directive • *MACRO*, 6–37
- Global expression • *MACRO*, 3–9
- Global label • *MACRO*, 2–2
  - use with NCS routines • *National Char Set*, NCS–36
- Global page-file section • *File Applications*, 1–16
- Global page table • *File Applications*, 1–16
  - displaying • *System Dump Analyzer*, SDA–111
- /GLOBAL qualifier • *System Dump Analyzer*, SDA–111
- /GLOBALS-/NOGLOBALS qualifier
  - with DELETE command • *Patch*, PAT–53
  - with DEPOSIT command • *Patch*, PAT–56
  - with EXAMINE command • *Patch*, PAT–63
  - with INSERT command • *Patch*, PAT–68
  - with REPLACE command • *Patch*, PAT–72
  - with SET MODE command • *Patch*, PAT–77
  - with VERIFY command • *Patch*, PAT–91
- Global section • *Programming Resources*, 5–15; *Routines Intro*, A–12t; *System Services Intro*, 11–11; *RTL Parallel Processing*, 3–1; *File Applications*, 1–16
  - characteristic • *System Services Intro*, 11–11
  - controlling access through access control lists • *Utility Routines*, ACL–1
  - creating • *System Services*, SYS–105
  - defining • *System Services Intro*, 11–8
  - deleting • *System Services*, SYS–140
  - for interprocess communication • *System Services Intro*, 8–11
  - linker-assigned name of • *Linker*, 5–6
  - mapping • *System Services Intro*, 11–15; *System Services*, SYS–105, SYS–339
  - multiprocessing • *Programming Resources*, 4–18
  - name • *System Services Intro*, 11–12
  - paging file • *System Services Intro*, 11–15
  - permanent • *Programming Resources*, 5–19
  - processing of by image activator • *Linker*, 4–12
  - temporary • *Programming Resources*, 5–19
  - writable • *Programming Resources*, 4–18
- Global section watchpoint • *Debugger*, 10–17
- Global selection
  - determining ownership of • *VAXTPU*, 7–199
  - fetching grab routine for • *VAXTPU*, 7–199
  - fetching information about • *VAXTPU*, 7–153
  - fetching read request for • *VAXTPU*, 7–199
  - fetching read routine for • *VAXTPU*, 7–174, 7–201
  - fetching ungrab routine for • *VAXTPU*, 7–202
  - fetching wait time for • *VAXTPU*, 7–202
  - obtaining data from • *VAXTPU*, 7–300
  - reading information about • *VAXTPU*, 7–299
  - requesting ownership of • *VAXTPU*, 7–380
  - sending information about to an application • *VAXTPU*, 7–546
  - specifying expiration period for • *VAXTPU*, 7–387
  - specifying grab routine for • *VAXTPU*, 7–382
  - specifying read routine for • *VAXTPU*, 7–385
  - specifying ungrab routine for • *VAXTPU*, 7–389
  - support for • *VAXTPU*, 4–6 to 4–8
- GLOBALS-NOGLOBALS mode • *Patch*, PAT–17
- /GLOBALS qualifier • *Librarian*, LIB–24
- Global symbol • *Programming Resources*, 5–11; *Linker*, 2–8; *Patch*, PAT–7; *MACRO*, 3–6, 6–101
  - See also Message symbol
  - See Symbol
  - absolute • *Linker*, 1–9, 3–11
  - attribute directive (.GLOBAL) • *MACRO*, 6–37
  - conversion of to universal • *Linker*, 3–12
  - defining • *MACRO*, 6–22, 6–34, 6–37
  - defining by option • *Linker*, 1–9, 3–11
  - defining for shareable image • *MACRO*, 6–96
  - designation of • *Linker*, 2–8
  - resolving • *Programming Resources*, 5–11
  - signaling with • *Programming Resources*, 9–11
  - strong definition of • *Linker*, 2–10
  - strong reference to • *Linker*, 2–9
  - weak definition of • *Linker*, 2–10
  - weak reference to • *Linker*, 2–10
- Global symbol table
  - See GST
- Global variable • *VAXTPU*, 3–4
- /GLOBAL_BUFFERS qualifier • *File Applications*, 7–22
- GLOBAL_BUFFER_COUNT attribute • *File Def Language*, FDL–20
- GLOBAL_BUFFER_COUNT secondary attribute • *File Applications*, 7–17, 7–22
- Go button
  - with DECwindows • *Debugger*, 1–9
- GO command • *Debugger*, 2–13, CD–105; *Delta/XDelta*, DELTA–33
  - multiprocess program • *Debugger*, 10–6

GO command (Cont.)
with DECwindows • *Debugger*, 1–24
GOLD key
restriction on defining in EVE • *VAXTPU*, 7–472
G operator • *System Dump Analyzer*, SDA–12
Grab routine
fetching event in • *VAXTPU*, 7–199
global selection
fetching • *VAXTPU*, 7–199
specifying • *VAXTPU*, 7–382
input focus • *VAXTPU*, 7–398
fetching • *VAXTPU*, 7–199
specifying • *VAXTPU*, 7–400
Granularity • *RTL Parallel Processing*, 5–1
in lock • *System Services Intro*, 12–2
/GRANULARITY qualifier • *File Def Language*, FDL–42, FDL–51
GRAPHIC_TABS keyword • *VAXTPU*, 7–483
Group logical name table • *System Services Intro*, 6–6
Group number • *File Def Language*, FDL–22
GROUP protection code • *File Def Language*, FDL–23
GSMATCH option • *Programming Resources*, 5–6
See also Linker Utility
GSMATCH processing • *Linker*, 3–8
GST (global symbol table) • *Librarian*, LIB–2; *Linker*, 1–6, 2–7, 6–13
building of in Pass 1 • *Linker*, 6–11
creating • *Debugger*, 5–4
limiting symbols in • *Linker*, LINK–29
shareable image • *Debugger*, 5–13
G symbol • *Delta/XDelta*, DELTA–9; *System Dump Analyzer*, SDA–14
G_floating data type • *MACRO*, 8–4, 9–102
.G_FLOATING directive • *MACRO*, 6–36
/G_FLOAT qualifier • *Debugger*, CD–62, CD–86

H

Half-duplex mode • *I/O User's I*, 8–10, 8–21
See also Duplex mode
HALT (Halt) instruction • *MACRO*, 9–74, 10–43
interrupt stack not valid • *MACRO*, E–10
synchronizing vector memory before • *MACRO*, 10–43
Handler
change and compatibility mode • *System Services Intro*, 10–6
condition • *Debugger*, 9–13
Hang up
function modifier • *I/O User's I*, 8–42
terminal • *I/O User's I*, 8–18, 8–24
Hardcopy terminal output • *File Def Language*, FDL–55
Hard-positioning option • *File Applications*, 4–31
Hardware clock
See Interval clock
Hardware error • *File Applications*, 10–1
vector • *MACRO*, 10–31, 10–47
HDR1 labels
accessing from XAB$B_MTACC field • *RMS*, 14–5
Header
crash dump • *System Dump Analyzer*, SDA–106
library • *Programming Resources*, 8–50
library module • *Programming Resources*, 8–48
/HEADER qualifier • *Linker*, LINK–10; *SUMSLP*, SUM–18; *System Dump Analyzer*, SDA–118
Heap storage • *RTL String Manipulation*, 2–3
HEIGHT parameter to SET built-in procedure • *VAXTPU*, 7–391
Help
online • *Debugger*, 2–8, CD–107
for debugger messages • *Debugger*, 2–8, CD–5
with DECwindows • *Debugger*, 1–19
HELP command • *Debugger*, 2–8, CD–107; *Patch*, PAT–67; *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–28; *System Dump Analyzer*, SDA–58
EDIT/FDL • *File Def Language*, FDL–62
recording output • *System Dump Analyzer*, SDA–71
Help files
comment lines in • *Librarian*, LIB–6
creating • *Librarian*, LIB–4 to LIB–5
formatting • *Librarian*, LIB–5
qualifier lines in • *Librarian*, LIB–6
restrictions in • *Librarian*, LIB–4
Help library • *Programming Resources*, 1–18; *Librarian*, LIB–1, LIB–4
character case in • *Librarian*, LIB–2
displaying text • *Programming Resources*, 8–52
index keywords in • *Librarian*, LIB–4
key names in • *Librarian*, LIB–4 to LIB–5
HELP LIBRARY command display • *Librarian*, LIB–8 to LIB–10
/HELP qualifier • *Librarian*, LIB–25
Help text
example of • *Librarian*, LIB–6 to LIB–8
retrieving • *Librarian*, LIB–8 to LIB–10

HELP_TEXT built-in procedure • *VAXTPU*, 7–228 to 7–229
%HEX • *Debugger*, 4–12, D–5
Hexadecimal/decimal conversion • *MACRO*, B–1
  table • *MACRO*, B–1
Hexadecimal dump • *Analyze/RMS_File*, ARMS–25
HEXADECIMAL mode • *Patch*, PAT–17
/HEXADECIMAL qualifier • *Debugger*, 4–12, CD–81, CD–83, CD–87
/HEXADECIMAL qualifier
  with DELETE command • *Patch*, PAT–53
  with DEPOSIT command • *Patch*, PAT–56
  with EVALUATE command • *Patch*, PAT–59
  with EXAMINE command • *Patch*, PAT–63
  with INSERT command • *Patch*, PAT–68
  with REPLACE command • *Patch*, PAT–72
  with SET MODE command • *Patch*, PAT–76
  with VERIFY command • *Patch*, PAT–91
Hexadecimal text
  converting to binary • *RTL Library*, LIB–76
Hexadecimal value of an expression • *System Dump Analyzer*, SDA–48
Hibernation • *System Services Intro*, 8–11
  alternate method • *System Services Intro*, 8–13
  and AST • *System Services Intro*, 5–3
  compared with suspension • *System Services Intro*, 8–11
  LIB$WAIT • *RTL Library*, LIB–465
HIBER system service
  use of • *RTL Parallel Processing*, 5–5
/HIDE qualifier • *Debugger*, CD–71
Hierarchical structure • *Analyze/RMS_File*, ARMS–1
Highest virtual block field
  See XAB$L_HBK field
High-level language
  argument evaluation • *Routines Intro*, 2–6
  argument transmission • *Routines Intro*, 2–6
  call from • *System Services Intro*, 2–17
  mapped into argument lists • *Routines Intro*, 2–6
High-speed terminal output • *File Def Language*, FDL–55
“High_index” string constant parameter to GET_ INFO • *VAXTPU*, 7–167
/HISTORY qualifier • *Librarian*, LIB–26
  used to limit listing output • *National Char Set*, NCS–31
Holder record • *System Services Intro*, 3–5
  adding • *System Services Intro*, 3–9
  format of • *System Services Intro*, 3–5
  modifying • *System Services Intro*, 3–13
  removing • *System Services Intro*, 3–15
/HOLD qualifier • *Debugger*, 10–3, 10–7, CD–162, CD–182, CD–236, CD–253
Home block • *File Applications*, 1–7
H operator • *System Dump Analyzer*, SDA–12
Host • *System Services*, SYS–206
HRD option • *File Def Language*, FDL–7
HSC40 disk controller • *I/O User's I*, 3–3
HSC50 disk controller • *I/O User's I*, 3–3
HSC70 disk controller • *I/O User's I*, 3–3
HSC disk • *I/O User's I*, 3–15
H symbol • *Delta/XDelta*, DELTA–9; *System Dump Analyzer*, SDA–14
HWCLK spin lock • *Device Support (A)*, 3–8, 3–9, 3–14, E–13, E–15; *Device Support (B)*, 3–29, 3–48
Hyperbolic arc tangent • *RTL Math*, MTH–21, MTH–84
Hyperbolic cosine • *RTL Math*, MTH–51, MTH–88
Hyperbolic sine • *RTL Math*, MTH–100, MTH–133
Hyperbolic tangent • *RTL Math*, MTH–108, MTH–143
Hyphen (-)
  line-continuation character • *Debugger*, CD–4
  subtraction operator • *Debugger*, D–7
H_floating data type • *MACRO*, 8–5
.H_FLOATING directive • *MACRO*, 6–38
H_floating-point storage directive (.H_FLOATING) • *MACRO*, 6–38
/H_FLOAT qualifier • *Debugger*, CD–63, CD–87

I

I/O • *Modular Procedures*, 2–16, A–4
  See also Input/output
  asynchronous • *Modular Procedures*, 3–25
  at AST level • *Modular Procedures*, 3–25
  file • *Modular Procedures*, 2–18
  synchronous • *Modular Procedures*, 3–25
I/O adapter • *Device Support (A)*, 1–6, 1–10 to 1–16, 1–22
  See also MBA
  See also Q22 bus
  See also UNIBUS adapter
  configuration register • *Device Support (B)*, 1–6
  data path register • *Device Support (B)*, 2–51
  displaying nexus value • *Device Support (A)*, 12–8, 12–11
  number of address bits • *Device Support (B)*, 1–8, 2–3
  on VAXBI bus • *Device Support (A)*, 16–2

I/O adapter (Cont.)
- type • *Device Support (A),* 16–9; *Device Support (B),* 1–7, 1–33, 2–3, 2–21

I/O adapter registers
- See Byte count register
- See Data path register
- See Map registers
- See MBA
- See Vector register

I/O address space • *Device Support (A),* 19–1 to 19–7
- access to during bus power failure • *Device Support (A),* 19–7
- error in mapping • *Device Support (A),* 19–7
- mapping to process address space • *Device Support (A),* 19–4, 19–5 to 19–7, 19–8
- of SCU/XMI bus • *Device Support (A),* 16–5
- of VAXBI bus • *Device Support (A),* 16–2
- rules for referencing • *Device Support (A),* 19–7

I/O and performance • *File Applications,* 3–1

I/O buffers
- pseudoterminal • *I/O User's I,* 9–4

I/O channel • *System Services Intro,* 7–14
- See Process I/O channel
- assigning • *System Services,* SYS–23
- deassigning • *System Services Intro,* 7–20; *System Services,* SYS–119
- index • *Routines Intro,* A–2t

I/O completion
- See I/O postprocessing
- recommended test • *System Services Intro,* 7–17
- status • *System Services Intro,* 7–19
- synchronizing • *System Services Intro,* 7–15

I/O counts • *Convert,* CONV–24

I/O database • *Device Support (A),* 1–4 to 1–7; *Device Support (B),* 1–1, 1–2
- creation • *Device Support (A),* 6–1, 6–3, 11–4, 12–3 to 12–7, 12–14, 15–7; *Device Support (B),* 1–33, 2–25
- displaying SDA information • *System Dump Analyzer,* SDA–98
- examining with XDELTA • *Device Support (A),* 13–10
- for MASSBUS configuration • *Device Support (A),* 15–7 to 15–8, 15–13
- for two-controller configuration • *Device Support (A),* 4–7
- global symbols • *System Dump Analyzer,* SDA–60
- initializing • *Device Support (A),* 11–4, 12–14
- locating • *Device Support (A),* 12–12
- referencing fields in • *Device Support (A),* 5–2

I/O database (Cont.)
- reinitializing • *Device Support (A),* 11–4

I/O device
- getting information about
  - asynchronously • *System Services,* SYS–203
  - synchronously • *System Services,* SYS–221

I/O driver
- card reader • *I/O User's I,* 2–1
- disk • *I/O User's I,* 3–1
- DMC11/DMR11 • *I/O User's II,* 1–1
- DR11–W/DRV11–WA • *I/O User's II,* 3–1
- DR32 • *I/O User's II,* 4–1
- Ethernet/802 drivers • *I/O User's II,* 6–1
- line printer • *I/O User's I,* 5–1
- magnetic tape • *I/O User's I,* 6–1
- mailbox • *I/O User's I,* 7–1

I/O function
- See also Function code
- See also Function modifier
- ACP-QIO interface • *I/O User's I,* 1–2
- analyzing • *Device Support (A),* 8–2
- arguments • *I/O User's II,* A–1 to A–6
- card reader • *I/O User's I,* 2–5
- codes • *System Services Intro,* 7–12, 7–14; *I/O User's I,* A–1; *I/O User's II,* A–1 to A–6
- disk • *I/O User's I,* 1–2, 3–24
- for DR11–W/DRV11–WA driver • *I/O User's II,* 3–9
- for asynchronous DDCMP driver • *I/O User's II,* 5–4
- for DMC11/DMR11 driver • *I/O User's II,* 1–5
- for DMP11/DMF32 driver • *I/O User's II,* 2–6
- for DR32 driver • *I/O User's II,* 4–20
- for Ethernet/802 driver • *I/O User's II,* 6–16
- indicating a buffered • *Device Support (A),* 4–11, 6–4
- indicating as legal to a device • *Device Support (A),* 4–11, 6–4
- line printer • *I/O User's I,* 5–5
- list of • *I/O User's I,* A–1 to A–9
- LPA11-K device • *I/O User's I,* 4–8
- magnetic tape • *I/O User's I,* 1–2, 6–13
- mailbox • *I/O User's I,* 7–5
- modifiers • *System Services Intro,* 7–13; *I/O User's II,* A–1 to A–6
- preprocessing • *Device Support (A),* 4–12
- terminal • *I/O User's I,* 8–26

I/O function code • *Device Support (A),* 4–11; *Device Support (B),* 1–39
- converting to device-specific function code • *Device Support (A),* 8–4
- defined by VMS • *Device Support (A),* 6–5 to 6–7
- defining device-specific • *Device Support (A),* 6–8

I/O function modifier • *Device Support (A),* 4–11
I/O mode
  how to switch for sequential files • *RMS,* 4–24
  procedure for delaying decision until stream connection • *RMS,* 4–24
  when mode switching allowed • *RMS,* 4–24
I/O operation
  logical • *System Services Intro,* 7–7
  physical • *System Services Intro,* 7–7
  quotas, privileges, and protection • *System Services Intro,* 7–2
  summary of • *System Services Intro,* 7–6
  virtual • *System Services Intro,* 7–8
I/O postprocessing • *Device Support (A),* 3–5, 10–1 to 10–4; *Device Support (B),* 1–41
  device-dependent • *Device Support (A),* 2–7, 4–19 to 4–20, 7–8, 10–2 to 10–4
  device-independent • *Device Support (A),* 2–7, 4–20, 7–8; *Device Support (B),* 3–72 to 3–73
  for aborted I/O request • *Device Support (B),* 3–10
  for buffered I/O • *Device Support (A),* 7–8, 14–25
  for DMA transfer • *Device Support (A),* 14–16, 14–24 to 14–26
  for full duplex device driver • *Device Support (B),* 3–5
  for I/O request involving no device activity • *Device Support (B),* 3–24 to 3–25
  synchronization flow • *Device Support (A),* 3–4
I/O postprocessing queue • *Device Support (A),* 10–3, 11–7, E–14; *Device Support (B),* 1–17, 1–79, 3–5, 3–95
I/O preprocessing
  See also FDT routine
  See also SYS$QIO
  completing • *Device Support (A),* 4–13, 6–4
  device-dependent • *Device Support (A),* 2–3 to 2–4, 4–10 to 4–13, 7–1 to 7–9
  device-independent • *Device Support (A),* 2–3, 4–4 to 4–10
  IPL requirements • *Device Support (A),* 3–4
I/O request
  aborting • *Device Support (A),* 7–5, 10–6; *Device Support (B),* 3–10 to 3–11
  as serviced by SCSI class and port drivers • *Device Support (A),* 17–22 to 17–24
  canceling • *System Services Intro,* 7–22; *Device Support (A),* 11–6 to 11–9; *Device Support (B),* 1–30, 1–78, 3–68
  canceling on channel • *System Services,* SYS–39
  completing • *Device Support (B),* 3–94 to 3–95
  example • *Device Support (A),* 2–1 to 2–7
I/O request (Cont.)
  outstanding on channel • *Device Support (B),* 1–12
  queuing • *System Services Intro,* 7–14
    asynchronously • *System Services,* SYS–379
    synchronously • *System Services,* SYS–384
  restarting after power failure • *Device Support (A),* 8–5
  retrying • *Device Support (A),* 10–5 to 10–6
  returning completion status of to process • *Device Support (A),* 2–7, 4–20, 7–4, 10–2, 10–3
  status • *Device Support (B),* 1–40
  synchronizing simultaneous processing of multiple • *Device Support (A),* 7–5
  validating device-dependent arguments • *Device Support (A),* 2–3
  validating device-independent arguments • *Device Support (A),* 2–2 to 2–3, 4–8 to 4–9
  with no parameters • *Device Support (A),* 7–9; *Device Support (B),* 3–62
  with one parameter • *Device Support (A),* 7–9; *Device Support (B),* 3–37
I/O request packet
  See IRP
I/O segment • *Linker,* 1–6, 2–11
I/O service
  synchronous version • *System Services Intro,* 7–18
I/O space
  of MASSBUS • *Device Support (A),* 15–4
  of Q22 bus • *Device Support (A),* 14–4
  of UNIBUS • *Device Support (A),* 14–4
  rules for referencing • *Device Support (A),* 5–3, 5–5
  writing to • *Device Support (A),* 5–4
I/O space references
  vector • *MACRO,* 10–29, 10–42, 10–43, 10–47
I/O status block
  See IOSB
  ACP-QIO interface • *I/O User's I,* 1–35
  card reader • *I/O User's I,* 2–11
  disk • *I/O User's I,* 3–36
  LAT port driver • *I/O User's I,* 8–56
  line printer • *I/O User's I,* 5–10
  LPA11-K • *I/O User's I,* 4–33
  LPA11-K device • *I/O User's I,* 4–33
  magnetic tape • *I/O User's I,* 6–28
  mailbox • *I/O User's I,* 7–12
  returned by generic SCSI class driver • *I/O User's I,* 11–11
  terminal • *I/O User's I,* 8–56
I/O status block (IOSB)
  See IOSB
I/O unit • *File Applications,* 3–6, 3–7, 3–11

- IAN (index bucket area number)
  - program example • *RMS,* 4–8
- IAS • *File Def Language,* FDL–38
- ICCS register
  - displaying • *System Dump Analyzer,* SDA–90
- Icon
  - fetching text of • *VAXTPU,* 7–199
  - implementing in DECwindows VAXTPU • *VAXTPU,* 7–393, 7–395
  - specifying text for • *VAXTPU,* 7–392
- ICONIFY_PIXMAP parameter to SET built-in • *VAXTPU,* 7–395
- ICON_PIXMAP parameter to SET built-in • *VAXTPU,* 7–393
- IDB$L_ADP • *Device Support (A),* 4–7
- IDB$L_CSR • *Device Support (A),* 4–7, 15–4, 15–5, 15–13, 16–9
- IDB$L_OWNER • *Device Support (A),* 3–26, 4–6, 4–7, 8–4, 8–7, 9–3, 11–2; *Device Support (B),* 3–86, 3–100
- IDB$V_NO_CSR • *Device Support (B),* 1–36
- IDB$W_UNITS • *Device Support (A),* 12–6, 16–9
- IDB (interrupt dispatch block) • *System Dump Analyzer,* SDA–99; *Device Support (A),* 1–6, 4–7 to 4–8, 14–23; *Device Support (B),* 1–35 to 1–37
  - address • *Device Support (A),* 4–6, 8–4, 14–30, 14–32
  - creation • *Device Support (A),* 12–4; *Device Support (B),* 2–22
  - for generic VAXBI device • *Device Support (A),* 16–9
  - for MBA • *Device Support (A),* 15–4, 15–7 to 15–8, 15–13, 15–15
  - size • *Device Support (B),* 2–22
- IDENT attribute • *File Def Language,* FDL–2, FDL–39
- .IDENT directive • *MACRO,* 6–39
- Identification directive (.IDENT) • *MACRO,* 6–39
  - in message source file • *Message,* MSG–20
- /IDENTIFICATION qualifier
  - in message definition • *Message,* MSG–22
- Identifier • *System Services Intro,* 3–2; *VAXTPU,* 3–4
  - adding to rights database • *System Services Intro,* 3–9
  - attributes • *System Services Intro,* 3–5
  - defining • *System Services Intro,* 3–2
  - description • *Programming Resources,* 6–1
  - determining holders of • *System Services Intro,* 3–10
  - format of • *System Services Intro,* 3–2, 3–3
  - general • *System Services Intro,* 3–4
  - global section • *Routines Intro,* A–12t

Identifier (Cont.)

  - removing from rights database • *System Services Intro,* 3–15
  - rights database • *Routines Intro,* A–12t
  - search string • *Debugger,* 6–7
  - sharing • *RTL Parallel Processing,* 5–9
  - system-defined • *System Services Intro,* 3–4
  - UIC format • *System Services Intro,* 3–3
  - user • *Routines Intro,* A–11t, A–12t
- Identifier ACE • *System Services Intro,* 3–23
- identifier data type • *Routines Intro,* A–7t
- Identifier name • *System Services Intro,* 3–3
  - translating • *System Services Intro,* 3–8
- /IDENTIFIER qualifier • *Debugger,* 6–7, CD–121
- Identifier record • *System Services Intro,* 3–5
  - adding to rights database • *System Services Intro,* 3–9
  - format of • *System Services Intro,* 3–5
  - modifying • *System Services Intro,* 3–12
  - removing from rights database • *System Services Intro,* 3–15
- Identifier value
  - translating • *System Services Intro,* 3–8
- IDENT keyword
  - using to identify conversion function • *National Char Set,* NCS–14, NCS–16
- Ident produced by EVE$BUILD • *VAXTPU,* G–2
- IDENT statement • *Command Def,* CDU–14, CDU–36; *VAXTPU,* 3–14 to 3–15
- IDX (index descriptor) • *System Dump Analyzer,* SDA–77
- IDX_NCMPR option • *File Def Language,* FDL–28
- IFAB (internal file access block) • *System Dump Analyzer,* SDA–77
- IF command • *Debugger,* 8–10, CD–109
- %IFDEF lexical keyword • *VAXTPU,* 3–36
- .IF directive • *MACRO,* 6–40
- IFI (internal file identifier) • *System Dump Analyzer,* SDA–76
  - removing • *System Services Intro,* 6–11
- IFL (index bucket fill size)
  - program example • *RMS,* 4–8
- %IF lexical keyword • *VAXTPU,* 3–36
- IFNORD macro • *Device Support (B),* 2–39 to 2–40
- IFNOWRT macro • *Device Support (B),* 2–39 to 2–40
- IFRD macro • *Device Support (B),* 2–39 to 2–40
  - example • *Device Support (B),* 2–40
- If state • *RTL Screen Management,* 3–3
  - composed input • *Programming Resources,* 7–28
- IF statement • *VAXTPU,* 3–22 to 3–23
- IFWRT macro • *Device Support (B),* 2–39 to 2–40
- .IF_FALSE directive • *MACRO,* 6–43

/IF_STATE qualifier • *Debugger,* 8–9, CD–51; *System Dump Analyzer,* SDA–44

.IF_TRUE directive • *MACRO,* 6–43

..IF_TRUE_FALSE directive • *MACRO,* 6–43

.IIF directive • *MACRO,* 6–46

ILLQBUSCFG bugcheck • *Device Support (B),* 1–22

Image
- See also Shareable image
- base address of, in map • *Linker,* 5–8
- compression of • *Utility Routines,* DCX–1
- exiting • *Programming Resources,* 9–26; *System Services Intro,* 8–14
- for subprocess • *System Services Intro,* 8–3
- length of, in map • *Linker,* 5–8
- privileged • *Programming Resources,* 6–2
- privileged, securing • *Debugger,* 5–6
- rundown activity • *System Services Intro,* 8–14
- shareable • *Programming Resources,* 5–3
- shareable, debugging • *Debugger,* 5–12
  - with DECwindows • *Debugger,* 1–30
- types of • *Linker,* 6–1

Image activation • *Linker,* 1–6, 2–11; *File Applications,* 5–5

Image activator
- description • *Linker,* 1–6
- global symbols • *System Dump Analyzer,* SDA–60
- GSMATCH processing • *Linker,* 3–8, 4–12
- locating a shareable image • *Linker,* 4–12
- mapping of shareable image • *Linker,* 4–1
- memory allocation • *Linker,* 6–7
- processing of .ADDRESS • *Linker,* 6–20

IMAGE clause
- for DEFINE SYNTAX statement • *Command Def,* CDU–23
- for DEFINE VERB statement • *Command Def,* CDU–31

Image exit • *System Services,* SYS–162

Image file
- linker's writing of • *Linker,* 6–21

Image header • *Linker,* 2–3, 2–11, LINK–10
- ID field • *Linker,* 1–8, 3–9
- image name field • *Linker,* 1–8, 3–10

Image I/O segment • *Linker,* 1–8, 3–9

Image I/O structures • *System Dump Analyzer,* SDA–77

Image-id field
- setting • *Linker,* 1–8, 3–9

Image initialization • *Linker,* 1–6, 2–11, 6–18

IMAGELIB.OLB
- See SYS$LIBRARY:IMAGELIB.OLB

Image map • *Linker,* LINK–11

Image map (Cont.)
- See also Linker Utility
- brief • *Linker,* LINK–3
- full • *Linker,* LINK–8
- linker's output • *Linker,* 2–6
- linker's writing of • *Linker,* 6–22
- linker output • *Linker,* 1–5
- module information in • *Linker,* 5–2, 5–3
- sections in • *Linker,* 1–5, 2–6, 5–2
- specification of • *Linker,* 1–12, 5–1
- symbol cross-referenced in • *Linker,* LINK–5
- type of • *Linker,* 1–12, 5–1

Image name field
- setting • *Linker,* 1–8, 3–10

/IMAGE qualifier • *System Dump Analyzer,* SDA–159

Image run-down • *Programming Resources,* 9–26
- effect on logical names • *System Services Intro,* 6–5
- forcing • *System Services,* SYS–191

Image section • *System Services Intro,* 11–19
- copy-on-reference • *Linker,* 4–3, 5–6
- demand-zero • *Linker,* 1–8, 3–7, 5–6, 6–19
- fix-up • *Linker,* 6–20, 6–21
- generation of • *Linker,* 6–3, 6–15
- initialization of • *Linker,* 6–18
- length of, in map • *Linker,* 5–5
- maximum number of • *Linker,* 3–10
- order of, in cluster • *Linker,* 6–17
- placement of program sections in • *Linker,* 6–15
- promotion of to global section • *Linker,* 4–1
- protection of • *Linker,* 5–6
- relocation of • *Linker,* 6–18
- type of • *Linker,* 2–11

Image section descriptor
- See ISD

Image specification
- effect of version number delimiter on overhead • *File Applications,* 5–5

Image termination • *Device Support (A),* 11–7; *Device Support (B),* 4–4

IMAGE_MANAGEMENT.EXE
- global symbols • *System Dump Analyzer,* SDA–60

IMGDEF.STB • *System Dump Analyzer,* SDA–60

Immediate conditional assembly block directive (.IIF) • *MACRO,* 6–46

Immediate mode • *MACRO,* 5–14
- contrasted with literal mode • *MACRO,* 5–15

Immediate mode addressing
- usage restricted in vector memory instructions • *MACRO,* 10–51, 10–53

Immediate value • *Routines Intro,* 2–3

Implementation table
VAX Ada • *Routines Intro*, A–13
VAX APL • *Routines Intro*, A–15
VAX BASIC • *Routines Intro*, A–18
VAX BLISS • *Routines Intro*, A–22
VAX C • *Routines Intro*, A–25
VAX COBOL • *Routines Intro*, A–28
VAX FORTRAN • *Routines Intro*, A–31
VAX MACRO • *Routines Intro*, A–36
VAX Pascal • *Routines Intro*, A–38
VAX PL/I • *Routines Intro*, A–42
VAX RPG II • *Routines Intro*, A–48
VAX SCAN • *Routines Intro*, A–51
VMS Usage • *Routines Intro*, A–1
INCB (Increment Byte) instruction • *MACRO*, 9–21
INCL (Increment Long) instruction • *MACRO*, 9–21
/INCLUDE positional qualifier • *Linker*, LINK–24
/INCLUDE qualifier • *Linker*, 2–4, 2–10
Inclusive OR operator • *MACRO*, 3–16
INCONSTATE bugcheck • *Device Support (B)*, 3–88, 3–97
INCW (Increment Word) instruction • *MACRO*, 9–21
Indefinite repeat argument directive (.IRP) • *MACRO*, 6–47
Indefinite repeat character directive (.IRPC) • *MACRO*, 6–49
Index
of a vector • *RTL Math*, MTH–149
INDEX (Compute Index) instruction • *MACRO*, 9–75
Index bucket
reclaiming • *Convert*, CONV–24
Index bucket area number
See IAN
Index bucket area number field
See XAB$B_IAN field
Index bucket fill size
See IFL
Index bucket fill size field
See XAB$W_IFL field
Index bucket size field
See XAB$B_IBS field
INDEX BUCKET structure • *File Applications*, 10–20
INDEX built-in procedure • *VAXTPU*, 7–230 to 7–231
Index compression
prohibition against using • *File Applications*, 3–3, 3–16, 3–25, 4–9
Index depth • *File Applications*, A–2
Index descriptor
See IDX
INDEXED attribute • *File Def Language*, FDL–22
Indexed file • *File Applications*, 2–18, 3–15

Indexed file (Cont.)
advantages and disadvantages of using • *File Applications*, 2–24
allocating • *File Applications*, A–1
alternate key • *File Applications*, 2–19
block allocation • *RMS*, 8–3
bucket size • *File Applications*, 3–6, 3–24, 7–20, A–1
bucket size for multiple areas • *RMS*, RMS–15
buffering • *File Applications*, 7–20
composition • *RMS*, RMS–18
compression • *File Applications*, 3–16, 3–25; *File Def Language*, FDL–28
creating • *RMS*, RMS–18
creating with multiple key • *RMS*, 4–5
default bucket size • *RMS*, 5–3
deferred-write option with • *File Applications*, 3–8
designing • *File Applications*, 3–15 to 3–28
determinating key value • *RMS*, RMS–48
determining keys and areas • *RMS*, 17–1
determining key size • *RMS*, 7–4
determining maximum record size • *RMS*, 5–21
determining number of buffers • *RMS*, 7–6
duplicate keys • *File Def Language*, FDL–27
establishing index • *RMS*, RMS–7
examining • *File Applications*, 10–19
example of processing duplicate keys • *RMS*, 7–8
example of specifying • *RMS*, 3–5
fast delete option • *RMS*, 7–15
fill factor • *File Applications*, 3–6
global buffers • *File Applications*, 7–21
identifying data area • *RMS*, 13–4
inhibiting index update • *RMS*, 13–12
initial extent quantity • *RMS*, 5–3
inserting records with Put service • *RMS*, RMS–71
invoking Get and Find services for • *RMS*, 7–4
key of reference • *RMS*, 7–3
key type • *File Applications*, 2–19
Level 1 index • *File Def Language*, FDL–28
loading • *Convert*, CONV–11
making contiguous • *File Applications*, 10–30
methods of accessing records • *RMS*, 7–5
optimizing performance • *File Applications*, 3–15 to 3–28
options • *RMS*, 7–10
positioning area • *RMS*, 8–7
primary key • *File Applications*, 2–19
Prolog 1 and Prolog 2 type • *File Applications*, 3–16
Prolog 3 • *Convert*, CONV–1
PROLOG selection • *RMS*, RMS–19
reclaiming buckets in • *File Applications*, 10–30

Indexed file
  record access • *File Applications,* 8–9 to 8–12, 8–12 to 8–13
  redesigning • *File Applications,* 10–28
  reformatting • *Convert,* CONV–1
  restriction against VFC format • *RMS,* 5–18
  restriction to changing primary key • *RMS,* RMS–100
  run-time options • *File Applications,* 9–12 to 9–13
  separating index levels • *RMS,* 13–11
  setting bucket size • *RMS,* 5–4
  size of data bucket • *RMS,* 13–4
  specifying bucket size • *RMS,* 8–5
  specifying index area • *RMS,* 13–10
  specifying index bucket size • *RMS,* 13–10
  string key options • *RMS,* 13–8
  structure • *Analyze/RMS_File,* ARMS–1
  tuning • *File Applications,* 3–15 to 3–28
  update-if option • *RMS,* 7–17
  use of areas in • *RMS,* 4–8
  use of end-of-file option • *RMS,* RMS–7
  verifying sort order • *RMS,* RMS–7
  with allocation options • *RMS,* 5–14
  with collating sequences • *RMS,* 13–3
  with deferred-write option • *RMS,* RMS–12
  with Get service • *RMS,* RMS–48
  with global buffers • *File Applications,* 3–27
  with XABKEY • *RMS,* 13–1
Indexed file compression • *File Applications,* 3–3
Indexed file organization • *File Applications,* 1–2
  reorganizing • *File Applications,* 10–31
/INDEXED qualifier • *File Applications,* 7–20
Indexing
  backward • *RTL Math,* 2–6
  forward • *RTL Math,* 2–6
Index keywords
  in help libraries • *Librarian,* LIB–4
Index levels • *File Def Language,* FDL–5
  comparing primary key and alternate keys • *RMS,* 13–10
Index mode • *MACRO,* 5–16
  operand specifier format • *MACRO,* 8–26
/INDEX qualifier • *System Dump Analyzer,* SDA–73, SDA–126
Index records • *File Def Language,* FDL–5
Index structure • *File Applications,* 3–15, 3–24
  Level 0 • *File Applications,* 3–17
  Level 1 • *File Applications,* 3–17
  primary • *File Applications,* 3–17
INDEX_AREA attribute • *File Def Language,* FDL–27, FDL–28
INDEX_AREA secondary attribute • *File Applications,* 3–24
INDEX_COMPRESSION attribute • *File Def Language,* FDL–5, FDL–28
INDEX_FILL attribute • *File Def Language,* FDL–5, FDL–28
INDEX_SPACE_OCCUPIED attribute • *File Def Language,* FDL–5
Indirection operator
  See Contents-of operator
Information
  retrieving about subordinate • *RTL Parallel Processing,* 2–4
INFORMATIONAL keyword • *VAXTPU,* 7–397
/INFORMATIONAL qualifier
  in message definition • *Message,* MSG–23
"Informational" string constant parameter to GET_ INFO • *VAXTPU,* 7–206
INFO_WINDOW identifier • *VAXTPU,* 7–506
INFO_WINDOW variable • *VAXTPU,* 4–29
INI$BRK • *Delta/XDelta,* DELTA–7, DELTA–29; *Device Support (A),* 13–6
Initial breakpoint in XDELTA • *Delta/XDelta,* DELTA–7
Initialization • *Modular Procedures,* 3–12, A–4
  at run time • *Modular Procedures,* 3–17
  automatic • *RTL Parallel Processing,* 2–1
  debugging session • *Debugger,* 3–1, 9–7
    with DECwindows • *Debugger,* 1–4
  of modular procedures • *Modular Procedures,* 3–12
  of storage • *Modular Procedures,* 3–14
  using LIB$INITIALIZE • *Modular Procedures,* 3–17, A–4
Initialization code • *Debugger,* 9–10
  with DECwindows • *Debugger,* 1–4
Initialization file
  See also Command procedure, debugger
  debugger • *Debugger,* 8–4, D–1
    with DECwindows • *Debugger,* 1–29
  default handling • *VAXTPU,* 4–22
  definition • *VAXTPU,* 1–11
  during a session • *VAXTPU,* 4–32
  effects on buffer settings • *VAXTPU,* 4–32
  EVE editor • *VAXTPU,* 4–31 to 4–33
Initialization macro
  advantages described • *RMS,* 3–7
  example • *RMS,* 3–5
  functions • *RMS,* 3–1
  multiple bit field • *RMS,* 3–5
  placement guidelines • *RMS,* 3–7
  using • *RMS,* 3–6
/INITIALIZATION qualifier • *VAXTPU,* 5–9 to 5–10

Initialization routine
  See Controller initialization routine
  See Unit initialization routine
"Initialization" string constant parameter to GET_INFO • *VAXTPU,* 7–177
Initialization table • *Device Support (A),* 6–2; *Device Support (B),* 1–34, 2–25
"Initialization_file" string constant parameter to GET_INFO • *VAXTPU,* 7–177
INITIALIZE command • *I/O User's I,* 6–27
  and window size • *File Applications,* 9–8
Initialize command table
  LPA11-K device • *I/O User's I,* 4–9
/INITIALIZE qualifier • *Patch,* PAT–19
  with SET PATCH_AREA command • *Patch,* PAT–79
Initializing variables • *VAXTPU,* 2–24
Initiator • *Device Support (A),* 17–2
  completing an operation (in AEN mode) • *Device Support (B),* 2–74
  enabling selection of • *Device Support (A),* 17–28 to 17–30; *Device Support (B),* 2–70, 2–73 to 2–90
  receiving data from target (in AEN mode) • *Device Support (B),* 2–80
  sending bytes to target (in AEN mode) • *Device Support (B),* 2–83
INIT processor state • *Device Support (B),* 1–16
"Init_file" string constant parameter to GET_INFO • *VAXTPU,* 7–177
Inner product
  of a vector • *RTL Math,* MTH–165
Input, debugger
  DBG$DECW$DISPLAY
    with DECwindows • *Debugger,* 1–36, D–1
  DBG$INPUT • *Debugger,* 9–5, D–1
Input address array • *System Services Intro,* 11–4
Input data register
  See DR11–W/DRV11–WA driver, IDR
Input device • *Device Support (B),* 1–75
Input file • *VAXTPU,* 1–9, 5–19
  concatenating • *Convert,* CONV–5
  default file type for • *National Char Set,* NCS–21
  restriction to using shareable image • *Linker,* 1–1
  specifying for NCS command • *National Char Set,* NCS–21
Input file specification • *Librarian,* LIB–11
  default file type • *Librarian,* LIB–12
Input focus
  determining ownership of • *VAXTPU,* 7–199
  fetching grab routine for • *VAXTPU,* 7–199
  fetching ungrab routine for • *VAXTPU,* 7–202
Input focus (Cont.)
  requesting • *VAXTPU,* 7–398
  specifying grab routine for • *VAXTPU,* 7–400
  specifying ungrab routine for • *VAXTPU,* 7–402
  support for • *VAXTPU,* 4–5 to 4–6
Input image file • *Patch,* PAT–3
  device driver image • *Patch,* PAT–3, PAT–19
  executable • *Patch,* PAT–3
  shareable • *Patch,* PAT–3
Input/output
  See also I/O
  asynchronous • *Programming Resources,* 7–47
  channel • *Programming Resources,* 7–45
  checking device type • *Programming Resources,* 7–50
  complex • *Programming Resources,* 7–2
  device • *Programming Resources,* 1–23
  echo • *Programming Resources,* 7–40
  exit handler • *Programming Resources,* 7–53
  file • *Programming Resources,* 1–23
  lowercase • *Programming Resources,* 7–42
  reading a single line • *Programming Resources,* 7–4
  reading several lines • *Programming Resources,* 7–5
  screen updates • *Programming Resources,* 7–31
  simple • *Programming Resources,* 7–1
  status of • *Programming Resources,* 7–49
  synchronous • *Programming Resources,* 7–46
  terminator • *Programming Resources,* 7–4
    end-of-file • *Programming Resources,* 7–54
    record • *Programming Resources,* 7–53
  timeout • *Programming Resources,* 7–41
  unsolicited input • *Programming Resources,* 7–36
  uppercase • *Programming Resources,* 7–42
  using SYS$QIO • *Programming Resources,* 7–45, 7–49
  using SYS$QIOW • *Programming Resources,* 7–45, 7–49
  writing simple character data • *Programming Resources,* 7–6
/INPUT qualifier • *Debugger,* 7–20, CD–123, CD–168, CD–263; *System Dump Analyzer,* SDA–162
Input queue
  See DR32 driver, INPTQ
Input source file • *SUMSLP,* SUM–1
INRANGE case constant • *VAXTPU,* 3–24
INSERT command • *Patch,* PAT–68
  with /ABSOLUTE qualifier • *Patch,* PAT–27
  with /INSTRUCTION qualifier • *Patch,* PAT–69
Inserted records • *VAXTPU,* 6–5
Inserting date • *VAXTPU,* 7–138, 7–268, 7–271

Inserting record
  program example • *RMS*, 4–16
Inserting time • *VAXTPU*, 7–138, 7–268, 7–271
Insertion of files/modules • *Librarian*, LIB–27
  See also /REPLACE qualifier
Insertion operations • *RTL Screen Management*, 2–8
INSERT keyword • *VAXTPU*, 7–404
Insert mode
  COPY_TEXT • *VAXTPU*, 7–53
  MOVE_TEXT • *VAXTPU*, 7–280
/INSERT qualifier • *Librarian*, LIB–12, LIB–27; *National Char Set*, NCS–32
INSQHI (Insert Entry into Queue at Head, Interlocked) instruction • *MACRO*, 9–89
INSQTI (Insert Entry into Queue at Tail, Interlocked) instruction • *MACRO*, 9–91
INSQUE (Insert Entry in Queue) instruction • *MACRO*, 9–93
Install
  privileged image • *Programming Resources*, 6–2
Installation
  of shareable image • *Linker*, 4–1, 4–12
  requirement for sharing • *Linker*, 4–2
  /SHARE • *Linker*, 4–12
Install Utility (INSTALL)
  benefits of using for shareable image • *Linker*, 1–9
Instruction • *MACRO*, 1–1, 9–1
  See also Vector instruction
  address • *MACRO*, 9–33
  arithmetic • *MACRO*, 9–5, 9–101, 9–144
  as operator • *MACRO*, 2–3
  character string • *MACRO*, 9–126
  control • *MACRO*, 9–42
  decimal string • *MACRO*, 9–144
  depositing • *Debugger*, 4–20, 4–22
    with DECwindows • *Debugger*, 1–26
  display (INST) • *Debugger*, 4–20, 7–8, 10–16, C–5
    for routine on call stack • *Debugger*, 7–10, CD–170
    with DECwindows • *Debugger*, 1–8, 1–11, 1–22
  display kind • *Debugger*, 7–17, C–1
  EXAMINE/INSTRUCTION command • *Debugger*, 4–20, 7–9, C–5
  EXAMINE/OPERANDS command • *Debugger*, 4–20
  examining • *Debugger*, 4–20, 7–8
    with DECwindows • *Debugger*, 1–22, 1–26
  floating-point • *MACRO*, 9–101
  format • *MACRO*, 8–16
  integer • *MACRO*, 9–5
  interlocked • *Programming Resources*, 4–18

Instruction (Cont.)
  logical • *MACRO*, 9–5
  operand • *Debugger*, 4–20, CD–87, CD–153
  optimized code • *Debugger*, 7–8, 9–1
    with DECwindows • *Debugger*, 1–11, 1–22
  packed decimal • *MACRO*, 9–144
  procedure call • *MACRO*, 9–63
  queue • *Programming Resources*, 4–19; *MACRO*, 9–82
  selecting from DECwindows window • *Debugger*, 1–23
  set • *MACRO*, 9–1
  SET SCOPE/CURRENT command • *Debugger*, 7–10, CD–170
  string • *MACRO*, 9–126, 9–144
  variable-length bit field • *MACRO*, 9–36
  vector • *MACRO*, 10–9, 10–18, 10–21
  window (INST), DECwindows • *Debugger*, 1–11, 1–22
/INSTRUCTION-/NOINSTRUCTION qualifier
  with DELETE command • *Patch*, PAT–53
  with DEPOSIT command • *Patch*, PAT–56, PAT–57
  with EVALUATE command • *Patch*, PAT–60
  with EXAMINE command • *Patch*, PAT–63
  with INSERT command • *Patch*, PAT–68
  with REPLACE command • *Patch*, PAT–72
  with SET MODE command • *Patch*, PAT–76
  with VERIFY command • *Patch*, PAT–91
INSTRUCTION-NOINSTRUCTION mode • *Patch*, PAT–15
Instruction notation
  operand specifier • *MACRO*, 9–2
  operation description • *MACRO*, 9–3
/INSTRUCTION qualifier • *Debugger*, 7–9, 7–21, CD–18, CD–31, CD–63, CD–87, CD–123, CD–132, CD–188, CD–265; *System Dump Analyzer*, SDA–51
Instructions, how to display • *Delta/XDelta*, DELTA–20
%INST_SCOPE • *Debugger*, 7–17, C–5
Insufficient virtual memory error
  reasons for • *RTL Parallel Processing*, PPL–11
INSV (Insert Field) instruction • *MACRO*, 9–41
INT2 value • *File Def Language*, FDL–32
INT4 value • *File Def Language*, FDL–32
INT8 value • *File Def Language*, FDL–32
INT built-in procedure • *VAXTPU*, 7–232 to 7–233
Integer
  data type • *MACRO*, 8–1
  in source statement • *MACRO*, 3–3
  unsigned • *MACRO*, 8–1, 8–2
Integer and floating-point routine • *RTL Library*, 2–12
Integer constants • *VAXTPU*, 3–5

INTEGER data type • *VAXTPU*, 2–5
Integer instructions • *MACRO*, 9–5
vector • *MACRO*, 10–57
Integer overflow • *RTL Library*, LIB–255
Integer overflow enable (IV) • *MACRO*, 8–15
Integer to floating-point conversion • *RTL Math*, 1–5
Integer type • *Debugger*, 4–15, 4–25, 4–27
Integration stage • *Modular Procedures*, 5–1
Integration testing • *Modular Procedures*, 4–1, 4–5
Integrity of file • *Analyze/RMS_File*, ARMS–13
Interactive command • *Analyze/RMS_File*, ARMS–21
Interactive mode • *Analyze/RMS_File*, ARMS–1, ARMS–10, ARMS–15, ARMS–21
Interactive processing of selective patches • *Patch*, PAT–35
/INTERACTIVE qualifier • *File Applications*, 10–11; *Analyze/RMS_File*, ARMS–1, ARMS–15
limitation • *Analyze/RMS_File*, ARMS–10, ARMS–13, ARMS–14, ARMS–20
using with /OUTPUT qualifier • *Analyze/RMS_File*, ARMS–16
/INTERFACE qualifier • *VAXTPU*, 5–10
Interlocked instructions • *Programming Resources*, 4–18; *MACRO*, 10–43
using in multiprocessing environment • *Device Support (A)*, E–13 to E–14
Interlocked queue
validating • *System Dump Analyzer*, SDA–164
Internal buffer • *File Applications*, 8–3
Internal file access block
See IFAB
Internal file identifier
See IFI
Internal file identifier field
See FAB$W_IFI field
Internal processor register
See IPR
See Processor register symbol
Internal record access block
See IRAB
Internal stream identifier field
See RAB$W_ISI field
Internal structure
of file • *Analyze/RMS_File*, ARMS–1
Interpreters
VAX APL • *Programming Resources*, 1–6
VAX BASIC • *Programming Resources*, 1–6
VAX LISP • *Programming Resources*, 1–8
Interprocess communication • *Programming Resources*, 3–7; *System Services Intro*, 8–7, 8–10
using event flags for • *System Services Intro*, 8–10
Interprocess communication (Cont.)
using global sections for • *System Services Intro*, 8–11
using lock management services for • *System Services Intro*, 8–11
using logical names for • *System Services Intro*, 8–10
using mailboxes • *Programming Resources*, 3–7; *System Services Intro*, 8–11
Interprocess control • *System Services Intro*, 8–7
Interprocessor interrupt • *Device Support (A)*, 3–4, 3–14; *Device Support (B)*, 1–16
Interrecord gap
See IRG
Interrupt • *Device Support (A)*, 3–3
See also Device interrupt
blocking • *Device Support (B)*, 2–27, 2–65
debugging session • *Debugger*, 3–5
dismissing • *Device Support (A)*, 10–1
execution of command • *Debugger*, 2–7, CD–38
with DECwindows • *Debugger*, 1–21
execution of program • *Debugger*, 2–7, 3–3, 10–6, 10–10, 10–14, CD–36, CD–38, CD–41, CD–152
with DECwindows • *Debugger*, 1–21
interprocessor • *Device Support (A)*, 3–4, 3–14; *Device Support (B)*, 1–16
reasons for DR32 • *I/O User's II*, 4–3
requesting an XDELTA • *Device Support (A)*, 13–7 to 13–8
requesting a software • *Device Support (A)*, 3–10; *Device Support (B)*, 2–67
Interrupt context • *Device Support (A)*, 1–8, 9–3
Interrupt dispatch block
See IDB
Interrupt dispatcher • *Device Support (A)*, 3–6, 14–24, 16–9, 16–11; *Device Support (B)*, 1–7, 1–9
for MASSBUS • *Device Support (A)*, 15–8 to 15–12, 15–15 to 15–16; *Device Support (B)*, 4–24
for Q22 bus • *Device Support (A)*, 14–26 to 14–34
for UNIBUS • *Device Support (A)*, 14–26 to 14–34; *Device Support (B)*, 1–25
Interrupt enable bit • *Device Support (A)*, 8–4
Interrupt expected bit
See UCB$V_INT
Interruption of program • *VAXTPU*, 4–20
Interrupt priority level
See IPL
/INTERRUPT qualifier • *System Dump Analyzer*, SDA–157

- Interrupt request for XDELTA • *Delta/XDelta,* DELTA–6 to DELTA–7
- Interrupts • *MACRO,* 10–43
- Interrupt service routine • *Device Support (A),* 1–3, 3–3, 3–15, 9–1 to 9–8, 14–24; *Device Support (B),* 1–73
  - address • *Device Support (A),* 6–3, 14–32, E–5; *Device Support (B),* 1–25, 2–26, 4–13
  - context • *Device Support (A),* 9–3; *Device Support (B),* 4–13
  - entry point • *Device Support (A),* 4–16; *Device Support (B),* 4–13
  - example • *Device Support (A),* 9–6 to 9–8
  - exit method • *Device Support (B),* 4–14
  - for connect to interrupt facility • *Device Support (A),* 19–10, 19–16 to 19–18
  - for LP11 printer • *Device Support (A),* 2–6 to 2–7
  - for MASSBUS device • *Device Support (A),* 15–12, 15–17; *Device Support (B),* 4–13
  - for solicited interrupt • *Device Support (A),* 9–3 to 9–4
  - for terminal port driver • *Device Support (A),* 18–18
  - for unsolicited interrupt • *Device Support (A),* 9–4 to 9–8; *Device Support (B),* 4–24
  - functions • *Device Support (A),* 4–16, 9–1; *Device Support (B),* 4–14
  - input • *Device Support (B),* 4–14
  - of CONINTERR.EXE • *Device Support (A),* 19–13
  - of UNIBUS adapter • *Device Support (A),* 14–29
  - preemption of device timeout handling • *Device Support (A),* 10–5
  - register usage • *Device Support (A),* 8–7; *Device Support (B),* 4–14
  - specifying more than one • *Device Support (B),* 4–13
  - synchronization requirements • *Device Support (A),* 3–6, 3–22, 9–3, E–11; *Device Support (B),* 4–13
- Interrupt stack • *Device Support (A),* 8–1
  - address • *Device Support (B),* 1–16
  - displaying contents • *System Dump Analyzer,* SDA–157
- Interrupt transfer routine • *Device Support (A),* 14–31
- Interrupt transfer vector
  - See VEC
- Interrupt vector • *Device Support (A),* 12–11
  - See Device interrupt vector
  - number • *Device Support (A),* 12–6
- Intersystem communication • *Programming Resources,* 3–26
- Interval clock • *Device Support (A),* 3–6, 3–8, 3–14
- Interval clock (Cont.)
  - interrupt service routine • *Device Support (A),* 3–8, 3–9
  - role in device timeouts • *Device Support (A),* 1–4
- /INTO qualifier • *Debugger,* CD–132, CD–189, CD–200, CD–265
- Intraprocess communication • *Programming Resources,* 3–1
  - common blocks • *Programming Resources,* 3–6
  - global symbols • *Programming Resources,* 3–6
- INVALIDATE spin lock • *Device Support (A),* 3–14
- INVALIDATE_TB macro • *Device Support (A),* E–15; *Device Support (B),* 2–41 to 2–42
- INVALID macro
  - replaced by INVALIDATE_TB macro • *Device Support (A),* E–15
- INVEXCEPTN bugcheck • *System Dump Analyzer,* SDA–16
- Invisible record • *VAXTPU,* 7–448
- INVOKE command • *File Applications,* 4–5; *File Def Language,* FDL–57, FDL–63
- Invoking
  - See also Bootstrap procedures for XDELTA
  - See also Interrupt request for XDELTA
  - ANALYZE/RMS_FILE • *Analyze/RMS_File,* ARMS–10
  - CONVERT • *Convert,* CONV–5
  - CONVERT/RECLAIM • *Convert,* CONV–5
  - CREATE/FDL • *File Def Language,* FDL–43
  - debugger • *Debugger,* 2–5, 2–6, 3–1, 10–1, 10–15, CD–41
    - with DECwindows • *Debugger,* 1–2, 1–4, 1–33
  - DELTA • *Delta/XDelta,* DELTA–1
  - EDIT/FDL • *File Def Language,* FDL–43
  - VAXTPU • *VAXTPU,* 1–9, 5–1
    - from a batch job • *VAXTPU,* 5–5
    - from DCL command procedure • *VAXTPU,* 5–2
    - interactively • *VAXTPU,* 5–1
    - restriction • *VAXTPU,* 5–1
  - XDELTA • *Delta/XDelta,* DELTA–2
- IO$M_NOW modifier
  - for Get and Put services • *RMS,* 7–14
- IO$V_INHERLOG • *Device Support (B),* 3–8
- IO$_AVAILABLE function • *Device Support (A),* 7–9
- IO$_CONINTREAD function • *Device Support (A),* 19–9, 19–10
- IO$_CONINTWRITE function • *Device Support (A),* 19–9, 19–10
- IO$_PACKACK function • *Device Support (A),* 7–9
- IO$_SENSECHAR function
  - servicing • *Device Support (B),* 3–49

IO$_SENSEMODE function
  servicing • *Device Support (B)*, 3–49
IO$_SETCHAR function • *Device Support (A)*, 11–9
  servicing • *Device Support (B)*, 3–50 to 3–51
IO$_SETMODE function • *Device Support (A)*, 18–15
  servicing • *Device Support (B)*, 3–50 to 3–51
IO$_TTY_PORT function • *Device Support (A)*, 18–14
IO$_UNLOAD function • *Device Support (A)*, 7–9
$IO650DEF macro • *Device Support (A)*, 19–1
$IO730DEF macro • *Device Support (A)*, 19–1
$IO750DEF macro • *Device Support (A)*, 19–1
$IO780DEF macro • *Device Support (A)*, 19–1
$IO790DEF macro • *Device Support (A)*, 19–1
$IO8NNDEF macro • *Device Support (A)*, 16–17, 19–1
$IO8PSDEF macro • *Device Support (A)*, 16–17
$IO8SSDEF macro • *Device Support (A)*, 16–16, 19–1
$IO9AQDEF macro • *Device Support (A)*, 16–17
$IO9CCDEF macro • *Device Support (A)*, 16–17, 19–1
IOC$ALLOSPT
  replaced by LDR$ALLOC_PT • *Device Support (A)*, E–7
IOC$ALOALTMAP • *Device Support (B)*, 1–10, 3–63 to 3–64, 3–93
IOC$ALOALTMAPN • *Device Support (A)*, 14–20; *Device Support (B)*, 3–63 to 3–64
IOC$ALOALTMAPSP • *Device Support (B)*, 3–63 to 3–64
IOC$ALOUBAMAP • *Device Support (B)*, 3–65 to 3–66, 3–90, 3–99
IOC$ALOUBAMAPN • *Device Support (A)*, 14–20; *Device Support (B)*, 3–65 to 3–66
IOC$APPLYECC • *Device Support (B)*, 1–83, 3–67
IOC$CANCELIO • *Device Support (A)*, 11–8 to 11–9; *Device Support (B)*, 1–77, 3–68, 4–4
IOC$DIAGBUFILL • *Device Support (B)*, 1–30, 1–42, 3–69
IOC$GL_CRBTMOUT • *Device Support (B)*, 1–22
IOC$GL_DEVLIST • *Device Support (A)*, 11–5; *Device Support (B)*, 1–27
IOC$GL_DPTLIST • *Device Support (A)*, 12–3, 12–8
IOC$GL_IRPFL
  replaced in VMS Version 5.0 • *Device Support (A)*, E–14
IOC$GL_LRPFL
  replaced in VMS Version 5.0 • *Device Support (A)*, E–14
IOC$GL_MUTEX • *Device Support (A)*, 11–12; *Device Support (B)*, 4–6
IOC$GL_PSFL
  replaced by IOC$GQ_POSTIQ • *Device Support (A)*, E–14
IOC$GL_SRPFL
  replaced in VMS Version 5.0 • *Device Support (A)*, E–14
IOC$GQ_IRPIQ • *Device Support (A)*, E–14
IOC$GQ_LRPIQ • *Device Support (A)*, E–14
IOC$GQ_SRPIQ • *Device Support (A)*, E–14
IOC$GW_MAXBUF • *Device Support (B)*, 3–20, 3–22
IOC$INITIATE • *Device Support (A)*, 3–23, 4–13 to 4–15, 8–1, 10–3; *Device Support (B)*, 1–30, 1–40, 1–41, 1–77, 1–79, 3–28, 3–38, 3–69, 3–70 to 3–71, 3–95, 4–17
IOC$IOPOST • *Device Support (A)*, 3–5; *Device Support (B)*, 1–41, 1–42, 1–43, 3–72 to 3–73
  unlocking process buffers • *Device Support (B)*, 3–109
IOC$LOADALTMAP • *Device Support (A)*, 14–22; *Device Support (B)*, 2–44, 3–74 to 3–75
IOC$LOADMBAMAP • *Device Support (A)*, 15–3 to 15–4; *Device Support (B)*, 2–45, 3–76
IOC$LOADUBAMAP • *Device Support (A)*, 14–21 to 14–22; *Device Support (B)*, 1–26, 2–46, 3–77 to 3–78
IOC$LOADUBAMAPA • *Device Support (A)*, 14–22; *Device Support (B)*, 3–77 to 3–78
IOC$MNTVER • *Device Support (B)*, 1–30
IOC$MOVFRUSER • *Device Support (A)*, 16–22; *Device Support (B)*, 2–21, 3–79
IOC$MOVFRUSER2 • *Device Support (B)*, 3–79
IOC$MOVTOUSER • *Device Support (A)*, 16–22; *Device Support (B)*, 2–21, 3–80 to 3–81
IOC$MOVTOUSER2 • *Device Support (B)*, 3–80 to 3–81
IOC$PURGDATAP • *Device Support (A)*, 14–24 to 14–25; *Device Support (B)*, 1–26, 2–51, 3–82 to 3–83
IOC$RELALTMAP • *Device Support (A)*, 14–26; *Device Support (B)*, 1–10, 1–73, 2–53, 3–84 to 3–85
IOC$RELCHAN • *Device Support (A)*, 10–2; *Device Support (B)*, 1–21, 1–36, 1–73, 2–54, 3–86, 3–95
  called by IOC$WFIRLCH • *Device Support (B)*, 3–106
IOC$RELDATAP • *Device Support (A)*, 14–25; *Device Support (B)*, 1–7, 1–9, 1–73, 2–55, 3–87
IOC$RELMAPREG • *Device Support (A)*, 14–26; *Device Support (B)*, 1–8, 1–9, 1–25, 1–26, 1–73, 2–56, 3–89 to 3–90
IOC$RELSCHAN • *Device Support (B)*, 1–21, 1–22, 1–36, 2–57, 3–91

- IOC$REQALTMAP • *Device Support (A),* 14–19; *Device Support (B),* 1–10, 1–73, 2–58, 3–92 to 3–93
- IOC$REQCOM • *Device Support (A),* 3–5, 3–23, 8–1, 10–3 to 10–4; *Device Support (B),* 1–30, 1–38, 1–41, 1–76, 1–77, 1–79, 1–81, 2–59, 3–13, 3–94 to 3–95, 4–17
  - error logging activities • *Device Support (A),* 11–10
- IOC$REQDATAP • *Device Support (A),* 14–17; *Device Support (B),* 1–7, 1–9, 1–26, 1–73, 2–60, 3–96 to 3–97
- IOC$REQDATAPNW • *Device Support (A),* 14–18; *Device Support (B),* 3–96 to 3–97
- IOC$REQMAPREG • *Device Support (A),* 14–19 to 14–20; *Device Support (B),* 1–8, 1–9, 1–25, 1–26, 1–73, 2–61, 3–98 to 3–99
- IOC$REQPCHANH • *Device Support (B),* 1–21, 1–36, 1–73, 2–62, 3–100 to 3–101
- IOC$REQPCHANL • *Device Support (A),* 8–2 to 8–4; *Device Support (B),* 1–21, 1–36, 1–73, 2–62, 3–100 to 3–101
- IOC$REQSCHANH • *Device Support (B),* 1–21, 1–22, 1–36, 2–63, 3–100 to 3–101
- IOC$REQSCHANL • *Device Support (B),* 1–21, 1–22, 1–36, 1–73, 2–63, 3–100 to 3–101
- IOC$RETURN • *Device Support (A),* 11–8; *Device Support (B),* 2–13, 3–102
- IOC$SEARCHDEV • *Device Support (B),* 1–74
- IOC$VERIFYCHAN • *Device Support (B),* 3–103
- IOC$WFIKPCH • *Device Support (A),* 4–16, 8–7; *Device Support (B),* 1–73, 1–77, 1–79, 3–104 to 3–106
- IOC$WFIRLCH • *Device Support (A),* 4–16; *Device Support (B),* 1–77, 1–79, 3–104 to 3–106
- $IODEF macro • *Device Support (A),* 6–5
- IOFORK macro • *Device Support (A),* 3–12, 3–24, 4–17, 9–4, 10–1, 14–24; *Device Support (B),* 2–43, 3–30
- IOLOCK10 fork lock • *Device Support (A),* 3–14
- IOLOCK11 fork lock • *Device Support (A),* 3–14
- IOLOCK8 fork lock • *Device Support (A),* 3–8, 3–13
- IOLOCK9 fork lock • *Device Support (A),* 3–14
- IOSB • *Routines Intro,* A–7t
- IOSB (I/O status block) • *Device Support (A),* 7–4, 10–2, 10–3; *Device Support (B),* 1–39, 1–41, 3–5, 3–10, 3–73, 3–95
  - asynchronous DDCMP driver • *I/O User's II,* 5–14
  - DMC11/DMR11 driver • *I/O User's II,* 1–9
  - DMP11/DMF32 driver • *I/O User's II,* 2–25
  - DR11–WDRV11–WA driver • *I/O User's II,* 3–15
  - DR32 driver • *I/O User's II,* 4–34
  - Ethernet/802 drivers • *I/O User's II,* 6–39
  - in synchronization • *System Services Intro,* 7–15
- IOSB (I/O status block) (Cont.)
  - return condition value field • *System Services Intro,* 7–19
  - validating access to • *Device Support (A),* 4–9
- IOTA (Generate Compressed Iota Vector) instruction • *MACRO,* 10–86
- $IOUV1DEF macro • *Device Support (A),* 19–1
- $IOUV2DEF macro • *Device Support (A),* 19–1
- IO_ROUTINES.EXE
  - global symbols • *System Dump Analyzer,* SDA–60
- io_status_block data type • *Routines Intro,* A–7t
- IPL$_ASTDEL • *Device Support (A),* 3–2, 3–4, 3–19, 4–9; *Device Support (B),* 3–10, 3–12, 3–31, 3–34, 3–37, 3–38, 3–40, 3–43, 3–49, 3–50, 3–56, 3–62, 3–73, 3–103, 3–114, 3–116, 3–117, 4–6, 4–11
  - PGFIPLHI bugcheck • *System Dump Analyzer,* SDA–19
- IPL$_EMB • *Device Support (B),* 3–8
- IPL$_FILSYS • *Device Support (A),* 3–13
- IPL$_IOLOCK8 • *Device Support (A),* 3–13
- IPL$_IOPOST • *Device Support (A),* 2–7, 3–2, 3–5, 4–20, 10–3, 11–7; *Device Support (B),* 3–5, 3–10, 3–25, 3–73, 3–95
- IPL$_JIB • *Device Support (A),* 3–13
- IPL$_MAILBOX • *Device Support (A),* 3–2, 3–8, 3–14, 9–7, 10–7; *Device Support (B),* 3–52, 3–61
- IPL$_MMG • *Device Support (A),* 3–13
- IPL$_POOL • *Device Support (A),* 3–2; *Device Support (B),* 3–14, 3–15
- IPL$_POWER • *Device Support (A),* 3–7, 8–5 to 8–6, 11–4, 12–4; *Device Support (B),* 4–8, 4–10
- IPL$_QUEUEAST • *Device Support (A),* 3–2, 3–7, 3–13, 19–15, 19–18; *Device Support (B),* 3–2, 3–3
- IPL$_RESCHED • *Device Support (A),* 3–2, 3–5, 3–7; *Device Support (B),* 2–31, 3–111, 3–113
- IPL$_SCHED • *Device Support (A),* 3–13
- IPL$_SYNCH • *Device Support (A),* 3–2, 3–7, 3–8
- IPL$_TIMER • *Device Support (A),* 3–13; *Device Support (B),* 3–29, 3–48
- IPL$_TIMERFORK • *Device Support (A),* 3–2, 3–8, 10–4, 10–5
- IPL (interrupt priority level) • *Device Support (A),* 1–7, 3–1 to 3–12
  - See also Device IPL
  - See also Fork IPL
  - hardware • *Device Support (A),* 3–1
  - lowering • *Device Support (A),* 3–9 to 3–12, 3–23, 8–7; *Device Support (B),* 2–97, 3–26, 3–30
  - modifying • *Device Support (B),* 2–17 to 2–18, 2–19 to 2–20, 2–27, 2–28, 2–33 to 2–34, 2–35 to 2–36, 2–47 to 2–48, 2–65, 2–96

IPL (interrupt priority level) (Cont.)
raising • *Device Support (A),* 3–9 to 3–12, 3–15; *Device Support (B),* 2–49, 2–65
relation to spin lock • *Device Support (A),* 3–15
saving • *Device Support (A),* 3–10; *Device Support (B),* 2–17, 2–33, 2–47, 2–64
software • *Device Support (A),* 3–2
IPR (internal processor register)
vector • *MACRO,* 10–3, 10–9
IRAB (internal record access block) • *System Dump Analyzer,* SDA–77
IRG (interrecord gap) • *File Applications,* 1–8
IRP$B_CARCON • *Device Support (B),* 1–41, 3–32, 3–41, 3–55
IRP$B_PRI • *Device Support (B),* 3–27
IRP$L_BCNT • *Device Support (A),* 8–2; *Device Support (B),* 3–32, 3–35, 3–41, 3–43, 3–46, 3–55, 3–56, 3–59, 3–70, 3–71, 3–72
writing • *Device Support (A),* 7–6
IRP$L_DIAGBUF • *Device Support (B),* 3–69, 3–70, 3–71
IRP$L_IOST2 • *Device Support (B),* 3–32, 3–41, 3–55
IRP$L_KEYDESC • *Device Support (B),* 3–72
IRP$L_MEDIA • *Device Support (A),* 7–4, 10–3, 11–7; *Device Support (B),* 1–41, 3–37, 3–51, 3–62
IRP$L_PID • *Device Support (A),* 11–8; *Device Support (B),* 3–68, 4–5
IRP$L_SVAPTE • *Device Support (A),* 8–2; *Device Support (B),* 3–33, 3–35, 3–41, 3–46, 3–55, 3–59, 3–70, 3–71
for buffered I/O • *Device Support (A),* 7–7, 7–8
IRP$V_BUFIO • *Device Support (B),* 3–72
IRP$V_DIAGBUF • *Device Support (B),* 3–69, 3–70, 3–71, 3–72
IRP$V_EXTEND • *Device Support (B),* 3–72
IRP$V_FUNC • *Device Support (A),* 7–6, 7–8, 11–7; *Device Support (B),* 3–32, 3–35, 3–41, 3–43, 3–46
IRP$V_KEY • *Device Support (B),* 3–72
IRP$V_MBXIO • *Device Support (B),* 3–72
IRP$V_PHYSIO • *Device Support (B),* 3–72
IRP$W_BOFF • *Device Support (A),* 7–7, 7–8, 8–2; *Device Support (B),* 3–33, 3–35, 3–41, 3–46, 3–55, 3–59, 3–70, 3–71, 3–72
IRP$W_CHAN • *Device Support (A),* 11–8; *Device Support (B),* 3–68, 4–5
IRP$W_FUNC • *Device Support (A),* 8–4
IRP$W_STS
for read function • *Device Support (A),* 7–6, 7–8
for write function • *Device Support (A),* 7–8
IRP (I/O request packet) • *System Dump Analyzer,* SDA–99, SDA–118; *Device Support (A),* 1–6 to 1–7; *Device Support (B),* 1–37 to 1–42
IRP (I/O request packet) (Cont.)
allocating • *Device Support (A),* 4–9
copying to UCB • *Device Support (A),* 8–2
creation • *Device Support (A),* 2–3, 4–9
current • *Device Support (B),* 1–77
deallocation • *Device Support (A),* 2–7; *Device Support (B),* 3–73
dequeuing from UCB • *Device Support (B),* 1–38
device-independent portion of • *Device Support (A),* 4–9 to 4–10
insertion in pending-I/O queue • *Device Support (A),* 2–4, 4–13, 7–4, 8–1; *Device Support (B),* 3–27, 3–28
insertion in postprocessing queue • *Device Support (A),* 2–7
removal from pending-I/O queue • *Device Support (A),* 2–7, 4–13, 10–3
size • *Device Support (B),* 1–37
storing data in • *Device Support (A),* 5–2, E–16
unlocking buffers specified in • *Device Support (B),* 3–109
.IRPC directive • *MACRO,* 6–49
.IRP directive • *MACRO,* 6–47
IRPE (I/O request packet extension) • *Device Support (B),* 1–40, 1–42 to 1–44, 3–72
address • *Device Support (B),* 1–42
allocating • *Device Support (B),* 1–42
deallocation • *Device Support (B),* 1–43, 3–73, 3–109
unlocking buffers specified in • *Device Support (B),* 3–73, 3–109
IRP lookaside list
displaying contents • *System Dump Analyzer,* SDA–118
/IRP qualifier • *System Dump Analyzer,* SDA–118
ISD (image section descriptor) • *Linker,* 2–11
in GSMATCH processing • *Linker,* 3–7
"is_managed" string constant parameter to GET_ INFO • *VAXTPU,* 7–214
"is_subclass" string constant parameter to GET_ INFO • *VAXTPU,* 7–214
Item list • *RMS,* 18–1
guidelines for supplying • *RMS,* 18–1
with ACL Editor routine • *Utility Routines,* ACL–3
with TPU routines • *Utility Routines,* TPU–49
Item list address field
See XAB$L_ITEMLIST field
See XAB$L_ITMLST field
Item list extended address block
See XABITM block
Item list length field
See XAB$W_ITMLST_LEN field

Itemlist read operations • *I/O User's I*, 8–29
item_list_2 data type • *Routines Intro*, A–8t
item_list_3 data type • *Routines Intro*, A–8t
item_list_pair data type • *Routines Intro*, A–9t
item_quota_list data type • *Routines Intro*, A–9t

J

Jacket routine • *RTL Library*, 2–1
JFB (journaling file block) • *System Dump Analyzer*, SDA–77
JIB$L_BYTCNT • *Device Support (A)*, 3–13, 7–6, 7–8, E–5; *Device Support (B)*, 3–12, 3–18, 3–20, 3–22
JIB$L_BYTLM • *Device Support (A)*, 3–13, E–5; *Device Support (B)*, 3–12, 3–18, 3–20, 3–22
JIB$V_BYTCNT_WAITERS • *Device Support (B)*, 3–18
JIB (job information block) • *System Dump Analyzer*, SDA–128; *Device Support (A)*, 3–13
JIB spin lock • *Device Support (A)*, 3–13; *Device Support (B)*, 3–18, 3–20, 3–23
JMP (Jump) instruction • *MACRO*, 9–58
Job
  getting information about
    asynchronously • *System Services*, SYS–222, SYS–257
    synchronously • *System Services*, SYS–238, SYS–297
Job attached bit
  See UCB$V_JOB
JOB command
  in card reader batch job • *I/O User's I*, 2–2
Job controller • *Device Support (B)*, 1–78
  function • *Utility Routines*, PSM–4
  major interface
    asynchronous • *System Services*, SYS–441
    synchronous • *System Services*, SYS–493
  request to symbiont • *Utility Routines*, SMB–5
  sending a message to • *Device Support (A)*, 9–7 to 9–8; *Device Support (B)*, 3–53, 3–61
Job information block
  See JIB
Job logical name table • *System Services Intro*, 6–5
Job quota • *Device Support (A)*, E–5
  byte count • *Device Support (A)*, 2–3, 3–13; *Device Support (B)*, 3–12, 3–18, 3–20 to 3–21, 3–22 to 3–23
  byte limit • *Device Support (A)*, 3–13; *Device Support (B)*, 3–12, 3–18, 3–20 to 3–21, 3–22 to 3–23
/JOURNAL command qualifier • *VAXTPU*, 1–11, 1–12
Journal file • *Patch*, PAT–6; *VAXTPU*, 7–307
  default name • *VAXTPU*, 1–12
  getting characteristics of • *VAXTPU*, 7–203
  getting name of • *VAXTPU*, 1–12, 5–11
  recovering buffer contents • *VAXTPU*, 7–307
  security caution • *VAXTPU*, 1–12, 7–59, 7–234, 7–235, 7–406
Journaling
  buffer change • *VAXTPU*, 1–11
  converting buffer to journal file name • *VAXTPU*, 7–172
  default file name • *VAXTPU*, 1–12
  EVE default behavior • *VAXTPU*, 1–12
  getting file name of buffer change journal • *VAXTPU*, 7–172
  getting journal file information • *VAXTPU*, 7–203
  keystroke
    enabling and disabling • *VAXTPU*, 7–408
  layered application control • *VAXTPU*, 1–12
  recovery of buffer contents • *VAXTPU*, 7–307
  role of source file • *VAXTPU*, 7–308
  sensing a safe buffer • *VAXTPU*, 7–175
  sensing the enable of buffer change journaling • *VAXTPU*, 1–12, 5–10
  sensing the enable of keystroke journaling • *VAXTPU*, 1–12, 5–11
  using both keystroke and buffer change journaling • *VAXTPU*, 1–12
Journaling extended address block
  See XABJNL block
Journaling file block
  See JFB
JOURNALING keyword • *VAXTPU*, 7–405
Journaling not supported
  error message • *Analyze/RMS_File*, ARMS–8
JOURNALING parameter
  SET built-in procedure • *VAXTPU*, 7–405
"journaling" string constant parameter
  GET_INFO built-in • *VAXTPU*, 1–12, 5–10
"Journaling" string constant parameter to GET_INFO • *VAXTPU*, 7–172
"Journaling_frequency" string constant parameter to GET_INFO • *VAXTPU*, 7–206
/JOURNAL qualifier • *Patch*, PAT–29; *VAXTPU*, 5–10
"journal" string constant parameter
  GET_INFO built-in • *VAXTPU*, 7–203
"Journal" string constant parameter to GET_INFO • *VAXTPU*, 7–177
JOURNAL_CLOSE built-in procedure • *VAXTPU*, 7–234

"Journal_file" GET_INFO request_string • *VAXTPU*, 7–177

"journal_file" string constant parameter
- GET_INFO built-in • *VAXTPU*, 1–12, 5–11, 7–172

"Journal_file" string constant parameter to GET_INFO • *VAXTPU*, 7–206

"journal_name" string constant parameter
- GET_INFO built-in • *VAXTPU*, 7–172

JOURNAL_OPEN built-in procedure • *VAXTPU*, 1–12, 5–11, 7–235 to 7–237
- controlling errors related to • *VAXTPU*, 7–408

JSB (Jump to Subroutine) instruction • *MACRO*, 9–59

JSB call format • *Routines Intro*, 1–4

JSB entry point • *RTL Math*, 1–2

JSB entry points • *Modular Procedures*, 2–12, A–2

/JSB qualifier • *Debugger*, 3–14, CD–132, CD–189, CD–265

K

KDA50 disk controller • *I/O User's I*, 3–3

KDB50 disk controller • *I/O User's I*, 3–3

Kernel mode
- changing to • *System Services*, SYS–66

Kernel-mode requirements • *Device Support (A)*, E–1

/KERNEL qualifier • *System Dump Analyzer*, SDA–157

Kernel stack • *Device Support (A)*, 8–1
- displaying contents • *System Dump Analyzer*, SDA–157

Kernel stack pointer • *System Dump Analyzer*, SDA–14

Key • *Librarian*, LIB–2, LIB–4, LIB–5
- See also Key map
- See also Library key
- See also Sort/Merge Utility
- alternate • *File Def Language*, FDL–5
  - duplicate values • *File Applications*, 3–22
  - performance of • *File Applications*, 3–22
- built-in procedures for defining
  - DEFINE_KEY • *VAXTPU*, 7–100
  - LAST_KEY • *VAXTPU*, 7–242
  - LOOKUP_KEY • *VAXTPU*, 7–254
  - SET (POST_KEY_PROCEDURE) • *VAXTPU*, 7–442
  - SET (PRE_KEY_PROCEDURE) • *VAXTPU*, 7–444
  - SET (SELF_INSERT) • *VAXTPU*, 7–470
  - SET (UNDEFINED_KEY) • *VAXTPU*, 7–490
  - UNDEFINE_KEY • *VAXTPU*, 7–532

Key (Cont.)
- creating a name for • *VAXTPU*, 7–238
- defining as simple or segmented • *RMS*, 13–13
- defining for SDA • *System Dump Analyzer*, SDA–43
- determining match method • *RMS*, 7–5
- duplicate values • *File Applications*, 2–20
- example of finding and deleting a record • *RMS*, 4–20
- example of updating a record • *RMS*, 4–21
- for Prolog 1 and 2 files • *File Applications*, 3–16
- length • *File Def Language*, FDL–28
- null value • *File Applications*, 2–20
- number of • *File Applications*, 3–23
- primary • *File Applications*, 3–16, 3–22
- segmented • *File Applications*, 3–16
- segment length • *File Def Language*, FDL–30
- selecting path • *RMS*, 4–12
- size • *File Applications*, 9–13, 9–15, 9–18
- size restriction for string type • *RMS*, 13–15
- type • *File Def Language*, FDL–30
- types of matches • *RMS*, 7–5
- use of to store indexed records sequentially • *File Applications*, 2–5

Key 0 • *File Applications*, 3–17

KEY attribute • *File Def Language*, FDL–2, FDL–26, FDL–40

Keyboard control character • *I/O User's I*, 8–4 to 8–6, 8–9

Key buffer • *File Applications*, 8–3, 9–13, 9–18

Key buffer address field
- See RAB$L_KBF field

Key-characteristics option • *File Applications*, 4–29

Key compression
- front • *File Applications*, 3–16
- prohibition against using • *File Applications*, 3–3, 3–16, 3–25, 4–9
- rear • *File Applications*, 3–16

Key definition
- creating • *Debugger*, 8–8, CD–50
- debugger predefined • *Debugger*, B–1
  - with DECwindows • *Debugger*, 1–31
- debugger predefined, multiprocess • *Debugger*, 10–17
- deleting • *Debugger*, 8–9, CD–59
- displaying • *Debugger*, 8–9, CD–223

Key definition extended address block
- See XABKEY block

KEY DESCRIPTOR
- how updated by CONVERT • *Convert*, CONV–11

KEY DESCRIPTOR structure • *File Applications*, 10–19

Key greater than
  See RAB$V_NXT option
Key-greater-than option
  See Next key option
Key greater than or equal
  See RAB$V_EQNXT option
Key-greater-than-or-equal option
  See Equal-or-next key option
Key line
  formatting • *Librarian*, LIB–5
Key map
  built-in procedures
    ADD_KEY_MAP • *VAXTPU*, 7–17
    CREATE_KEY_MAP • *VAXTPU*, 7–63
    REMOVE_KEY_MAP • *VAXTPU*, 7–313
    SHOW (KEY_MAP) • *VAXTPU*, 7–505
    SHOW (KEY_MAPS) • *VAXTPU*, 7–505
Key map list
  See also Key
  built-in procedures
    CREATE_KEY_MAP_LIST • *VAXTPU*, 7–65
    SET (KEY_MAP_LIST) • *VAXTPU*, 7–410
    SHOW (KEY_MAP_LIST) • *VAXTPU*, 7–505
    SHOW (KEY_MAP_LISTS) • *VAXTPU*, 7–505
  example of fetching • *VAXTPU*, B–19 to B–22
Key match
  approximate • *File Applications*, 8–11
  exact • *File Applications*, 8–11
  generic • *File Applications*, 8–11
  generic and approximate • *File Applications*, 8–12
Key name
  character restrictions in • *Librarian*, LIB–4
  in help libraries • *Librarian*, LIB–4 to LIB–5, LIB–9
  table • *VAXTPU*, 2–6
Key name buffer address field
  See XAB$L_KNM field
KEY NULL_VALUE attribute • *File Def Language*, FDL–29
Key number • *Librarian*, LIB–5
  See also Module
Key of reference • *File Applications*, 2–5; *Convert*, CONV–16
  establishing • *RMS*, RMS–48
Key of reference field
  See RAB$B_KRF field
  See XAB$_REF field
Key option
  comparing primary and alternate keys • *RMS*, 13–8
Key options flag field
  See XAB$B_FLG field
Keypad
  reading from • *Programming Resources*, 7–25
Keypad mode • *Debugger*, 8–8, CD–50, CD–153, CD–223, B–1
Key position field
  See XAB$W_POS0 through XAB$W_POS7 field
KEY primary attribute • *File Applications*, 4–29
  DATA_AREA secondary attribute • *File Applications*, 3–24
  DATA_FILL secondary attribute • *File Applications*, 3–26
  INDEX_AREA secondary attribute • *File Applications*, 3–24
  INDEX_FILL secondary attribute • *File Applications*, 3–26
  LEVEL1_INDEX_AREA secondary attribute • *File Applications*, 3–24
  TYPE secondary attribute • *File Applications*, 3–22
KEY PROLOG attribute • *Convert*, CONV–19; *File Def Language*, FDL–27, FDL–28
/KEY qualifier • *Convert*, CONV–16; *System Dump Analyzer*, SDA–44
Key size field
  See RAB$B_KSZ field
  See XAB$B_SIZ0 through XAB$B_SIZ7 field
Key state • *Debugger*, 8–9, CD–50, CD–223, B–1
Key string buffer
  program example • *RMS*, 4–16
Key string descriptor
  program example • *RMS*, 4–16
Key string length
  program example • *RMS*, 4–16
Keystroke journaling
  and buffer change journaling • *VAXTPU*, 7–307
  comparative to buffer change journaling • *VAXTPU*, 1–11
  enabling and disabling • *VAXTPU*, 7–408
  sensing the enable • *VAXTPU*, 1–12, 5–11
KEYSTROKE_RECOVERY keyword • *VAXTPU*, 7–408
KEYSTROKE_RECOVERY parameter
  SET built-in procedure • *VAXTPU*, 7–408
Key table
  reading from • *Programming Resources*, 7–28
Key value clause • *Command Def*, CDU–28
256 keyword
  for /FORMAT qualifier • *National Char Set*, NCS–29
Keyword • *Command Def*, CDU–2; *Librarian*, LIB–4; *File Def Language*, FDL–2; *VAXTPU*, 3–12

Keyword (Cont.)
- See also DEFINE TYPE statement
- abbreviating • *File Def Language,* FDL–40
- ALL
  - with EXPAND_NAME • *VAXTPU,* 7–135
  - with REMOVE_KEY_MAP • *VAXTPU,* 7–313
  - with SET (BELL) • *VAXTPU,* 7–355
  - with SET (DEBUG) • *VAXTPU,* 7–364
  - with UPDATE • *VAXTPU,* 7–538
- ANCHOR • *VAXTPU,* 7–24 to 7–25
  - with SEARCH • *VAXTPU,* 7–327, 7–328
  - with SEARCH_QUIETLY • *VAXTPU,* 7–332
- BELL • *VAXTPU,* 7–355
  - with SET (MESSAGE_ACTION_TYPE) • *VAXTPU,* 7–426
- BLANK_TABS • *VAXTPU,* 7–483
- BLINK
  - with SELECT • *VAXTPU,* 7–337
  - with SET (PROMPT_AREA) • *VAXTPU,* 7–446
  - with SET (STATUS_LINE) • *VAXTPU,* 7–476
  - with SET (VIDEO) • *VAXTPU,* 7–492
- BOLD
  - with SELECT • *VAXTPU,* 7–337
  - with SET (PROMPT_AREA) • *VAXTPU,* 7–446
  - with SET (STATUS_LINE) • *VAXTPU,* 7–476
  - with SET (VIDEO) • *VAXTPU,* 7–492
- BROADCAST
  - with SET (BELL) • *VAXTPU,* 7–355
- BUFFER_BEGIN
  - with POSITION • *VAXTPU,* 7–287
  - with SEARCH • *VAXTPU,* 7–327
  - with SEARCH_QUIETLY • *VAXTPU,* 7–332
- BUFFER_END
  - with POSITION • *VAXTPU,* 7–287
  - with SEARCH • *VAXTPU,* 7–327
  - with SEARCH_QUIETLY • *VAXTPU,* 7–332
- COMMENT
  - with LOOK_UP_KEY • *VAXTPU,* 7–254
- CROSS_WINDOW_BOUNDS • *VAXTPU,* 7–361
- DEBUG • *VAXTPU,* 7–362, 7–363, 7–364
- DEVICE
  - with FILE_PARSE • *VAXTPU,* 7–140
  - with FILE_SEARCH • *VAXTPU,* 7–143
- DIRECTORY
  - with FILE_PARSE • *VAXTPU,* 7–140
  - with FILE_SEARCH • *VAXTPU,* 7–143
- EOB_TEXT • *VAXTPU,* 7–374
- EXACT
  - with LEARN_BEGIN • *VAXTPU,* 7–244
  - with SEARCH • *VAXTPU,* 7–328
  - with SEARCH_QUIETLY • *VAXTPU,* 7–333

Keyword (Cont.)
- FACILITY_NAME • *VAXTPU,* 7–378
- for /FORMAT qualifier • *National Char Set,* NCS–29
- FORWARD • *VAXTPU,* 7–85, 7–379
  - with SEARCH • *VAXTPU,* 7–328
  - with SEARCH_QUIETLY • *VAXTPU,* 7–333
- GRAPHIC_TABS • *VAXTPU,* 7–483
- how to define • *Command Def,* CDU–7 to CDU–8, CDU–30
- INFORMATIONAL • *VAXTPU,* 7–397
- in keyword table • *RTL Library,* LIB–261
- INSERT • *VAXTPU,* 7–404
- JOURNALING • *VAXTPU,* 7–405
- key name • *VAXTPU,* 2–6
- KEYSTROKE_RECOVERY • *VAXTPU,* 7–408
- KEYWORDS
  - with EXPAND_NAME • *VAXTPU,* 7–135
- KEY_MAP
  - with LOOK_UP_KEY • *VAXTPU,* 7–254
- KEY_MAP_LIST • *VAXTPU,* 7–410
- LEFT_MARGIN • *VAXTPU,* 7–412
- LEFT_MARGIN_ACTION • *VAXTPU,* 7–414
- lexical • *VAXTPU,* 3–36
- LINE_BEGIN • *VAXTPU,* 7–249 to 7–250
  - with POSITION • *VAXTPU,* 7–288
  - with SEARCH • *VAXTPU,* 7–327
  - with SEARCH_QUIETLY • *VAXTPU,* 7–332
- LINE_END • *VAXTPU,* 7–251
  - with POSITION • *VAXTPU,* 7–288
  - with SEARCH • *VAXTPU,* 7–327
  - with SEARCH_QUIETLY • *VAXTPU,* 7–332
- LINE_NUMBER • *VAXTPU,* 7–416
- MARGINS • *VAXTPU,* 7–419
- MAX_LINES • *VAXTPU,* 7–421
- MESSAGE_FLAGS • *VAXTPU,* 7–427
- MODIFIABLE • *VAXTPU,* 7–429
- MOUSE
  - with POSITION • *VAXTPU,* 7–288, 7–289
- NAME
  - with FILE_PARSE • *VAXTPU,* 7–141
  - with FILE_SEARCH • *VAXTPU,* 7–144
- NODE
  - with FILE_PARSE • *VAXTPU,* 7–140
  - with FILE_SEARCH • *VAXTPU,* 7–143
- NONE
  - with SELECT • *VAXTPU,* 7–337
  - with SET (MESSAGE_ACTION_TYPE) • *VAXTPU,* 7–426
  - with SET (PROMPT_AREA) • *VAXTPU,* 7–446
  - with SET (STATUS_LINE) • *VAXTPU,* 7–476
  - with SET (VIDEO) • *VAXTPU,* 7–492

Keyword (Cont.)
NO_EXACT
with LEARN_BEGIN • *VAXTPU,* 7–244
with SEARCH • *VAXTPU,* 7–328
with SEARCH_QUIETLY • *VAXTPU,* 7–333
NO_TRANSLATE • *VAXTPU,* 7–483
NO_WRITE • *VAXTPU,* 7–434
occluded • *VAXTPU,* 3–12
OFF
with CREATE_WINDOW • *VAXTPU,* 7–77
with HELP_TEXT • *VAXTPU,* 7–228
with QUIT • *VAXTPU,* 7–291
with SET (AUTO_REPEAT) • *VAXTPU,* 7–353
with SET (BELL) • *VAXTPU,* 7–355
with SET (COLUMN_MOVE_VERTICAL) • *VAXTPU,* 7–359
with SET (CROSS_WINDOW_BOUNDS) • *VAXTPU,* 7–361
with SET (DEBUG) • *VAXTPU,* 7–363, 7–364
with SET (INFORMATIONAL) • *VAXTPU,* 7–397
with SET (LINE_NUMBER) • *VAXTPU,* 7–416
with SET (MODIFIABLE) • *VAXTPU,* 7–429
with SET (MOUSE) • *VAXTPU,* 7–432
with SET (NO_WRITE) • *VAXTPU,* 7–434
with SET (PAD) • *VAXTPU,* 7–437
with SET (PAD_OVERSTRUCK_TABS) • *VAXTPU,* 7–439
with SET (SCREEN_UPDATE) • *VAXTPU,* 7–460
with SET (SCROLLING) • *VAXTPU,* 7–467
with SET (SELF_INSERT) • *VAXTPU,* 7–470
with SET (SUCCESS) • *VAXTPU,* 7–479
with SET (TIMER) • *VAXTPU,* 7–486
with SET (TRACEBACK) • *VAXTPU,* 7–488
with SPAWN • *VAXTPU,* 7–515
ON
with CREATE WINDOW • *VAXTPU,* 7–77
with CREATE_WINDOW • *VAXTPU,* 7–77
with HELP_TEXT • *VAXTPU,* 7–228
with QUIT • *VAXTPU,* 7–291
with SET (AUTO_REPEAT) • *VAXTPU,* 7–353
with SET (BELL) • *VAXTPU,* 7–355
with SET (COLUMN_MOVE_VERTICAL) • *VAXTPU,* 7–359
with SET (CROSS_WINDOW_BOUNDS) • *VAXTPU,* 7–361
with SET (DEBUG) • *VAXTPU,* 7–363
with SET (INFORMATIONAL) • *VAXTPU,* 7–397
with SET (LINE_NUMBER) • *VAXTPU,* 7–416
with SET (MODIFIABLE) • *VAXTPU,* 7–429

Keyword
ON (Cont.)
with SET (MOUSE) • *VAXTPU,* 7–432
with SET (NO_WRITE) • *VAXTPU,* 7–434
with SET (PAD) • *VAXTPU,* 7–437
with SET (PAD_OVERSTRUCK_TABS) • *VAXTPU,* 7–439
with SET (SCREEN_UPDATE) • *VAXTPU,* 7–460
with SET (SCROLLING) • *VAXTPU,* 7–467
with SET (SELF_INSERT) • *VAXTPU,* 7–470
with SET (SUCCESS) • *VAXTPU,* 7–479
with SET (TIMER) • *VAXTPU,* 7–486
with SET (TRACEBACK) • *VAXTPU,* 7–488
with SPAWN • *VAXTPU,* 7–515
OUTPUT_FILE • *VAXTPU,* 7–435
OVERSTRIKE • *VAXTPU,* 7–436
PAD • *VAXTPU,* 7–437
PAD_OVERSTRUCK_TABS • *VAXTPU,* 7–439
PAGE BREAK • *VAXTPU,* 7–286
PAGE_BREAK
with SEARCH • *VAXTPU,* 7–327
with SEARCH_QUIETLY • *VAXTPU,* 7–332
PERMANENT • *VAXTPU,* 7–441
POST_KEY_PROCEDURE • *VAXTPU,* 7–442
PROCEDURES
with EXPAND_NAME • *VAXTPU,* 7–135
PROGRAM • *VAXTPU,* 7–362
with LOOK_UP_KEY • *VAXTPU,* 7–254
PROMPT_AREA • *VAXTPU,* 7–446
REMAIN • *VAXTPU,* 7–312
with SEARCH • *VAXTPU,* 7–327
with SEARCH_QUIETLY • *VAXTPU,* 7–332
returned by CURRENT_DIRECTION • *VAXTPU,* 7–85
returned by READ_KEY • *VAXTPU,* 7–301
REVERSE • *VAXTPU,* 7–85, 7–453
with SEARCH • *VAXTPU,* 7–328
with SEARCH_QUIETLY • *VAXTPU,* 7–333
with SELECT • *VAXTPU,* 7–337
with SET (MESSAGE_ACTION_TYPE) • *VAXTPU,* 7–426
with SET (PROMPT_AREA) • *VAXTPU,* 7–446
with SET (STATUS_LINE) • *VAXTPU,* 7–476
with SET (VIDEO) • *VAXTPU,* 7–492
RIGHT_MARGIN • *VAXTPU,* 7–454
RIGHT_MARGIN_ACTION • *VAXTPU,* 7–456
SCREEN_UPDATE • *VAXTPU,* 7–460
SCROLLING • *VAXTPU,* 7–467
SELF_INSERT • *VAXTPU,* 7–470
SHIFT_KEY • *VAXTPU,* 7–472

- Keyword
  - SPECIAL_GRAPHICS
    - with SET (STATUS_LINE) • *VAXTPU,* 7–476
  - STATUS_LINE • *VAXTPU,* 7–476
  - SUCCESS • *VAXTPU,* 7–479
  - SYSTEM • *VAXTPU,* 7–480
  - TEXT • *VAXTPU,* 7–483
  - TIMER • *VAXTPU,* 7–486
  - TRACEBACK • *VAXTPU,* 7–488
  - TYPE
    - with FILE_PARSE • *VAXTPU,* 7–141
    - with FILE_SEARCH • *VAXTPU,* 7–144
  - UNANCHOR • *VAXTPU,* 7–530 to 7–531
    - with SEARCH_QUIETLY • *VAXTPU,* 7–333
  - UNDEFINED_KEY • *VAXTPU,* 7–490
  - UNDERLINE
    - with SELECT • *VAXTPU,* 7–337
    - with SET (PROMPT_AREA) • *VAXTPU,* 7–446
    - with SET (STATUS_LINE) • *VAXTPU,* 7–476
    - with SET (VIDEO) • *VAXTPU,* 7–492
  - VARIABLES
    - with EXPAND_NAME • *VAXTPU,* 7–135
  - VERSION
    - with FILE_PARSE • *VAXTPU,* 7–141
    - with FILE_SEARCH • *VAXTPU,* 7–144
  - VIDEO • *VAXTPU,* 7–492
  - with SET • *VAXTPU,* 7–347 to 7–348
  - with SHOW • *VAXTPU,* 7–505 to 7–506
- Keyword argument • *MACRO,* 4–3
- Keyword clause
  - types used in collating sequence expression • *National Char Set,* NCS–13
  - types used in conversion function expressions • *National Char Set,* NCS–15
- Keyword constants • *VAXTPU,* 3–5
- KEYWORD data type • *VAXTPU,* 2–5 to 2–7
- Keyword path • *Command Def,* CDU–11
  - obtaining values of command string keywords • *Utility Routines,* CLI–10
  - referencing command string keywords • *Utility Routines,* CLI–13
- KEYWORDS keyword
  - with EXPAND_NAME • *VAXTPU,* 7–135
- KEY_GREATER_EQUAL attribute • *File Def Language,* FDL–10
- KEY_GREATER_EQUAL secondary attribute • *File Applications,* 8–9
- KEY_GREATER_THAN attribute • *File Def Language,* FDL–10
- KEY_GREATER_THAN secondary attribute • *File Applications,* 8–9, 8–10
- KEY_LIMIT attribute • *File Def Language,* FDL–11
- KEY_MAP keyword
  - with LOOK_UP_KEY • *VAXTPU,* 7–254
- KEY_MAP_LIST keyword • *VAXTPU,* 7–410
- "Key_map_list" string constant parameter to GET_INFO • *VAXTPU,* 7–172
- KEY_NAME built-in procedure • *VAXTPU,* 7–238 to 7–241
- KEY_NCMPR option • *File Def Language,* FDL–27
- KEY_OF_REFERENCE attribute • *File Def Language,* FDL–11
- "Key_type" string constant parameter to GET_INFO • *VAXTPU,* 7–162
- KFQSA adapter • *I/O User's I,* 3–5
- KGE option • *File Def Language,* FDL–10, FDL–11
- KILL_SELECTION client message • *VAXTPU,* 7–344
- Known file list
  - image lookup • *File Applications,* 5–5
- KSP symbol • *System Dump Analyzer,* SDA–14

L

- Label
  - created local • *MACRO,* 4–7
  - global • *MACRO,* 2–2
  - user-defined local • *MACRO,* 3–7, 4–7
- %LABEL • *Debugger,* 3–11, D–6
- LABEL clause
  - for DEFINE TYPE statement • *Command Def,* CDU–28
  - for PARAMETER clause • *Command Def,* CDU–23, CDU–32
  - for QUALIFIER clause • *Command Def,* CDU–25, CDU–34
- Label descriptor • *Routines Intro,* 2–29
- Laboratory Peripheral Accelerator
  - See LPA11-K device
- Language
  - current • *Debugger,* 4–10, CD–145
  - identifying • *Debugger,* CD–226
  - multilanguage program • *Debugger,* 9–7
    - with DECwindows • *Debugger,* 1–29
  - native to VMS • *File Def Language,* FDL–41
  - setting • *Debugger,* 4–10, CD–145
  - support by debugger • *Debugger,* E–1
    - with DECwindows • *Debugger,* 1–2
- Language expression
  - compared to address expression • *Debugger,* 4–8
    - with DECwindows • *Debugger,* 1–23
  - DEPOSIT command • *Debugger,* 4–3, CD–61
  - EVALUATE command • *Debugger,* 4–5, CD–81

Language expression (Cont.)
  evaluating • *Debugger*, 4–5
    with DECwindows • *Debugger*, 1–27
  FOR command • *Debugger*, 8–10, CD–103
  IF command • *Debugger*, 8–10, CD–109
  REPEAT command • *Debugger*, 8–11, CD–115
  WHEN clause • *Debugger*, 3–14
  WHILE command • *Debugger*, 8–11, CD–277
Language extension • *Routines Intro*, 2–6
Language independence
  testing for • *Modular Procedures*, 4–1, 4–4
Language-Sensitive Editor • *Modular Procedures*, 1–12; *Debugger*, CD–78
Language support procedure • *Routines Intro*, 2–4
Large request packet
  See LRP
"last" string parameter to ADD_KEY_MAP • *VAXTPU*, 7–17
Last-chance exception vector • *Programming Resources*, 9–13
Last-chance handler • *Debugger*, 9–13
"Last" string constant parameter to GET_INFO • *VAXTPU*, 7–166, 7–167, 7–169, 7–181, 7–183, 7–184, 7–191, 7–218
LAST_KEY built-in procedure • *VAXTPU*, 7–242
LAT port driver (LTDRIVER) • *I/O User's I*, 8–1
LBR$CLOSE • *Programming Resources*, 8–36
LBR$CLOSE routine • *Utility Routines*, LBR–20
LBR$DELETE_DATA • *Programming Resources*, 8–42
LBR$DELETE_DATA routine • *Utility Routines*, LBR–21
LBR$DELETE_KEY • *Programming Resources*, 8–42
LBR$DELETE_KEY routine • *Utility Routines*, LBR–23
LBR$FIND routine • *Utility Routines*, LBR–25
LBR$FLUSH routine • *Utility Routines*, LBR–27
LBR$GET_HEADER • *Programming Resources*, 8–50
LBR$GET_HEADER routine • *Utility Routines*, LBR–29
LBR$GET_HELP routine • *Utility Routines*, LBR–31
LBR$GET_HISTORY routine • *Utility Routines*, LBR–34
LBR$GET_INDEX • *Programming Resources*, 8–53
LBR$GET_INDEX routine • *Utility Routines*, LBR–36
LBR$GET_RECORD • *Programming Resources*, 8–43
LBR$GET_RECORD routine • *Utility Routines*, LBR–38
LBR$INI_CONTROL • *Programming Resources*, 8–36; *Utility Routines*, LBR–40
LBR$INSERT_KEY • *Programming Resources*, 8–40
LBR$INSERT_KEY routine • *Utility Routines*, LBR–42
LBR$LOOKUP_KEY • *Programming Resources*, 8–40, 8–42, 8–43, 8–48
LBR$LOOKUP_KEY routine • *Utility Routines*, LBR–44
LBR$OPEN • *Programming Resources*, 8–36
LBR$OPEN routine • *Utility Routines*, LBR–46
LBR$OUTPUT_HELP • *Programming Resources*, 8–52
LBR$OUTPUT_HELP routine • *Utility Routines*, LBR–50
LBR$PUT_END • *Programming Resources*, 8–40
LBR$PUT_END routine • *Utility Routines*, LBR–55
LBR$PUT_HISTORY routine • *Utility Routines*, LBR–56
LBR$PUT_RECORD • *Programming Resources*, 8–40
LBR$PUT_RECORD routine • *Utility Routines*, LBR–58
LBR$REPLACE_KEY • *Programming Resources*, 8–40
LBR$REPLACE_KEY routine • *Utility Routines*, LBR–60
LBR$RET_RMSSTV routine • *Utility Routines*, LBR–62
LBR$SEARCH routine • *Utility Routines*, LBR–63
LBR$SET_INDEX routine • *Utility Routines*, LBR–65
LBR$SET_LOCATE routine • *Utility Routines*, LBR–67
LBR$SET_MODULE • *Programming Resources*, 8–48
LBR$SET_MODULE routine • *Utility Routines*, LBR–68
LBR$SET_MOVE routine • *Utility Routines*, LBR–70
LBR$_KEYNOTFND • *Programming Resources*, 8–40
LBR routines
  control index • *Utility Routines*, LBR–7
  current index number
    setting • *Utility Routines*, LBR–65
  data record
    reading • *Utility Routines*, LBR–38
    writing • *Utility Routines*, LBR–58
  end-of-module record
    writing • *Utility Routines*, LBR–55
  examples • *Utility Routines*, LBR–7 to LBR–19
    creating a new library • *Utility Routines*, LBR–7 to LBR–10
    deleting a module from a library • *Utility Routines*, LBR–16 to LBR–19
    extracting a module from a library • *Utility Routines*, LBR–14 to LBR–16
    inserting a module into a library • *Utility Routines*, LBR–10 to LBR–14
  header • *Utility Routines*, LBR–2

- LBR routines
  - help text
    - outputting • *Utility Routines*, LBR–50
    - retrieving • *Utility Routines*, LBR–31
  - index • *Utility Routines*, LBR–2
    - searching • *Utility Routines*, LBR–63
  - introduction • *Utility Routines*, LBR–1 to LBR–19
  - library
    - closing • *Utility Routines*, LBR–20
    - creating • *Utility Routines*, LBR–46
    - opening • *Utility Routines*, LBR–46
    - shareable image • *Utility Routines*, LBR–1
    - structure • *Utility Routines*, LBR–2 to LBR–5
    - types • *Utility Routines*, LBR–1
    - user-developed • *Utility Routines*, LBR–1
  - library file
    - flushing • *Utility Routines*, LBR–27
  - library header information
    - reading • *Utility Routines*, LBR–29
    - retrieving • *Utility Routines*, LBR–29
  - library index
    - getting contents • *Utility Routines*, LBR–36
    - initializing • *Utility Routines*, LBR–40
    - searching for key • *Utility Routines*, LBR–36
  - library key • *Utility Routines*, LBR–2
    - creating ASCII or binary • *Utility Routines*, LBR–47
    - deleting • *Utility Routines*, LBR–23
    - finding • *Utility Routines*, LBR–25
    - inserting • *Utility Routines*, LBR–42
    - looking up • *Utility Routines*, LBR–44
    - replacing • *Utility Routines*, LBR–60
  - library update history record
    - retrieving • *Utility Routines*, LBR–34
  - locate mode
    - setting record access mode to • *Utility Routines*, LBR–67
  - module • *Utility Routines*, LBR–2
    - accessing with RFA • *Utility Routines*, LBR–25
    - deleting data records • *Utility Routines*, LBR–21
    - deleting header • *Utility Routines*, LBR–21
  - module header
    - reading • *Utility Routines*, LBR–68
    - setting • *Utility Routines*, LBR–68
    - updating • *Utility Routines*, LBR–68
  - move mode
    - setting record access to • *Utility Routines*, LBR–70
  - summary • *Utility Routines*, LBR–5 to LBR–6
  - update history records
    - writing • *Utility Routines*, LBR–56
- LBR routines (Cont.)
  - virtual memory
    - recovering • *Utility Routines*, LBR–27
  - VMS RMS status value
    - returning • *Utility Routines*, LBR–62
- ;L command • *Delta/XDelta*, DELTA–44
  - privileges required for • *Delta/XDelta*, DELTA–14
- LDPCTX (Load Process Context) instruction • *MACRO*, 9–193, 10–47
- LDR$ALLOC_PT • *Device Support (A)*, 16–18, E–7; *Device Support (B)*, 3–107
- LDR$DEALLOC_PT • *Device Support (B)*, 3–108
- LDR$GL_FREE_PT • *Device Support (B)*, 3–107, 3–108
- LDR$GL_SPTBASE • *Device Support (B)*, 3–107, 3–108
- Leading separate numeric string
  - data type • *MACRO*, 8–11
- LEARN data type • *VAXTPU*, 2–7 to 2–8
- LEARN_ABORT built-in procedure • *VAXTPU*, 7–243
- LEARN_BEGIN built-in procedure • *VAXTPU*, 7–244 to 7–246
- LEARN_END built-in procedure • *VAXTPU*, 7–244 to 7–246
- Left margin
  - setting records • *VAXTPU*, 7–448
- /LEFT qualifier • *Debugger*, CD–98, CD–110, CD–118
- LEFT_MARGIN keyword • *VAXTPU*, 7–412
- "Left_margin" string constant parameter to GET_INFO • *VAXTPU*, 7–172, 7–186
- LEFT_MARGIN_ACTION keyword • *VAXTPU*, 7–414
- "Left_margin_action" string constant parameter to GET_INFO • *VAXTPU*, 7–172
- Legal function bit mask • *Device Support (A)*, 4–11
- LENGTH attribute • *File Def Language*, FDL–28, FDL–29
- LENGTH built-in procedure • *VAXTPU*, 7–247 to 7–248
- Length field
  - using to indicate constant (keyword) value • *RMS*, 2–4
  - using to indicate mask or bit offset • *RMS*, 2–3
- Length modes • *Patch*, PAT–16
  - See also Entry and display modes
- Length of key segment • *File Def Language*, FDL–30
- %LENGTH operator • *MACRO*, 4–8
- LEQUAL keyword
  - with GSMATCH option • *Programming Resources*, 5–5
- Level
  - number of • *File Applications*, A–2

LEVEL1_INDEX_AREA attribute • *File Def Language*, FDL–27, FDL–28

LEVEL1_INDEX_AREA secondary attribute • *File Applications*, 3–24

LEVEL1_RECORD_COUNT attribute • *File Def Language*, FDL–5

Level of prompting • *File Def Language*, FDL–55

Level of root bucket field

  See XAB$B_LVL field

Levels of abstraction • *Modular Procedures*, 2–2

Lexical element • *VAXTPU*, 3–1

Lexical function

  See Built-in symbol

  F$SEARCH • *Device Support (A)*, 13–24

Lexical keywords • *VAXTPU*, 3–36 to 3–38

LF character • *File Def Language*, FDL–35

LIB$ADAWI • *RTL Library*, LIB–3

LIB$ADDX • *Programming Resources*, 3–24; *RTL Library*, LIB–7

LIB$ADD_TIME • *Programming Resources*, 3–24

LIB$ADD_TIMES • *RTL Library*, LIB–5

LIB$ANALYZE_SDESC • *RTL Library*, LIB–10; *RTL String Manipulation*, 2–4

LIB$ASN_WTH_MBX • *RTL Library*, 2–23, LIB–12

LIB$AST_IN_PROG • *RTL Library*, 2–22, LIB–15

LIB$ATTACH • *RTL Library*, 2–9, LIB–17

LIB$BBCCI • *RTL Library*, LIB–19

LIB$BBSSI • *RTL Library*, LIB–21

LIB$CALLG • *RTL Library*, 2–16, LIB–23

LIB$CHAR • *RTL Library*, LIB–25

LIB$CONVERT_DATE_STRING • *RTL Library*, LIB–27

LIB$CRC • *RTL Library*, 2–16, LIB–31

LIB$CRC_TABLE • *RTL Library*, 2–16, LIB–33

LIB$CREATE_DIR • *RTL Library*, 2–24, LIB–36

LIB$CREATE_USER_VM_ZONE • *RTL Library*, 5–12, 5–17, LIB–40

LIB$CREATE_VM_ZONE • *Programming Resources*, 10–1; *RTL Library*, 5–6, 5–16, LIB–44

LIB$CRF_INS_KEY • *RTL Library*, 8–1, LIB–50

LIB$CRF_INS_REF • *RTL Library*, 8–1, LIB–52

LIB$CRF_OUTPUT • *RTL Library*, 8–1, LIB–55

LIB$CURRENCY • *RTL Library*, LIB–59

LIB$CVTF_FROM_INTERNAL_TIME • *RTL Library*, LIB–70

LIB$CVTF_TO_INTERNAL_TIME • *RTL Library*, LIB–74

LIB$CVT_DTB • *RTL Library*, LIB–76

LIB$CVT_DX_DX • *RTL Library*, LIB–61

LIB$CVT_FROM_INTERNAL_TIME • *RTL Library*, LIB–67

LIB$CVT_HTB • *RTL Library*, LIB–76

LIB$CVT_OTB • *RTL Library*, LIB–76

LIB$CVT_TO_INTERNAL_TIME • *RTL Library*, LIB–72

LIB$CVT_VECTIM • *RTL Library*, LIB–78

LIB$DATE_TIME • *Programming Resources*, 3–23; *RTL Library*, LIB–80

LIB$DAY • *Programming Resources*, 3–25; *RTL Library*, LIB–82

LIB$DAY_OF_WEEK • *RTL Library*, LIB–84

LIB$DECODE_FAULT • *RTL Library*, 4–30, LIB–86

LIB$DEC_OVER • *Programming Resources*, 9–26; *RTL Library*, 4–32, LIB–104

LIB$DELETE_FILE • *RTL Library*, LIB–106

LIB$DELETE_LOGICAL • *RTL Library*, 2–8, LIB–114

LIB$DELETE_SYMBOL • *RTL Library*, 2–8, LIB–116

LIB$DELETE_VM_ZONE • *RTL Library*, 5–6, LIB–118

LIB$DIGIT_SEP • *RTL Library*, LIB–120

LIB$DISABLE_CTRL • *RTL Library*, 2–9, LIB–122

LIB$DO_COMMAND • *RTL Library*, 2–6, LIB–124

LIB$EDIV • *RTL Library*, LIB–126

LIB$EMODD • *RTL Library*, LIB–128

LIB$EMODF • *RTL Library*, LIB–130

LIB$EMODG • *RTL Library*, LIB–132

LIB$EMODH • *RTL Library*, LIB–134

LIB$EMUL • *RTL Library*, LIB–136

LIB$ENABLE_CTRL • *RTL Library*, 2–9, LIB–138

LIB$ESTABLISH • *RTL Library*, 4–3, 4–13, 4–20, LIB–140

LIB$EXTV • *RTL Library*, LIB–142

LIB$EXTZV • *RTL Library*, LIB–145

LIB$FFC • *RTL Library*, LIB–147

LIB$FFS • *RTL Library*, LIB–147

LIB$FID_TO_NAME • *RTL Library*, LIB–149

LIB$FILE_SCAN • *RTL Library*, LIB–151

LIB$FILE_SCAN_END • *RTL Library*, LIB–153

LIB$FIND_FILE routine • *RTL Library*, LIB–155; *File Applications*, 5–8 to 5–12

LIB$FIND_FILE_END • *RTL Library*, LIB–159

LIB$FIND_IMAGE_SYMBOL • *RTL Library*, LIB–160

LIB$FIND_VM_ZONE • *RTL Library*, 5–6, LIB–163

LIB$FIXUP_FLT • *RTL Library*, 4–30, LIB–165

LIB$FLT_UNDER • *Programming Resources*, 9–26; *RTL Intro*, 3–7; *RTL Library*, 4–32, LIB–167

LIB$FORMAT_DATE_TIME • *RTL Library*, LIB–169

LIB$FREE_DATE_TIME_CONTEXT • *RTL Library*, LIB–172

LIB$FREE_EF • *RTL Library*, LIB–174

LIB$FREE_LUN • *RTL Library*, LIB–175

LIB$FREE_TIMER • *Programming Resources*, 3–21; *RTL Library*, LIB–176

LIB$FREE_VM • *RTL Library*, 5–3, LIB–177

LIB$FREE_VM_PAGE • *RTL Library*, 5–3, LIB–179

LIB$GETDVI • *RTL Library*, LIB–181
LIB$GETJPI • *RTL Library*, LIB–186
LIB$GETQUI • *Programming Resources*, 3–22; *RTL Library*, LIB–191
LIB$GETSYI • *RTL Library*, LIB–196
LIB$GET_COMMAND • *RTL Library*, LIB–199
LIB$GET_COMMON • *RTL Library*, 2–5, 2–35, LIB–202
LIB$GET_DATE_FORMAT • *RTL Library*, LIB–204
LIB$GET_EF • *RTL Library*, LIB–206
LIB$GET_FOREIGN • *RTL Library*, 2–3, LIB–208
LIB$GET_INPUT • *Programming Resources*, 7–3; *RTL Intro*, 3–3; *RTL Library*, LIB–212; *RTL String Manipulation*, 2–8
 example • *Programming Resources*, 7–4; *RMS*, 4–12
 obtaining several lines of input with • *Programming Resources*, 7–5
 obtaining single line of input with • *Programming Resources*, 7–4
 prompt • *Programming Resources*, 7–4
LIB$GET_LUN • *Programming Resources*, 7–3; *RTL Library*, LIB–215
LIB$GET_MAXIMUM_DATE_LENGTH • *RTL Library*, LIB–216
LIB$GET_SYMBOL • *RTL Library*, 2–8, LIB–219
LIB$GET_USERS_LANGUAGE • *RTL Library*, LIB–222
LIB$GET_VM • *RTL Library*, 5–3, LIB–223; *RTL String Manipulation*, 2–3
LIB$GET_VM_PAGE • *Programming Resources*, 10–1; *RTL Library*, 5–3, LIB–225
LIB$ICHAR • *RTL Library*, LIB–227
LIB$INDEX • *RTL Library*, LIB–229
LIB$INITIALIZE • *Modular Procedures*, 3–17; *Debugger*, 9–10; *RTL Library*, 7–1
 See also Initialization
LIB$INIT_DATE_TIME_CONTEXT • *RTL Library*, LIB–231
LIB$INIT_TIMER • *Programming Resources*, 3–20; *RTL Library*, LIB–235
LIB$INSERT_KEY • *Programming Resources*, 8–45
LIB$INSERT_TREE • *RTL Library*, 2–31, LIB–237
LIB$INSQHI • *RTL Library*, LIB–248
LIB$INSQTI • *RTL Library*, LIB–251
LIB$INSV • *RTL Library*, LIB–253
LIB$INT_OVER • *Programming Resources*, 9–26; *RTL Library*, 4–32, LIB–255
LIB$LEN • *RTL Library*, LIB–257
LIB$LOCC • *RTL Library*, LIB–258
LIB$LOOKUP_KEY • *RTL Library*, LIB–261
LIB$LOOKUP_TREE • *RTL Library*, 2–31, LIB–265
LIB$LP_LINES • *RTL Library*, LIB–267
LIB$MATCHC • *RTL Library*, LIB–270
LIB$MATCH_COND • *Programming Resources*, 9–16; *RTL Library*, 4–10, 4–30, LIB–272
LIB$MOVC3 • *RTL Library*, LIB–275
LIB$MOVC5 • *RTL Library*, LIB–276
LIB$MOVTC • *RTL Library*, LIB–278
LIB$MOVTUC • *RTL Library*, LIB–295
LIB$MULTF_DELTA_TIME • *RTL Library*, LIB–298
LIB$MULT_DELTA_TIME • *Programming Resources*, 3–24; *RTL Library*, LIB–297
LIB$PAUSE • *RTL Library*, LIB–299
LIB$POLYD • *RTL Library*, LIB–300
LIB$POLYF • *RTL Library*, LIB–302
LIB$POLYG • *RTL Library*, LIB–305
LIB$POLYH • *RTL Library*, LIB–307
LIB$PUT_COMMON • *RTL Library*, 2–5, 2–35, LIB–309
LIB$PUT_OUTPUT • *Programming Resources*, 7–3; *RTL Library*, LIB–311
 example • *Programming Resources*, 7–7; *RMS*, 4–12
 writing simple output with • *Programming Resources*, 7–6
LIB$RADIX_POINT • *RTL Library*, LIB–313
LIB$REMQHI • *RTL Library*, LIB–315
LIB$REMQTI • *RTL Library*, LIB–317
LIB$RENAME_FILE • *RTL Library*, LIB–319
LIB$RESERVE_EF • *RTL Library*, LIB–327
LIB$RESET_VM_ZONE • *RTL Library*, 5–13, 5–14, LIB–329
LIB$REVERT • *RTL Library*, 4–3, 4–20, LIB–331
LIB$RUN_PROGRAM • *RTL Library*, 2–5, LIB–332
LIB$SCANC • *RTL Library*, LIB–334
LIB$SCOPY_DXDX • *RTL Library*, LIB–336; *RTL String Manipulation*, 2–7
LIB$SCOPY_R_DX • *RTL Library*, LIB–338
LIB$SET_INDEX • *Programming Resources*, 8–45
LIB$SET_LOGICAL • *RTL Library*, 2–8, LIB–340
LIB$SET_SYMBOL • *RTL Library*, 2–8, LIB–343
LIB$SFREE1_DD • *RTL Library*, LIB–347
LIB$SFREEN_DD • *RTL Library*, LIB–348
LIB$SGET1_DD • *RTL Library*, LIB–350
LIB$SHOW_TIMER • *Programming Resources*, 3–20; *RTL Intro*, 3–1; *RTL Library*, LIB–352
LIB$SHOW_VM • *RTL Library*, LIB–356
LIB$SHOW_VM_ZONE • *RTL Library*, 5–6, LIB–359
LIB$SIGNAL • *RTL Intro*, 3–1; *RTL Library*, 4–2, 4–3, 4–7, 4–10, 4–11, 4–12, 4–14, 4–16, 4–22, 4–23 to 4–26, 4–31, LIB–365
 invoking • *Programming Resources*, 9–5
LIB$SIGNAL (or LIB$STOP)

LIB$SIGNAL (or LIB$STOP) (Cont.)
  using to signal errors • *RMS*, 2–6
  using to signal VMS RMS errors • *RMS*, 2–6
LIB$SIG_TO_RET • *RTL Library*, 4–29, LIB–369
  establishing • *Programming Resources*, 9–6
LIB$SIG_TO_STOP • *RTL Library*, 4–29, LIB–372
LIB$SIM_TRAP • *RTL Library*, 4–21, 4–29, LIB–374
LIB$SKPC • *RTL Library*, LIB–376
LIB$SPANC • *RTL Library*, LIB–378
LIB$SPAWN • *RTL Library*, 2–9, LIB–382
LIB$STAT_TIMER • *Programming Resources*, 3–21; *RTL Library*, LIB–388
LIB$STAT_VM • *RTL Library*, LIB–392
LIB$STOP • *RTL Library*, 4–2, 4–3, 4–4, 4–7, 4–10, 4–12, 4–14, 4–16, 4–21, 4–22, 4–23 to 4–26, LIB–394
LIB$STOP routine • *File Applications*, 5–12
LIB$SUBX • *Programming Resources*, 3–24; *RTL Library*, LIB–399
LIB$SUB_TIME • *Programming Resources*, 3–24
LIB$SUB_TIMES • *RTL Library*, LIB–397
LIB$SYS_ASCTIM • *RTL Library*, LIB–401
LIB$SYS_FAO • *RTL Library*, LIB–404
LIB$SYS_FAOL • *RTL Library*, LIB–406
LIB$SYS_GETMSG • *RTL Library*, LIB–408
LIB$TPARSE • *RTL Library*, LIB–411
LIB$TRAVERSE_TREE • *RTL Library*, 2–31, LIB–459
LIB$TRA_ASC_EBC • *RTL Library*, LIB–453
LIB$TRA_EBC_ASC • *RTL Library*, LIB–457
LIB$TRIM_FILESPEC • *RTL Library*, LIB–461
LIB$VERIFY_VM_ZONE • *RTL Library*, 5–6, LIB–464
LIB$WAIT • *RTL Library*, LIB–465
LIBRARIAN
  See Librarian Utility
Librarian routines
  See LBR routines
LIBRARIAN routines • *Librarian*, LIB–10
Librarian Utility (LIBRARIAN)
  See also LIBRARY command
  character case of library keys • *Librarian*, LIB–2
  command qualifiers • *Librarian*, LIB–13 to LIB–45
  creating libraries • *Programming Resources*, 1–17
  DCL command LIBRARY • *Librarian*, LIB–11
  DCL qualifiers • *Librarian*, LIB–14 to LIB–45
  default logical names • *Programming Resources*, 1–18
  directing output from • *Librarian*, LIB–12
    See also /LIST qualifier
    See also /OUTPUT qualifier
  exiting • *Librarian*, LIB–12
  format • *Librarian*, LIB–11

Librarian Utility (LIBRARIAN) (Cont.)
  global symbol table (GST) • *Librarian*, LIB–2
  help files • *Librarian*, LIB–4 to LIB–5
  help libraries • *Librarian*, LIB–1, LIB–4 to LIB–5
  HELP LIBRARY command display • *Librarian*, LIB–8 to LIB–10
  help text example • *Librarian*, LIB–6 to LIB–8
  input file specification • *Librarian*, LIB–11
  input_file_spec type • *Librarian*, LIB–12
  invoking • *Librarian*, LIB–12
  key lines in help files • *Librarian*, LIB–5 to LIB–6
  LIBRARIAN routines • *Librarian*, LIB–10
  library
    types of • *Programming Resources*, 1–18
  LIBRARY command • *Programming Resources*, 1–19
  library file specification • *Librarian*, LIB–11
  library-file-spec type • *Librarian*, LIB–11
  library header • *Librarian*, LIB–2
  library index • *Librarian*, LIB–2
  macro libraries • *Librarian*, LIB–1
  module header • *Librarian*, LIB–2
  module name table (MNT) • *Librarian*, LIB–2
  object libraries • *Librarian*, LIB–1
  overview • *Librarian*, LIB–10
  restrictions • *Librarian*, LIB–12
  retrieval of help text • *Librarian*, LIB–8 to LIB–10
  shareable image libraries • *Librarian*, LIB–1, LIB–3
  text libraries • *Librarian*, LIB–1
  types of libraries • *Librarian*, LIB–1
Library • *Message*, MSG–5
  adding module with LBR routine • *Programming Resources*, 8–40
  closing with LBR$ routine • *Programming Resources*, 8–36
  compressing • *Programming Resources*, 8–25
  creating with LBR routine • *Programming Resources*, 8–36
  creation of • *Linker*, 1–5, 2–4
  default object • *Programming Resources*, 5–1
  default user • *Linker*, LINK–21
  deleting module with LBR routine • *Programming Resources*, 8–42
  expanding • *Programming Resources*, 8–25
  identification of • *Linker*, LINK–24, LINK–25
  initializing with LBR routine • *Programming Resources*, 8–36
  input to linker • *Linker*, 1–5, 2–3, 6–3
  inserting module with LBR routine • *Programming Resources*, 8–40
  listing index entries • *Programming Resources*, 8–53

Library (Cont.)
- macro • *Programming Resources,* 5–3, 5–13
- message object module • *Programming Resources,* 9–9
- module header • *Programming Resources,* 8–48
- multiple indexes • *Programming Resources,* 8–45
- multiple keys • *Programming Resources,* 8–45
- object • *Programming Resources,* 5–1, 5–12
  - adding modules • *Programming Resources,* 5–2
  - creating • *Programming Resources,* 5–2
  - deleting a module • *Programming Resources,* 5–2
  - extracting a module • *Programming Resources,* 5–2
  - listing modules • *Programming Resources,* 5–2
  - replacing modules • *Programming Resources,* 5–2
  - system default • *Programming Resources,* 5–2
  - user default • *Programming Resources,* 5–2
- opening with LBR routine • *Programming Resources,* 8–36
- processing index entries • *Programming Resources,* 8–53
- processing index entry with LBR routine • *Programming Resources,* 8–53
- processing of default • *Linker,* 6–14
- reformatting • *Librarian,* LIB–15, LIB–20
- replacing module • *Programming Resources,* 8–40
- shareable image • *Programming Resources,* 5–8
  - adding • *Programming Resources,* 5–8
  - deleting • *Programming Resources,* 5–8
  - listing • *Programming Resources,* 5–8
  - replacing • *Programming Resources,* 5–8
- symbol table • *Linker,* 2–10
- system default • *Programming Resources,* 5–12; *Linker,* 1–5, 2–4, 6–14
- system default object library • *Linker,* LINK–17, LINK–18
- text • *Programming Resources,* 5–3
- type of • *Linker,* 2–3
- types of • *Librarian,* LIB–1
- updating • *Modular Procedures,* 6–5
- user • *Linker,* 2–4
- user default • *Programming Resources,* 5–12
- user-default shareable image • *Linker,* 6–14

LIBRARY command • *Programming Resources,* 1–19; *Librarian,* LIB–11; *Linker,* 2–3
- /CREATE qualifier • *Programming Resources,* 5–2
- creating a new library using /CREATE • *Librarian,* LIB–17

LIBRARY command (Cont.)
- cross-referencing
  - /CROSS_REFERENCE qualifier • *Librarian,* LIB–19
- /DELETE qualifier • *Programming Resources,* 5–2
- directing output • *Librarian,* LIB–12
- exiting • *Librarian,* LIB–12
- /EXTRACT qualifier • *Programming Resources,* 5–2
- format of • *Librarian,* LIB–11
- input file specification • *Librarian,* LIB–11
  - default file type • *Librarian,* LIB–12
- invoking • *Librarian,* LIB–12
- library file specification • *Librarian,* LIB–11
  - default file type • *Librarian,* LIB–11
- /LIST qualifier • *Programming Resources,* 5–2
- qualifiers for • *Librarian,* LIB–13 to LIB–45
- /REPLACE qualifier • *Programming Resources,* 5–2
- restrictions on • *Librarian,* LIB–12
- specifying time in • *Librarian,* LIB–14

.LIBRARY directive • *MACRO,* 6–51

Library facility • *Modular Procedures,* 3–2

Library file
- processing of • *Linker,* 6–9, 6–13
- used as linker input • *Linker,* 1–5

Library file specification • *Librarian,* LIB–11

Library header • *Librarian,* LIB–2

Library index • *Librarian,* LIB–2

Library key • *Librarian,* LIB–2

Library module
- extracting with LBR routine • *Programming Resources,* 8–43

/LIBRARY positional qualifier • *Linker,* LINK–25

Library procedure • *Routines Intro,* 2–4

/LIBRARY qualifier • *Linker,* 2–4; *National Char Set,* NCS–33

Library routine • *Convert,* CONV–1; *File Def Language,* FDL–41, FDL–42

Library size
- See /COMPRESS qualifier
- See /CREATE qualifier

Limit option
- See RAB$V_LIM option

LIM option • *File Def Language,* FDL–11

%LINE • *Debugger,* D–6
- EXAMINE command • *Debugger,* 4–20
- EXAMINE/SOURCE command • *Debugger,* 6–4
- GO command • *Debugger,* CD–105
- SET BREAK command • *Debugger,* 3–11
- SET TRACE command • *Debugger,* 3–11

- %LINE (Cont.)
  - STEP command • *Debugger*, 3–7
- Linear recurrence
  - definition of • *RTL Math*, 2–7
- Line break
  - in data from global selection • *VAXTPU*, 7–300
- LINE command • *VAXTPU*, 4–18
- Line composition • *RTL Screen Management*, 3–2
- Line editing
  - inhibit • *Programming Resources*, 7–42
- Line feed • *File Def Language*, FDL–33
- LINEFEED key command • *Delta/XDelta*, DELTA–22
- LINEFEED key equivalent • *Delta/XDelta*, DELTA–22
- Line mode • *Debugger*, CD–153
- Line-mode editing • *VAXTPU*, C–3
  - example • *VAXTPU*, A–1
- Line number
  - See also %LINE
  - selecting from DECwindows window • *Debugger*, 1–23
  - source display • *Debugger*, 6–1, 6–3, 6–4
    - with DECwindows • *Debugger*, 1–9
  - traceback information • *Debugger*, 2–14, 5–3
  - treated as symbol • *Debugger*, 5–9
- Line-oriented output • *RTL Screen Management*, 2–9
- Line printer
  - carriage control • *I/O User's I*, 5–6, 5–8
  - character case • *I/O User's I*, 5–4
  - character formatting • *I/O User's I*, 5–2
  - device characteristics • *I/O User's I*, 5–3
  - driver • *I/O User's I*, 5–1
  - error recovery • *I/O User's I*, 5–3
  - form feed • *I/O User's I*, 5–4
  - function codes • *I/O User's I*, 5–5, A–5
  - I/O functions
    - IO$_SENSEMODE • *I/O User's I*, 5–9
    - IO$_SETCHAR • *I/O User's I*, 5–9
    - IO$_SETMODE • *I/O User's I*, 5–9
    - IO$_WRITELBLK • *I/O User's I*, 5–5
    - IO$_WRITEPBLK • *I/O User's I*, 5–5
    - IO$_WRITEVBLK • *I/O User's I*, 5–5
  - I/O status block • *I/O User's I*, 5–10
  - printall mode • *I/O User's I*, 5–4
  - programming example • *I/O User's I*, 5–11
  - sense mode function • *I/O User's I*, 5–9
  - set characteristics • *I/O User's I*, 5–9
  - set mode function • *I/O User's I*, 5–9
  - status returns • *I/O User's I*, A–5
  - supported devices • *I/O User's I*, 5–1
  - SYS$GETDVI returns • *I/O User's I*, 5–3
  - write function • *I/O User's I*, 5–5
- Line printer
  - write function (Cont.)
    - carriage control • *I/O User's I*, 5–6
- /LINE qualifier • *Debugger*, 3–14, CD–18, CD–31, CD–87, CD–132, CD–189, CD–266
- "Line" string constant parameter to GET_INFO • *VAXTPU*, 7–172
- Line terminator
  - deleting • *VAXTPU*, 7–28
  - terminal • *I/O User's I*, 8–9
- LINE_BEGIN keyword • *VAXTPU*, 7–69, 7–249 to 7–250, 7–273
  - with POSITION • *VAXTPU*, 7–288
  - with SEARCH • *VAXTPU*, 7–327
  - with SEARCH_QUIETLY • *VAXTPU*, 7–332
- "Line_editing" string constant parameter to GET_ INFO • *VAXTPU*, 7–199
- LINE_END keyword • *VAXTPU*, 7–69, 7–251, 7–273
  - with POSITION • *VAXTPU*, 7–288
  - with SEARCH • *VAXTPU*, 7–327
  - with SEARCH_QUIETLY • *VAXTPU*, 7–332
- LINE_NUMBER keyword • *VAXTPU*, 7–416
- "Line_number" string constant parameter to GET_ INFO • *VAXTPU*, 7–179, 7–206
- Line_Plot graph • *File Applications*, 4–12, A–2
- LINK command • *Debugger*, 5–4, 6–2
  - in command procedure • *Linker*, 3–5
  - invoking linker • *Linker*, 1–2
  - qualifiers • *Linker*, 1–3
    - incompatibility among • *Linker*, LINK–1
  - shareable image • *Debugger*, 5–12
  - with DECwindows • *Debugger*, 1–3
- .LINK directive • *MACRO*, 6–52
  - /INCLUDE qualifier • *MACRO*, 6–52
  - /LIBRARY qualifier • *MACRO*, 6–52
  - /SELECTIVE_SEARCH qualifier • *MACRO*, 6–53
  - /SHAREABLE qualifier • *MACRO*, 6–53
- Linker Utility (LINK) • *Programming Resources*, 1–11 to 1–13; *Librarian*, LIB–1, LIB–3
  - additional controls • *Linker*, 1–12
  - CLUSTER option • *Programming Resources*, 5–6
  - cluster processing order • *Linker*, 1–13
  - command qualifier summary • *Programming Resources*, 1–13
  - conceptual overview • *Linker*, 2–1
  - DCL qualifiers • *Linker*, LINK–1 to LINK–28
  - directing output from • *Linker*, 1–1
  - examples • *Linker*, LINK–31
  - exiting • *Linker*, 1–1
  - GSMATCH option • *Programming Resources*, 5–5, 5–6
  - how to invoke • *Linker*, 1–1

- Linker Utility (LINK) (Cont.)
  - image map • *Programming Resources*, 1–13; *Linker*, 1–12, 5–1
  - input • *Programming Resources*, 1–12
    - file types • *Linker*, 1–4
  - introduction to basic use of • *Linker*, 1–1
  - linker operations • *Linker*, 6–1
  - map
    - use in crash dump analysis • *System Dump Analyzer*, SDA–15
  - object language • *Programming Resources*, 1–13
  - options file • *Programming Resources*, 1–13; *Linker*, 1–6
    - creating • *Modular Procedures*, 5–8
    - descriptions • *Linker*, 1–7 to 1–9
    - how to build • *Linker*, 1–7
    - updating • *Modular Procedures*, 6–6
  - options files • *Linker*, 3–1
  - output • *Programming Resources*, 1–12
    - brief description • *Linker*, 1–5
    - qualifiers used to direct • *Linker*, 1–5
  - parameter
    - for creating executable image • *Linker*, 1–1
  - primary functions • *Linker*, 1–6
  - qualifiers for directing output • *Linker*, 1–2
  - searching object libraries • *Programming Resources*, 5–2
  - shareable image • *Linker*, 1–9
  - shareable images • *Linker*, 4–1
  - UNIVERSAL option • *Programming Resources*, 5–5
  - VAX object language • *Linker*, 7–1
- Link options
  - See Options
- LINK/SHAREABLE command • *Programming Resources*, 5–14
- LINK_CACHE_ENABLE attribute • *File Def Language*, FDL–32
- LINK_TIMEOUT attribute • *File Def Language*, FDL–32
- LIS file • *Delta/XDelta*, DELTA–10, DELTA–11, DELTA–12
- LISP
  - See VAX LISP
- List
  - specifying as a resource value • *VAXTPU*, 4–13
- LIST clause
  - for VALUE clause • *Command Def*, CDU–34
  - with keywords • *Command Def*, CDU–29
  - with parameters • *Command Def*, CDU–24
  - with qualifiers • *Command Def*, CDU–26
- .LIST directive • *MACRO*, 6–55
- .LIST directive (Cont.)
  - See also .SHOW directive
- Listing
  - obtaining
    - See /LIST qualifier
- Listing control directive
  - .IDENT • *MACRO*, 6–39
  - .LIST • *MACRO*, 6–55
  - .NLIST • *MACRO*, 6–65
  - .NOSHOW • *MACRO*, 6–67, 6–89
  - .PAGE • *MACRO*, 6–75
  - .SHOW • *MACRO*, 6–89
- Listing directives • *Message*, MSG–25, MSG–28
- Listing level count • *MACRO*, 6–90
- /LISTING qualifier • *Command Def*, CDU–40; *SUMSLP*, SUM–16
- Listing table of contents • *MACRO*, 6–94
- List Names and Addresses of Loaded Executive Images command • *Delta/XDelta*, DELTA–44
- /LIST qualifier • *Debugger*, 6–1; *Librarian*, LIB–12, LIB–28; *Message*, MSG–11
  - default output destination • *National Char Set*, NCS–34
  - for obtaining listing of NCS library • *National Char Set*, NCS–34
  - information provided by • *National Char Set*, NCS–34
  - LIBRARY command • *Programming Resources*, 5–2
  - specifying output file • *National Char Set*, NCS–34
  - using with /BEFORE • *Librarian*, LIB–14; *National Char Set*, NCS–23
  - using with /FULL • *Librarian*, LIB–23; *National Char Set*, NCS–30
  - using with /HISTORY • *Librarian*, LIB–26; *National Char Set*, NCS–31
  - using with /NAMES • *Librarian*, LIB–33
  - using with /ONLY • *Librarian*, LIB–35; *National Char Set*, NCS–38
  - using with other qualifiers • *National Char Set*, NCS–34
  - using with /SINCE • *Librarian*, LIB–42; *National Char Set*, NCS–41
- Literal directive (.LITERAL)
  - in message source file • *Message*, MSG–21
- Literal mode • *MACRO*, 5–10
  - contrasted with immediate mode • *MACRO*, 5–15
  - operand specifier format • *MACRO*, 8–23
- LKB (lock block) • *System Dump Analyzer*, SDA–108
- LMF$GROUP_TABLE.EXE
  - global symbols • *System Dump Analyzer*, SDA–60

- LNK$LIBRARY • *Programming Resources*, 5–1; *Linker*, LINK–22
  - See also Library
  - See also Linker Utility
- LOADALT macro • *Device Support (A)*, 14–10, 14–22; *Device Support (B)*, 2–44, 3–74
- Load Base Register command • *Delta/XDelta*, DELTA–40
- LOADER$_PTE_NOT_EMPTY status • *Device Support (B)*, 3–108
- LOADMBA macro • *Device Support (A)*, 15–3, 15–13, 15–14 to 15–15; *Device Support (B)*, 2–45, 3–76
- Load option
  - See RAB$V_LOA option
- LOADUBA macro • *Device Support (A)*, 14–10, 14–11, 14–21; *Device Support (B)*, 2–46, 3–77
- LOA option • *File Def Language*, FDL–10, FDL–11
- $LOCAL$INI$ buffer • *VAXTPU*, 4–22
- Local buffer caching
  - with lock management service • *System Services Intro*, 12–14
- LOCAL clause
  - for PLACEMENT clause • *Command Def*, CDU–25, CDU–34
- LOCAL declaration • *VAXTPU*, 3–34 to 3–35
- Local disk UCB extension • *Device Support (B)*, 1–69, 1–82 to 1–84
  - required for error logging • *Device Support (A)*, 11–9; *Device Support (B)*, 3–9
  - required for IOC$APPLYECC routine • *Device Support (B)*, 3–67
- Local label
  - saving • *MACRO*, 6–87
  - user-defined • *MACRO*, 3–7
- Local label block
  - ending • *MACRO*, 6–22
  - starting • *MACRO*, 6–22
- Local processor • *Device Support (A)*, 1–7
- /LOCAL qualifier • *Debugger*, 8–6, CD–47, CD–57, CD–250
- "Local" string constant parameter to GET_INFO • *VAXTPU*, 7–179
- Local symbol • *Programming Resources*, 5–11; *Linker*, 2–8; *Patch*, PAT–8; *MACRO*, 3–6
  - See Symbol
  - signaling with • *Programming Resources*, 9–11
- Local tape UCB extension • *Device Support (B)*, 1–69, 1–81 to 1–82
  - required for error logging • *Device Support (A)*, 11–9; *Device Support (B)*, 3–9
- Local variable • *VAXTPU*, 3–4, 3–20, 3–34
- Locate mode
  - and record retrieval • *File Applications*, 8–2
  - comparing with move mode for buffer handling • *RMS*, 7–15
- Locate mode option
  - See RAB$V_LOC option
- %LOCATE operator • *MACRO*, 4–9
- LOCATE_MODE attribute • *File Def Language*, FDL–11
- LOCATE_MOUSE built-in procedure • *VAXTPU*, 7–252 to 7–253
- Location
  - examining • *System Dump Analyzer*, SDA–51
  - SDA default • *System Dump Analyzer*, SDA–51
  - translating to VAX MACRO instruction • *System Dump Analyzer*, SDA–51
- Location control directive
  - .ALIGN • *MACRO*, 6–5
  - .BLKx • *MACRO*, 6–12
- Location counter alignment directive (.ODD) • *MACRO*, 6–71
- Location counter control directive (.EVEN) • *MACRO*, 6–33
- Location field in XABALL
  - See XAB$L_LOC field
- LOCC (Locate Character) instruction • *MACRO*, 9–130
- Lock
  - See Spin lock
  - choice of mode • *System Services Intro*, 12–3
  - concept of • *System Services Intro*, 12–1
  - conversion • *System Services Intro*, 12–6, 12–10
  - deadlock detection • *System Services Intro*, 12–6
  - dequeuing • *System Services Intro*, 12–13
  - displaying SDA information • *System Dump Analyzer*, SDA–143
  - getting information about
    - asynchronously • *System Services*, SYS–239
    - synchronously • *System Services*, SYS–252
  - level • *System Services Intro*, 12–4
  - mode • *System Services Intro*, 12–3
  - root • *File Applications*, 3–29
- Lock block
  - See LKB
- Lock database
  - in a VAXcluster • *System Services*, SYS–249
- Lockdown (poor man's) • *Device Support (A)*, E–16 to E–17; *Device Support (B)*, 2–49 to 2–50, 2–97
- Lock ID • *Device Support (B)*, 1–73
- /LOCKID qualifier • *System Dump Analyzer*, SDA–143
- LOCKING.EXE • *System Dump Analyzer*, SDA–60

LOCK macro • *Device Support (A)*, 3–9, 3–10, E–4; *Device Support (B)*, 2–47 to 2–48, 3–111
Lock management routines
  global symbols • *System Dump Analyzer*, SDA–60
Lock management service • *System Services Intro*, 1–3
  for interprocess communication • *System Services Intro*, 8–11
Lock manager • *Programming Resources*, 4–13; *Modular Procedures*, 3–21; *Routines Intro*, A–9t; *Device Support (B)*, 1–73
  See also Synchronization
  displaying SDA information • *System Dump Analyzer*, SDA–108
  queueing a lock request • *Programming Resources*, 4–14
Lock mode • *System Dump Analyzer*, SDA–144
Lock record for read option
  See RAB$V_REA option
Lock record for write option
  See RAB$V_RLK option
Lock request
  dequeuing • *System Services*, SYS–136
  queuing • *System Services Intro*, 12–4
    asynchronously • *System Services*, SYS–148
    synchronously • *System Services*, SYS–158
  synchronizing • *System Services Intro*, 12–8
/LOCKS qualifier • *System Dump Analyzer*, SDA–127
Lock status block • *System Services Intro*, 12–9; *System Services*, SYS–150
Lock value block • *System Services*, SYS–150
  description • *System Services Intro*, 12–12
  using • *System Services Intro*, 12–15
Lock values • *Routines Intro*, A–9t
lock_id data type • *Routines Intro*, A–9t
LOCK_ON_READ attribute • *File Def Language*, FDL–11
LOCK_ON_READ secondary attribute • *File Applications*, 7–11
LOCK_ON_WRITE attribute • *File Def Language*, FDL–11
LOCK_ON_WRITE secondary attribute • *File Applications*, 7–11
/LOCK_STATE qualifier • *Debugger*, CD–51
lock_status_block data type • *Routines Intro*, A–9t
LOCK_SYSTEM_PAGES macro • *Device Support (B)*, 2–49
lock_value_block data type • *Routines Intro*, A–10t
Logarithm
  base 2 • *RTL Math*, MTH–94, MTH–114
  common • *RTL Math*, MTH–96, MTH–116
  natural • *RTL Math*, MTH–92, MTH–112
Logarithm (Cont.)
  natural complex • *RTL Math*, MTH–35, MTH–37
Log file
  as command procedure • *Debugger*, 8–5
  debugger • *Debugger*, 8–5, CD–159
    with DECwindows • *Debugger*, 1–29
  name of • *Debugger*, 8–5, CD–147, CD–227
Logical AND operator
  See AND operator
Logical-block-position option • *File Applications*, 4–31
Logical exclusive OR operator
  See Exclusive OR operator
Logical functions, vector • *MACRO*, 10–64
Logical I/O
  access checks • *System Services Intro*, 7–10
  operations • *System Services Intro*, 7–7
  privilege • *System Services Intro*, 7–4, 7–7
Logical I/O function
  translation from virtual function to • *Device Support (A)*, 2–3
  translation to physical function • *Device Support (B)*, 3–31, 3–40, 3–54
Logical inclusive OR operator
  See Inclusive OR operator
Logical instruction • *MACRO*, 9–5
Logical name • *System Services Intro*, 7–26; *RTL Library*, LIB–340
  advantages • *File Applications*, 5–4
  attributes • *System Services Intro*, 6–8
  concealed attribute • *File Applications*, 5–7
  concealed-device • *File Applications*, 6–15
  creating • *System Services Intro*, 6–12; *System Services*, SYS–68
  debugger • *Debugger*, D–1
  defining • *System Services Intro*, 6–1
  deleting • *System Services Intro*, 6–17; *System Services*, SYS–127
  duplicating • *System Services Intro*, 6–13
  example program • *File Applications*, 5–5 to 5–6
  for interprocess communication • *System Services Intro*, 8–10
  format convention • *System Services Intro*, 6–10
  getting information about • *System Services*, SYS–520
  image rundown • *System Services Intro*, 6–5
  multivalued • *System Services Intro*, 6–2
  parsing • *File Applications*, 5–7
  rooted-device • *File Applications*, 6–15
  RTL routines • *RTL Library*, LIB–114
  search list • *File Applications*, 5–7, 6–7 to 6–8
  supersession • *System Services Intro*, 6–15

Logical name (Cont.)
  system services • *System Services Intro,* 6–1
  translation of • *System Services Intro,* 6–17; *System Services,* SYS–520; *File Applications,* 5–7, 6–5 to 6–7
  types of • *File Applications,* 5–6 to 5–7
Logical names
  EVE$INIT • *VAXTPU,* 4–31
  TPU$COMMAND • *VAXTPU,* 5–6
  TPU$DEBUG • *VAXTPU,* 5–8
  TPU$SECTION • *VAXTPU,* 5–16
Logical name system service call
  example of
    SYS$CRELNM • *System Services Intro,* 6–12
    SYS$CRELNT • *System Services Intro,* 6–16
    SYS$DELLNM • *System Services Intro,* 6–17
    SYS$TRNLNM • *System Services Intro,* 6–18
Logical name table
  controlling access through access control lists • *Utility Routines,* ACL–1
  creating • *System Services Intro,* 6–16; *System Services,* SYS–74
  default • *System Services Intro,* 6–3
  deleting • *System Services,* SYS–127
  directory • *System Services Intro,* 6–3
  group • *System Services Intro,* 6–6
  job • *System Services Intro,* 6–5
  predefined logical names • *System Services Intro,* 6–2
  process • *System Services Intro,* 6–4
  process-private • *System Services Intro,* 6–6
  quotas • *System Services Intro,* 6–9
  search list • *System Services Intro,* 6–11
    modifying • *System Services Intro,* 6–12
  shareable • *System Services Intro,* 6–6, 6–16
  system • *System Services Intro,* 6–6
  types of • *System Services Intro,* 6–2
  user-defined • *System Services Intro,* 6–6
Logical name translation
  requirements for parsing • *RMS,* 4–9
Logical name translation access mode subfield
  See FAB$V_LNM_MODE subfield
Logical NOT operator (#) • *System Dump Analyzer,* SDA–12
Logical operator • *System Dump Analyzer,* SDA–12
Logical operators
  AND operator • *VAXTPU,* 3–7
  NOT operator • *VAXTPU,* 3–7
  OR operator • *VAXTPU,* 3–7
  XOR operator • *VAXTPU,* 3–7
LOGICAL option • *File Applications,* 4–31
Logical OR operator (|) • *System Dump Analyzer,* SDA–12
Logical predecessor • *Debugger,* 4–8, 4–20, D–5
  with DECwindows • *Debugger,* 1–8
Logical successor • *Debugger,* 4–8, 4–20, D–5
  with DECwindows • *Debugger,* 1–8
Logical unit number
  allocating • *RTL Library,* 2–17
  RTL routine to free • *RTL Library,* LIB–175
Logical unit numbers • *Modular Procedures,* 2–16
Logical value • *File Def Language,* FDL–2
Logical XOR operator (\) • *System Dump Analyzer,* SDA–13
logical_name data type • *Routines Intro,* A–10t
LOGICAL_NAMES.EXE
  global symbols • *System Dump Analyzer,* SDA–60
/LOG qualifier • *Debugger,* CD–51, CD–59; *Librarian,* LIB–30
  See also /DELETE qualifier
  See also /REPLACE qualifier
  CREATE/FDL • *File Def Language,* FDL–45
  for verifying NCS library operations • *National Char Set,* NCS–35
.LONG directive • *MACRO,* 6–56
Longest record length field
  See XAB$W_LRL field
LONG mode • *Patch,* PAT–16
/LONG qualifier
  with ALIGN command • *Patch,* PAT–38
  with DELETE command • *Patch,* PAT–52
  with DEPOSIT command • *Patch,* PAT–55
  with EVALUATE command • *Patch,* PAT–59
  with EXAMINE command • *Patch,* PAT–62
  with REPLACE command • *Patch,* PAT–71
  with SET MODE command • *Patch,* PAT–76
  with VERIFY command • *Patch,* PAT–90
Longword
  to convert with FAO • *VAXTPU,* 7–138
  to convert with MESSAGE • *VAXTPU,* 7–268
  to convert with MESSAGE_TEXT • *VAXTPU,* 7–271
Longword access enable bit
  See VEC$V_LWAE
Longword-aligned random-access mode • *Device Support (A),* 14–3, 14–11, 14–14 to 14–15; *Device Support (B),* 1–26
Longword condition value • *System Services Intro,* 1–5
Longword data type • *MACRO,* 8–2
/LONGWORD qualifier • *Debugger,* CD–63, CD–87
Longwords • *System Services Intro,* 2–4

Longword storage directive (.LONG) • *MACRO*, 6–56
longword_signed data type • *Routines Intro*, A–10t
longword_unsigned data type • *Routines Intro*, A–10t
Lookaside list
  See Nonpaged pool
  displaying contents • *System Dump Analyzer*, SDA–118
LOOKUP_KEY built-in procedure • *VAXTPU*, 7–254 to 7–257
Loopback mode • *Device Support (B)*, 1–91
LOOP statement • *VAXTPU*, 3–21 to 3–22
Lowest level of index area number field
  See XAB$B_LAN field
"Low_index" string constant parameter to GET_INFO • *VAXTPU*, 7–167
LPA11-K device
  AST
    address • *I/O User's I*, 4–12, 4–14
    quota • *I/O User's I*, 4–14
    synchronization • *I/O User's I*, 4–14
  buffer management • *I/O User's I*, 4–16
  buffer overrun • *I/O User's I*, 4–12, 4–14, 4–31
  buffer queue control • *I/O User's I*, 4–16
  clock rate • *I/O User's I*, 4–10
  data buffer • *I/O User's I*, 4–14
  data sampling • *I/O User's I*, 4–1
  data transfer command table • *I/O User's I*, 4–11
  data transfer start command • *I/O User's I*, 4–12
  data transfer stop command • *I/O User's I*, 4–14
  data underrun/overrun • *I/O User's I*, 4–12
  device characteristics • *I/O User's I*, 4–5 to 4–8
  device configuration • *I/O User's I*, 4–2, 4–10, 4–34
  device initialization • *I/O User's I*, 4–4, 4–8 to 4–9, 4–32, 4–34
  driver • *I/O User's I*, 4–1
  errors • *I/O User's I*, 4–2
  features • *I/O User's I*, 4–3
  function codes • *I/O User's I*, 4–8, A–4
  function modifier
    IO$M_SETEVF • *I/O User's I*, 4–11, 4–14
  high-level language support routines • *I/O User's I*, 4–15
  I/O functions
    IO$_INITIALIZE • *I/O User's I*, 4–9
    IO$_LOADMCODE • *I/O User's I*, 4–8
    IO$_SETCLOCK • *I/O User's I*, 4–10
    IO$_STARTDATA • *I/O User's I*, 4–11
    IO$_STARTMPROC • *I/O User's I*, 4–9
  I/O status block • *I/O User's I*, 4–33
  initialize command table • *I/O User's I*, 4–9
  initialize function • *I/O User's I*, 4–9

LPA11-K device (Cont.)
  load microcode function • *I/O User's I*, 4–8
  maintenance status register • *I/O User's I*, 4–10, 4–33
  microcode loading • *I/O User's I*, 4–4, 4–8, 4–32, 4–34
  modes of operation • *I/O User's I*, 4–1
  operator process • *I/O User's I*, 4–35
  programming examples • *I/O User's I*, 4–37, 4–39, 4–44
  RSX-11M/M-PLUS and VMS differences • *I/O User's I*, 4–35
  set clock function • *I/O User's I*, 4–10
  start data transfer request function • *I/O User's I*, 4–11
  start microprocessor function • *I/O User's I*, 4–9
  status returns • *I/O User's I*, 4–9, 4–10, 4–11, 4–14, 4–33, A–5
  stop command • *I/O User's I*, 4–14
  subroutines
    argument usage • *I/O User's I*, 4–16 to 4–19
    list • *I/O User's I*, 4–15
  supported device • *I/O User's I*, 4–1
  supporting software • *I/O User's I*, 4–3
  SYS$CANCEL routine • *I/O User's I*, 4–14
  SYS$GETDVI returns • *I/O User's I*, 4–5
  timeout error • *I/O User's I*, 4–2
LRP (large request packet) • *System Dump Analyzer*, SDA–118
LRP lookaside list
  displaying • *System Dump Analyzer*, SDA–118
/LRP qualifier • *System Dump Analyzer*, SDA–118
LUN (logical unit number) • *Device Support (A)*, 17–2
LWAE (longword access enable) bit
  See VEC$V_LWAE

M

MA780 multiport memory
  configuring a dump file for • *System Dump Analyzer*, SDA–3
Machine check • *Device Support (A)*, 3–14, 13–22, 19–7
  condition handler • *Device Support (A)*, 19–7
Machine check code
  base address • *System Dump Analyzer*, SDA–14
Machine check protection block • *Device Support (A)*, 16–13, 16–14
Machine checks • *MACRO*, 10–43, 10–47
Macro • *File Def Language*, FDL–41; *MACRO*, 4–1

Macro (Cont.)
  applicable VAX MACRO syntax rules • *RMS*, 3–5
  arguments for service completion routines • *RMS*, 3–11
  capabilities listed • *RMS*, 4–1
  control block initialization • *RMS*, 3–1
  for defining VMS RMS symbol • *RMS*, 3–1
  for initializing VMS RMS control blocks • *RMS*, 3–1
  for invoking VMS RMS at run time • *RMS*, 3–1
  format • *Device Support (B)*, 2–1
  for VMS RMS control block store • *RMS*, 3–1
  library location • *RMS*, 3–2
  names and control blocks • *RMS*, 3–2
  naming conventions • *RMS*, 3–2
  nested • *MACRO*, 4–4
  passing numeric value to • *MACRO*, 4–6
  rules applicable to programming • *RMS*, 3–6
  service • *RMS*, 3–1
  syntax applicable to VMS RMS • *RMS*, 3–1
  using • *RMS*, 3–6
  VMS RMS types • *RMS*, 3–1
  with the same name as an opcode • *MACRO*, 6–58
MACRO
  See also VAX MACRO
  See also VAX MACRO instruction
  See Instructions
  CALLG (Call Procedure with General Argument List) instruction • *System Services Intro*, 2–10
  calling system service using • *System Services Intro*, 2–9
  CALLS (Call Procedure with Stack Argument List) instruction • *System Services Intro*, 2–10
  expansion • *System Services Intro*, 2–8
  system service • *System Services Intro*, 2–1, 2–5
MACRO-32 file format, from NCS library
  See /FORMAT qualifier
MACRO-32 output, from NCS library
  See /MACRO qualifier
Macro argument • *MACRO*, 4–1
  actual • *MACRO*, 4–1
  concatenated • *MACRO*, 4–5
  delimited • *MACRO*, 4–3, 4–5
  formal • *MACRO*, 4–1
  keyword • *MACRO*, 4–3
  positional • *MACRO*, 4–3
  string • *MACRO*, 4–3
Macro call • *MACRO*, 4–1
  as operator • *MACRO*, 2–3
  listing • *MACRO*, 6–89
  number of arguments • *MACRO*, 6–63
Macro call directive (.MCALL) • *MACRO*, 6–60
MACRO data type declaration • *Routines Intro*, A–36
Macro definition • *MACRO*, 4–1
  default value • *MACRO*, 4–2
  end • *MACRO*, 6–27
  labeling in • *MACRO*, 4–7
  listing • *MACRO*, 6–89
Macro definition directive (.MACRO) • *MACRO*, 6–57
Macro deletion directive (.MDELETE) • *MACRO*, 6–61
.MACRO directive • *MACRO*, 6–57
Macro exit directive (.MEXIT) • *MACRO*, 6–62
Macro expansion
  listing • *MACRO*, 6–89
  printing • *MACRO*, 4–1
  terminating • *MACRO*, 6–62
Macro field
  example of initializing • *RMS*, 3–5
  setting at run time • *RMS*, 3–5
MACRO implementation table • *Routines Intro*, A–36
Macroinstruction
  See Macro
Macro library • *Programming Resources*, 1–18, 5–13; *Librarian*, LIB–1
  adding a name to • *MACRO*, 6–51
  character case in • *Librarian*, LIB–2
Macro library directive (.LIBRARY) • *MACRO*, 6–51
Macro link directive (.LINK) • *MACRO*, 6–52
Macro name • *MACRO*, 3–6
Macro operator
  %EXTRACT • *MACRO*, 4–10
  %LENGTH • *MACRO*, 4–8
  %LOCATE • *MACRO*, 4–9
  string • *MACRO*, 4–8
/MACRO qualifier • *Librarian*, LIB–31; *National Char Set*, NCS–36
Macro string operator
  summary • *MACRO*, C–8
Magnetic tape
  ACP control function • *I/O User's I*, 1–30, 6–15
  ACP create file operation • *I/O User's I*, 1–26
  available function • *I/O User's I*, 6–27
  BOT marker • *I/O User's I*, 6–19, 6–20
  byte count
   read • *I/O User's I*, 6–17
   write • *I/O User's I*, 6–19
  data check • *I/O User's I*, 6–8, 6–17, 6–18
  data security erase function • *I/O User's I*, 6–27
  density • *I/O User's I*, 6–26
  device characteristics • *I/O User's I*, 6–11 to 6–12
  driver • *I/O User's I*, 6–1

Magnetic tape (Cont.)
  end-of-volume detection • *I/O User's I*, 6–20
  EOF status • *I/O User's I*, 6–17
  EOT
    marker • *I/O User's I*, 6–20 to 6–21
    status • *I/O User's I*, 6–17, 6–19, 6–21
  error recovery • *I/O User's I*, 6–9
  extended characteristics • *I/O User's I*, 6–12
  features • *I/O User's I*, 6–6
  file • *File Def Language*, FDL–21
  file attributes • *I/O User's I*, 6–9
  file expiration • *File Def Language*, FDL–16
  file protection • *File Def Language*, FDL–22
  function codes • *I/O User's I*, 6–13, A–6
  function modifiers
    IO$M_DATACHECK • *I/O User's I*, 6–8, 6–17, 6–18
    IO$M_ERASE • *I/O User's I*, 6–18
    IO$M_INHEXTGAP • *I/O User's I*, 6–10
    IO$M_INHRETRY • *I/O User's I*, 6–9
    IO$M_NOWAIT • *I/O User's I*, 6–19, 6–21, 6–22
    IO$M_REVERSE • *I/O User's I*, 6–17
  I/O functions • *I/O User's I*, 6–13
    See also ACP-QIO interface
    arguments • *I/O User's I*, 6–15
    IO$_ACCESS • *I/O User's I*, 6–13
    IO$_ACPCONTROL • *I/O User's I*, 1–31, 6–15
    IO$_AVAILABLE • *I/O User's I*, 6–27
    IO$_CREATE • *I/O User's I*, 6–13
    IO$_DEACCESS • *I/O User's I*, 6–13
    IO$_DSE • *I/O User's I*, 6–13, 6–27
    IO$_FLUSH • *I/O User's I*, 6–13
    IO$_MODIFY • *I/O User's I*, 6–13
    IO$_PACKACK • *I/O User's I*, 6–27
    IO$_READLBLK • *I/O User's I*, 6–17
    IO$_READPBLK • *I/O User's I*, 6–17
    IO$_READVBLK • *I/O User's I*, 6–17
    IO$_REWIND • *I/O User's I*, 6–19
    IO$_REWINDOFF • *I/O User's I*, 6–21
    IO$_SENSEMODE • *I/O User's I*, 6–22
    IO$_SETCHAR • *I/O User's I*, 6–23
    IO$_SETMODE • *I/O User's I*, 6–23
    IO$_SKIPFILE • *I/O User's I*, 6–19
    IO$_SKIPRECORD • *I/O User's I*, 6–20
    IO$_UNLOAD • *I/O User's I*, 6–22
    IO$_WRITELBLK • *I/O User's I*, 6–18
    IO$_WRITEOF • *I/O User's I*, 6–21
    IO$_WRITEPBLK • *I/O User's I*, 6–18
    IO$_WRITEVBLK • *I/O User's I*, 6–18
  I/O status block • *I/O User's I*, 6–28

Magnetic tape (Cont.)
  master adapters • *I/O User's I*, 6–8
  pack acknowledge function • *I/O User's I*, 6–27
  parity • *I/O User's I*, 6–26
  positioning • *I/O User's I*, 1–31
  programming example • *I/O User's I*, 6–28
  quotas • *I/O User's I*, 6–13
  read function • *I/O User's I*, 6–17
  read reverse function • *I/O User's I*, 6–17, 6–18
  rewind function • *I/O User's I*, 6–19
  rewind offline function • *I/O User's I*, 6–21
  sense mode function • *I/O User's I*, 6–22
  set characteristics function • *I/O User's I*, 6–23
  set mode function • *I/O User's I*, 6–23
    characteristics • *I/O User's I*, 6–25
  skip file function • *I/O User's I*, 6–19
  skip record function • *I/O User's I*, 6–20
  slave formatter • *I/O User's I*, 6–8
  starting position • *File Def Language*, FDL–21
  status returns • *I/O User's I*, A–7
  streaming tape systems • *I/O User's I*, 6–10
  supported devices • *I/O User's I*, 6–1
  SYS$GETDVI returns • *I/O User's I*, 6–11
  tape controllers • *I/O User's I*, 6–3
  tape mark • *I/O User's I*, 6–17, 6–20
  thrashing • *I/O User's I*, 6–10
  TMSCP magnetic tapes • *I/O User's I*, 6–1
  TU58 magnetic tape
    See Disk, TU58
  unload function • *I/O User's I*, 6–22
  write end-of-file function • *I/O User's I*, 6–21
  write function • *I/O User's I*, 6–18
Magnetic tape accessibility field
  See XAB$B_MTACC field
Magnetic tape processing
  run-time options • *File Applications*, 9–13 to 9–14
MAIL • *Utility Routines*, MAIL–1
  action routine • *Utility Routines*, MAIL–8
    calling sequence • *Utility Routines*, MAIL–9
    folder • *Utility Routines*, MAIL–11, MAIL–15
    mail file • *Utility Routines*, MAIL–15
    send • *Utility Routines*, MAIL–18
  address list • *Utility Routines*, MAIL–17
    creating • *Utility Routines*, MAIL–17
    username type • *Utility Routines*, MAIL–17
  bodypart
    creating • *Utility Routines*, MAIL–17
  condition handling • *Utility Routines*, MAIL–6
  context • *Utility Routines*, MAIL–4
    initiating • *Utility Routines*, MAIL–4
    mail file • *Utility Routines*, MAIL–9

- MAIL
  - context (Cont.)
    - message • *Utility Routines*, MAIL–12
    - send • *Utility Routines*, MAIL–16
    - terminating • *Utility Routines*, MAIL–4
    - user profile • *Utility Routines*, MAIL–19
  - deleted bytes threshold • *Utility Routines*, MAIL–12
  - disk space
    - reclaim • *Utility Routines*, MAIL–12
  - folder • *Utility Routines*, MAIL–2
    - creating • *Utility Routines*, MAIL–15
    - deleting • *Utility Routines*, MAIL–15
  - folder names
    - displaying • *Utility Routines*, MAIL–11
  - introduction • *Utility Routines*, MAIL–1
  - item code • *Utility Routines*, MAIL–8
    - Boolean • *Utility Routines*, MAIL–8
    - input • *Utility Routines*, MAIL–8, MAIL–21
    - output • *Utility Routines*, MAIL–8, MAIL–23
  - item descriptor
    - declaring • *Utility Routines*, MAIL–8
    - null • *Utility Routines*, MAIL–8
  - item list • *Utility Routines*, MAIL–6
    - declaring • *Utility Routines*, MAIL–8
    - terminating • *Utility Routines*, MAIL–8
  - mail file • *Utility Routines*, MAIL–3
    - alternate • *Utility Routines*, MAIL–10
    - closing • *Utility Routines*, MAIL–10
    - compressing • *Utility Routines*, MAIL–12
    - creating • *Utility Routines*, MAIL–15
    - default • *Utility Routines*, MAIL–10
    - opening • *Utility Routines*, MAIL–10
    - purging • *Utility Routines*, MAIL–12
    - specifying • *Utility Routines*, MAIL–10 to MAIL–11
    - wastebasket • *Utility Routines*, MAIL–12
  - mail file context
    - initiating • *Utility Routines*, MAIL–9
    - terminating • *Utility Routines*, MAIL–9
  - message • *Utility Routines*, MAIL–1
    - attribute • *Utility Routines*, MAIL–17
    - copying • *Utility Routines*, MAIL–15
    - creating • *Utility Routines*, MAIL–17
    - deleting • *Utility Routines*, MAIL–16
    - displaying • *Utility Routines*, MAIL–14
    - marking • *Utility Routines*, MAIL–14
    - modifying • *Utility Routines*, MAIL–14
    - moving • *Utility Routines*, MAIL–15
    - printing • *Utility Routines*, MAIL–14
    - reading • *Utility Routines*, MAIL–14
    - selecting • *Utility Routines*, MAIL–13
    - sending • *Utility Routines*, MAIL–17, MAIL–18
- MAIL (Cont.)
  - message attribute
    - creating • *Utility Routines*, MAIL–17
  - message context
    - initiating • *Utility Routines*, MAIL–13
    - terminating • *Utility Routines*, MAIL–13
  - message format
    - standard • *Utility Routines*, MAIL–1
  - message header
    - creating • *Utility Routines*, MAIL–17
  - message ID
    - external • *Utility Routines*, MAIL–2
  - null item list • *Utility Routines*, MAIL–8
  - programming examples • *Utility Routines*, MAIL–25
  - send context
    - initiating • *Utility Routines*, MAIL–16
    - terminating • *Utility Routines*, MAIL–16
  - signaling error • *Utility Routines*, MAIL–6
    - disabling • *Utility Routines*, MAIL–6
  - thread • *Utility Routines*, MAIL–5 to MAIL–6
  - user common database • *Utility Routines*, MAIL–3, MAIL–19
  - user context
    - initiating • *Utility Routines*, MAIL–19
    - terminating • *Utility Routines*, MAIL–19
  - user profile
    - flags • *Utility Routines*, MAIL–20
    - form • *Utility Routines*, MAIL–20
    - forward addressing • *Utility Routines*, MAIL–20
    - personal name • *Utility Routines*, MAIL–20
    - queue name • *Utility Routines*, MAIL–20
  - user profile entry • *Utility Routines*, MAIL–3, MAIL–19
    - adding • *Utility Routines*, MAIL–20
    - deleting • *Utility Routines*, MAIL–20
    - modifying • *Utility Routines*, MAIL–20
- MAIL$MAILFILE_BEGIN • *Utility Routines*, MAIL–34
- MAIL$MAILFILE_CLOSE • *Utility Routines*, MAIL–38
- MAIL$MAILFILE_COMPRESS • *Utility Routines*, MAIL–41
- MAIL$MAILFILE_END • *Utility Routines*, MAIL–44
- MAIL$MAILFILE_INFO_FILE • *Utility Routines*, MAIL–46
- MAIL$MAILFILE_MODIFY • *Utility Routines*, MAIL–49
- MAIL$MAILFILE_OPEN • *Utility Routines*, MAIL–52
- MAIL$MAILFILE_PURGE_WASTE • *Utility Routines*, MAIL–55
- MAIL$MESSAGE_BEGIN • *Utility Routines*, MAIL–58
- MAIL$MESSAGE_COPY • *Utility Routines*, MAIL–62
- MAIL$MESSAGE_DELETE • *Utility Routines*, MAIL–67
- MAIL$MESSAGE_END • *Utility Routines*, MAIL–69

MAIL$MESSAGE_GET • *Utility Routines*, MAIL–71
MAIL$MESSAGE_INFO • *Utility Routines*, MAIL–76
MAIL$MESSAGE_MODIFY • *Utility Routines*, MAIL–80
MAIL$MESSAGE_SELECT • *Utility Routines*, MAIL–83
MAIL$SEND_ABORT • *Utility Routines*, MAIL–87
MAIL$SEND_ADD_ADDRESS • *Utility Routines*, MAIL–89
MAIL$SEND_ADD_ATTRIBUTE • *Utility Routines*, MAIL–91
MAIL$SEND_ADD_BODYPART • *Utility Routines*, MAIL–98
MAIL$SEND_BEGIN • *Utility Routines*, MAIL–94
MAIL$SEND_END • *Utility Routines*, MAIL–101
MAIL$SEND_MESSAGE • *Utility Routines*, MAIL–103
MAIL$USER_BEGIN • *Utility Routines*, MAIL–105
MAIL$USER_DELETE_INFO • *Utility Routines*, MAIL–110
MAIL$USER_END • *Utility Routines*, MAIL–112
MAIL$USER_GET_INFO • *Utility Routines*, MAIL–114
MAIL$USER_SET_INFO • *Utility Routines*, MAIL–118
Mailbox • *Programming Resources*, 3–7; *System Services Intro*, 2–1, 7–30; *RTL Library*, 2–23, LIB–12; *Device Support (B)*, 1–75, 1–76, 1–77
  See also Terminal
  assigning channel to • *System Services*, SYS–82
  associated with device • *Device Support (B)*, 1–77
  buffered I/O quota for • *Device Support (B)*, 1–73
  controlling access through access control lists • *Utility Routines*, ACL–1
  creating • *Programming Resources*, 3–8; *System Services*, SYS–82; *I/O User's I*, 7–1
  deleting • *I/O User's I*, 7–2
    permanent • *System Services*, SYS–85, SYS–130
    temporary • *System Services*, SYS–85
  device characteristics • *I/O User's I*, 7–4
  disable terminal • *I/O User's I*, 8–21
  driver • *I/O User's I*, 7–1
  for interprocess communication • *System Services Intro*, 8–11
  function codes • *I/O User's I*, 7–5, A–7
  function modifiers
    IO$M_NORSWAIT • *I/O User's I*, 7–7
    IO$M_NOW • *I/O User's I*, 7–2, 7–6, 7–7, 7–9, 7–10
    IO$M_READATTN • *I/O User's I*, 7–9
    IO$M_SETPROT • *I/O User's I*, 7–11
  I/O function • *Device Support (B)*, 1–40
  I/O functions
    IO$_READLBLK • *I/O User's I*, 7–5

Mailbox
  I/O functions (Cont.)
    IO$_READPBLK • *I/O User's I*, 7–5
    IO$_READVBLK • *I/O User's I*, 7–5
    IO$_WRITELBLK • *I/O User's I*, 7–6
    IO$_WRITEOF • *I/O User's I*, 7–9
    IO$_WRITEPBLK • *I/O User's I*, 7–6
    IO$_WRITEVBLK • *I/O User's I*, 7–6
  I/O status block • *I/O User's I*, 7–12
  input/output
    asynchronous • *Programming Resources*, 3–9
    immediate • *Programming Resources*, 3–9
    synchronous • *Programming Resources*, 3–9
    using SYS$QIO • *Programming Resources*, 3–9
    using SYS$QIOW • *Programming Resources*, 3–9
  in shared memory • *Device Support (B)*, 1–78
  list of operations • *I/O User's I*, 7–1
  marked for deletion • *Device Support (B)*, 1–78
  message format • *I/O User's I*, 7–3; *I/O User's II*, 1–3
    terminal • *I/O User's I*, 8–18
  message size • *I/O User's I*, 7–2
  multiport memory • *I/O User's I*, 7–1
  name • *System Services Intro*, 7–32
  of job controller • *Device Support (A)*, 9–7, E–7
  of OPCOM process • *Device Support (A)*, 10–7, E–7
  permanent • *Programming Resources*, 3–8; *I/O User's I*, 7–2, 7–3, 7–4; *Device Support (B)*, 1–78
  programming example • *I/O User's I*, 7–14
  protection • *System Services Intro*, 7–4, 7–5; *I/O User's I*, 7–2, 7–4, 7–11
  read attention AST function • *I/O User's I*, 7–9
  read function • *I/O User's I*, 7–5
  reading data from • *Programming Resources*, 3–9
  sending a message to • *Device Support (B)*, 3–52 to 3–53, 3–61
  set attention AST function • *I/O User's I*, 7–9
  set protection function • *I/O User's I*, 7–11
  status returns • *I/O User's I*, A–7
  synchronizing access to • *Device Support (A)*, 3–8, 3–14
  SYS$GETDVI returns • *I/O User's I*, 7–4
  system • *System Services Intro*, 7–33
    messages • *System Services Intro*, 7–33
  temporary • *Programming Resources*, 3–8; *I/O User's I*, 7–2, 7–4
  terminal/mailbox interaction • *I/O User's I*, 8–17
  termination • *System Services Intro*, 8–18

Mailbox (Cont.)
- volume protection • *I/O User's I*, 7–11
- write attention AST function • *I/O User's I*, 7–9
- write end-of-file message function • *I/O User's I*, 7–9
- write function • *I/O User's I*, 7–6
- writing data to • *Programming Resources*, 3–9

Mailbox driver • *Device Support (A)*, 12–5
MAILBOX spin lock • *Device Support (A)*, 3–14; *Device Support (B)*, 3–52, 3–61
Main headings • *Routines Intro*, 1–1
Maintenance function • *Device Support (A)*, 18–15
Main window widget • *VAXTPU*, 4–16
Major ID • *Linker*, 3–7
- of shareable image in map • *Linker*, 5–6

MANAGE CHILDREN routine
- See MANAGE_WIDGET built-in procedure

MANAGE CHILD routine
- See MANAGE_WIDGET built-in procedure

MANAGE_WIDGET built-in procedure • *VAXTPU*, 7–258
- example of use • *VAXTPU*, B–4 to B–11

Managing widget
- controlling mapping • *VAXTPU*, 7–418

Manual unlock option
- See RAB$V_ULK option

MANUAL_UNLOCKING attribute • *File Def Language*, FDL–11
MANUAL_UNLOCKING secondary attribute • *File Applications*, 7–15
Map
- See Image map

MAP built-in procedure • *VAXTPU*, 7–259 to 7–260
MAP file • *Delta/XDelta*, DELTA–10, DELTA–11, DELTA–12
Mapped file • *Programming Resources*, 8–4
- closing • *Programming Resources*, 8–9
- saving • *Programming Resources*, 8–9

MAPPED_WHEN_MANAGED parameter to SET built-in procedure • *VAXTPU*, 7–418
Mapping
- controlling in relation to widget • *VAXTPU*, 7–418

/MAP qualifier • *Linker*, 1–5, 2–6, LINK–11
Map register base register
- See MBA$L_MAP

Map registers • *Device Support (A)*, 1–22, 14–3, 14–4 to 14–7, 14–15, 14–19 to 14–22; *Device Support (B)*, 1–8, 1–25, 1–26, 2–3
- allocating • *Device Support (B)*, 3–65 to 3–66
- allocating permanent • *Device Support (A)*, 11–2, 14–20 to 14–21, E–12; *Device Support (B)*, 1–25

Map registers (Cont.)
- byte offset bit • *Device Support (B)*, 3–77
- calculating the number needed • *Device Support (A)*, 14–19
- format • *Device Support (A)*, 14–6 to 14–7, 14–21
- invalidating • *Device Support (A)*, 14–7, 14–13, 14–22
- loading • *Device Support (A)*, 14–21 to 14–22; *Device Support (B)*, 2–46, 3–77 to 3–78
- number of active • *Device Support (B)*, 1–9, 1–10
- number of disabled • *Device Support (B)*, 1–10
- of MBA • *Device Support (A)*, 15–3; *Device Support (B)*, 2–45, 3–76
- of Q22 bus • *Device Support (A)*, 14–6
- of UBA • *Device Support (A)*, 14–6
- operation • *Device Support (A)*, 14–6 to 14–7
- releasing • *Device Support (A)*, 10–2, 14–26; *Device Support (B)*, 2–56, 3–89 to 3–90
- requesting • *Device Support (A)*, 14–19 to 14–21; *Device Support (B)*, 2–61, 3–98 to 3–99

Map register valid bit • *Device Support (A)*, 14–21
Map register wait queue • *Device Support (A)*, 14–19, 14–26, E–14; *Device Support (B)*, 1–8, 3–90, 3–99
"Map_count" string constant parameter to GET_INFO • *VAXTPU*, 7–173
Margin
- default • *VAXTPU*, 7–412, 7–419, 7–454
- left
  - setting records • *VAXTPU*, 7–448
- setting • *VAXTPU*, 7–412, 7–419, 7–454
- source display • *Debugger*, 6–9, CD–148, CD–228

margin action
- setting • *VAXTPU*, 7–414

Margin action
- default • *VAXTPU*, 7–414

Margin Action
- default • *VAXTPU*, 7–456
- setting • *VAXTPU*, 7–456

MARGINS keyword • *VAXTPU*, 7–419
MARK built-in procedure • *VAXTPU*, 7–261 to 7–263
MARK data type • *VAXTPU*, 2–8 to 2–10
Marker
- deleting • *VAXTPU*, 2–10, 7–108
- determining if record containing is unmodifiable • *VAXTPU*, 7–186
- fetching display value of record containing • *VAXTPU*, 7–186
- padding effects • *VAXTPU*, 2–10
- video attributes • *VAXTPU*, 2–9, 7–261

/MARK_CHANGE qualifier • *Debugger*, CD–71

Mask
- entry • *MACRO*, 9–63
- EXAMINE/FMASK command • *Debugger*, 11–13
- EXAMINE/TMASK command • *Debugger*, 11–13
- masked vector operation • *Debugger*, 11–6, 11–10, 11–13, 11–14
- register • *MACRO*, 3–13
- register, VMR • *Debugger*, 11–6, 11–10, 11–13, 11–14
- register save • *MACRO*, 6–29, 6–59

MASK directive • *MACRO*, 6–59

Masked vector operations • *MACRO*, 10–12

mask_byte data type • *Routines Intro*, A–10t

mask_longword data type • *Routines Intro*, A–10t

mask_quadword data type • *Routines Intro*, A–10t

mask_word data type • *Routines Intro*, A–10t

MASSBUS
- configuration • *Device Support (A)*, 15–1, 15–5
- I/O address space • *Device Support (A)*, 19–1
- I/O database • *Device Support (A)*, 15–4, 15–7 to 15–8
- servicing multiunit controller on • *Device Support (A)*, 15–2, 15–6, 15–8, 15–12, 15–14, 15–16
- servicing single-unit controller on • *Device Support (A)*, 15–6 to 15–8, 15–11, 15–12, 15–13, 15–16

MASSBUS adapter
- See MBA

MASSBUS driver
- DPT for • *Device Support (A)*, 15–15
- interrupt service routine • *Device Support (A)*, 15–17
- start I/O routine • *Device Support (A)*, 15–13
- unit initialization routine • *Device Support (A)*, 15–12
- unsolicited interrupt service routine • *Device Support (A)*, 15–16

Master adapter • *I/O User's I*, 6–8

Master/slave software model • *RTL Parallel Processing*, 1–3 to 1–4
- characteristics of • *RTL Parallel Processing*, 1–3
- queuing model • *RTL Parallel Processing*, 1–3
- self-scheduling model • *RTL Parallel Processing*, 1–3, 1–4
- true model • *RTL Parallel Processing*, 1–3, 1–4

MATCH built-in procedure • *VAXTPU*, 7–264 to 7–265

MATCHC (Match Characters) instruction • *MACRO*, 9–131
- RTL routine to access • *RTL Library*, LIB–270

Match operations • *Librarian*, LIB–2

Mathematical functions

Mathematical functions (Cont.)
- using system routines • *Programming Resources*, 1–24

Mathematics routine
- additional routines • *RTL Math*, A–1 to A–16

MAXBUF system parameter
- limiting size of user's ACL buffer • *RMS*, 14–3

Maximize-version option • *File Applications*, 4–27

MAXIMIZE_VERSION attribute • *File Def Language*, FDL–20

MAXIMIZE_VERSION secondary attribute • *File Applications*, 4–27

Maximum number of history records
- NCS library, specifying • *National Char Set*, NCS–24, NCS–25

Maximum record number field
- See FAB$L_MRN field

Maximum record number option • *File Applications*, 4–29

Maximum record size
- default value for remote file access • *RMS*, 5–22
- indexed file • *File Applications*, 3–22

Maximum record size field
- See FAB$W_MRS field

Maximum record size field in XABFHC
- See XAB$W_MRZ field

Maximum-record-size option • *File Applications*, 4–29

Maximum value • *RTL Math*, 1–7

Maximum version option
- See FAB$V_MXV option

"Maximum_parameters" string constant parameter to GET_INFO • *VAXTPU*, 7–190

MAX_LINES keyword • *VAXTPU*, 7–421

"Max_lines" string constant parameter to GET_ INFO • *VAXTPU*, 7–173

MAX_RECORD_NUMBER attribute • *File Def Language*, FDL–20

MAX_RECORD_NUMBER secondary attribute • *File Applications*, 4–29

MBA$INT • *Device Support (A)*, 15–15 to 15–16; *Device Support (B)*, 4–24

MBA$L_AS • *Device Support (A)*, 15–5, 15–9 to 15–10, 15–11

MBA$L_BCR • *Device Support (A)*, 15–4, 15–5, 15–14; *Device Support (B)*, 3–76

MBA$L_CAR • *Device Support (A)*, 15–5

MBA$L_CR • *Device Support (A)*, 15–5

MBA$L_CSR • *Device Support (A)*, 15–5, 15–14

MBA$L_DR • *Device Support (A)*, 15–5

MBA$L_ERB • *Device Support (A)*, 15–5, 15–12

MBA$L_MAP • *Device Support (A)*, 15–5; *Device Support (B)*, 3–76

MBA$L_SMR • *Device Support (A),* 15–5
MBA$L_SR • *Device Support (A),* 15–5, 15–11, 15–13
MBA$L_VAR • *Device Support (A),* 15–4, 15–5, 15–14, 15–15; *Device Support (B),* 3–76
MBA (MASSBUS adapter) • *Device Support (A),* 1–11
address space • *Device Support (A),* 15–4 to 15–6
data path • *Device Support (A),* 15–3
functions • *Device Support (A),* 15–1, 15–9 to 15–10
nexus value of • *Device Support (A),* 12–5
obtaining ownership • *Device Support (A),* 15–2, 15–3, 15–6 to 15–11, 15–14
registers • *Device Support (A),* 15–1 to 15–6
device • *Device Support (A),* 15–5, 15–12 to 15–13, 15–13
external • *Device Support (A),* 15–2
internal • *Device Support (A),* 15–3
map • *Device Support (A),* 15–3 to 15–6; *Device Support (B),* 2–45, 3–76
releasing secondary data channel • *Device Support (B),* 3–91
subunit number • *Device Support (A),* 15–1
unit number • *Device Support (A),* 12–6, 15–1, 15–12 to 15–13
$MBADEF macro • *Device Support (A),* 15–4 to 15–6
MBZ field • *MACRO,* 7–1
.MCALL directive • *MACRO,* 6–60
MCHECK spin lock • *Device Support (A),* 3–14
$MCHKDEF macro • *Device Support (A),* 16–13, 16–14
MCHK symbol • *System Dump Analyzer,* SDA–14
MCOMB (Move Complemented Byte) instruction • *MACRO,* 9–22
MCOML (Move Complemented Long) instruction • *MACRO,* 9–22
;M command • *Delta/XDelta,* DELTA–43
privileges required for • *Delta/XDelta,* DELTA–14
MCOMW (Move Complemented Word) instruction • *MACRO,* 9–22
.MDELETE directive • *MACRO,* 6–61
MEAN_DATA_LENGTH attribute • *File Def Language,* FDL–5
MEAN_INDEX_LENGTH attribute • *File Def Language,* FDL–5
Measurement
converting units of • *VAXTPU,* 7–50
Mechanism argument vector • *RTL Library,* 4–7, 4–11, 4–20
Mechanism array • *Programming Resources,* 9–15; *System Dump Analyzer,* SDA–17, SDA–22
Mechanism array argument • *System Services Intro,* 10–10
Mechanism entry • *Routines Intro,* 1–10; *System Services Intro,* 1–8
Media ID • *Device Support (B),* 1–80
MEGA spin lock • *Device Support (A),* 3–14
Memory
See also Buffer
See also Nonpaged pool
See Shared memory
See Vector memory
See Virtual memory zone
allocating and freeing blocks of • *RTL Library,* 5–4
allocating and freeing pages of • *RTL Library,* 5–4
allocating strings • *RTL String Manipulation,* STR–46
allocation algorithms • *RTL Library,* 5–7
deallocating strings • *RTL String Manipulation,* STR–45
detecting corruption in • *Device Support (A),* 13–23 to 13–27
detecting parity errors in • *Device Support (A),* 14–25; *Device Support (B),* 2–51
effect of debugger • *Debugger,* 3–23
error resulting from exceeding • *VAXTPU,* 5–1
examining • *System Dump Analyzer,* SDA–51
formatting • *System Dump Analyzer,* SDA–56
locking page into • *System Services Intro,* 11–7; *System Services,* SYS–335
nonpaged system dynamic • *File Applications,* 9–8
reasons for insufficient virtual memory error • *RTL Parallel Processing,* PPL–11
releasing with the FDL$RELEASE routine • *File Applications,* 4–15
testing accessibility of • *Device Support (B),* 2–39 to 2–40
unlocking page from • *System Services,* SYS–526
Memory allocation • *Linker,* 1–6, 2–10
absolute program section • *Linker,* 6–4
algorithm for • *Linker,* 6–15
based image • *Linker,* 1–7, 3–5
cluster • *Linker,* 6–17
information about, in map • *Linker,* 5–8
relocatable program section • *Linker,* 6–4
shareable image • *Linker,* 6–7
steps in • *Linker,* 6–15
system image • *Linker,* 6–2
Memory cache • *File Applications,* 3–12, 3–14
Memory fragmentation • *RTL Library,* 5–5
Memory interconnect to VAXBI adapter • *Device Support (A),* 16–1, 16–7, 16–10
ADP address • *Device Support (A),* 16–10

- Memory location
  - decoding • *System Dump Analyzer*, SDA–53
  - examining • *System Dump Analyzer*, SDA–52
- Memory management • *Programming Resources*, 10–1
  - exception • *MACRO*, E–4
  - fault • *MACRO*, E–4
  - using system routines • *Programming Resources*, 1–23
  - vector • *MACRO*, 10–47
    - memory management disabled • *MACRO*, 10–47
    - TB • *MACRO*, 10–7, 10–8, 10–20, 10–32, 10–34, 10–41, 10–47
  - virtual memory • *Programming Resources*, 1–23
- Memory management exceptions
  - vector • *MACRO*, 10–28
    - asynchronous MME handling • *MACRO*, 10–30
    - fault parameter • *MACRO*, 10–28
      - PTE bit • *MACRO*, 10–29
      - VAL bit • *MACRO*, 10–29
      - VAS bit • *MACRO*, 10–29
      - VIO bit • *MACRO*, 10–29
    - fault stack frame • *MACRO*, 10–28
    - synchronous MME handling • *MACRO*, 10–30
    - system control block (SCB) • *MACRO*, 10–28
- Memory management resources
  - synchronizing access to • *Device Support (A)*, 3–13
- Memory management services • *System Services Intro*, 1–2; *RTL Library*, 5–3
- Memory region
  - examining • *System Dump Analyzer*, SDA–54
- Memory synchronization
  - required use of • *MACRO*, 10–42
- Menu • *RTL Screen Management*, 2–14
  - creating • *RTL Screen Management*, 2–14
  - creating with SMG$ routines • *Programming Resources*, 7–22
  - deleting • *RTL Screen Management*, 2–14
  - reading • *Programming Resources*, 7–23
  - selecting • *RTL Screen Management*, 2–15
- Menu bar widget • *VAXTPU*, 4–16
- Menu position
  - of widget
    - fetching in VAXTPU • *VAXTPU*, 7–210
  - setting in VAXTPU • *VAXTPU*, 7–422
- MENU_POSITION parameter to SET built-in procedure • *VAXTPU*, 7–422
- “menu_position” string constant parameter to GET_ INFO • *VAXTPU*, 7–210
- MERGE command • *Programming Resources*, 8–13
  - file interface • *Programming Resources*, 8–19
  - record interface • *Programming Resources*, 8–21
- /MERGE qualifier • *Convert*, CONV–1, CONV–17
- Message
  - chaining • *Programming Resources*, 9–23
  - construction of • *Message*, MSG–2
  - debugger • *Debugger*, 2–8, CD–5
    - with DECwindows • *Debugger*, 1–20
  - definition of • *Message*, MSG–22
  - displaying • *Programming Resources*, 9–22
  - example of • *Message*, MSG–1
  - format of • *Message*, MSG–1
  - formatting and outputting • *System Services*, SYS–371
  - logging • *Programming Resources*, 9–24
  - obtaining text of • *System Services*, SYS–253
  - sending to error logger • *System Services*, SYS–441
  - sending to operator • *System Services*, SYS–495
  - system • *System Services Intro*, 2–17
  - writing to terminal • *System Services*, SYS–30, SYS–38
- MESSAGE
  - See Message Utility
- Message buffer • *VAXTPU*, 4–18
- MESSAGE built-in procedure • *VAXTPU*, 7–266 to 7–269
- Message code • *Message*, MSG–2
- MESSAGE command • *Message*, MSG–4, MSG–9
  - format of • *Message*, MSG–8
  - parameter for • *Message*, MSG–8
  - qualifiers • *Message*, MSG–8 to MSG–14
- MESSAGE commands • *Message*, MSG–15
- Message definition
  - in message source file • *Message*, MSG–22
  - qualifiers for • *Message*, MSG–22, MSG–23
  - statements • *Message*, MSG–3
- Message display directive
  - (.ERROR) • *MACRO*, 6–31
  - (.PRINT) • *MACRO*, 6–76
- Message examples • *Message*, MSG–29
- Message file
  - See also Nonexecutable message file
- Message format
  - See Mailbox
- Message object module
  - linking • *Message*, MSG–4
- Message pointer
  - creating • *Message*, MSG–5
  - example • *Message*, MSG–29

Message pointer (Cont.)
use of • *Message*, MSG–4, MSG–5
Messages • *SUMSLP*, SUM–13; *VAXTPU*, D–1 to D–10
Message source file
comments in • *Message*, MSG–7
compiling • *Message*, MSG–4
elements of • *Message*, MSG–3
expressions in • *Message*, MSG–7
format • *Message*, MSG–3
sample of • *Message*, MSG–18
symbols in • *Message*, MSG–7
Message source file statements • *Message*, MSG–6, MSG–15
base message number directive (.BASE) • *Message*, MSG–16
end directive (.END) • *Message*, MSG–17
facility directive (.FACILITY) • *Message*, MSG–18
identification directive (.IDENT) • *Message*, MSG–20
listing directives • *Message*, MSG–25, MSG–28
literal directive (.LITERAL) • *Message*, MSG–21
message definition • *Message*, MSG–22
page directive (.PAGE) • *Message*, MSG–25
severity directive (.SEVERITY) • *Message*, MSG–26
title directive (.TITLE) • *Message*, MSG–7, MSG–28
Message symbol • *Message*, MSG–2, MSG–6, MSG–22; *System Services*, SYS–376
Message text
specifying variables in • *Programming Resources*, 9–9
Message Utility (MESSAGE) • *Programming Resources*, 1–19, 9–7; *RTL Library*, 4–26 to 4–28
accessing message object module • *Programming Resources*, 9–10
command qualifiers • *Message*, MSG–9 to MSG–28
compiling message file • *Programming Resources*, 9–9
compiling the message source file • *Message*, MSG–4
constructing messages • *Message*, MSG–2
controlling output • *Message*, MSG–9
creating a message object library • *Programming Resources*, 9–10
definition statements • *Programming Resources*, 1–19
directives • *Programming Resources*, 1–19
.END • *Programming Resources*, 9–8
examples • *Message*, MSG–28
Message Utility (MESSAGE)
examples (Cont.)
creating pointer files • *Message*, MSG–29
image containing message data • *Message*, MSG–29
exiting • *Message*, MSG–8
.FACILITY • *Programming Resources*, 9–8
facility name • *Programming Resources*, 9–8
facility number • *Programming Resources*, 9–8
FAO parameters • *Programming Resources*, 9–12
/FAO_COUNT • *Programming Resources*, 9–9
invoking • *Message*, MSG–8
linking the message object module • *Message*, MSG–4
logging messages • *Programming Resources*, 9–24
message object module • *Programming Resources*, 9–9
messages
creating • *Programming Resources*, 1–19
message source file • *Message*, MSG–3
message text • *Programming Resources*, 9–9
message text variables • *Programming Resources*, 9–9
modifying a message source file • *Programming Resources*, 9–10
program example • *Message*, MSG–3
SET MESSAGE command • *Message*, MSG–5
.SEVERITY • *Programming Resources*, 9–8
source file • *Programming Resources*, 1–19
source module • *Programming Resources*, 9–7
.TITLE • *Programming Resources*, 9–9
using message pointers • *Message*, MSG–4
Message warning display directive (.WARN) • *MACRO*, 6–99
Message window
in EVE editor • *VAXTPU*, 4–16
MESSAGE_ACTION_LEVEL keyword • *VAXTPU*, 7–424
"Message_action_level" string constant parameter to GET_INFO • *VAXTPU*, 7–206
MESSAGE_ACTION_TYPE keyword • *VAXTPU*, 7–426
MESSAGE_BUFFER identifier • *VAXTPU*, 7–266
MESSAGE_BUFFER variable • *VAXTPU*, 4–29
MESSAGE_FLAGS keyword • *VAXTPU*, 7–427
"Message_flags" string constant parameter to GET_INFO • *VAXTPU*, 7–207
MESSAGE_ROUTINES.EXE
global symbols • *System Dump Analyzer*, SDA–61
MESSAGE_TEXT built-in procedure • *VAXTPU*, 7–270 to 7–272

.MEXIT directive • *MACRO,* 6–62
MFD (master file directory) • *File Applications,* 6–12
MFPR (Move from Processor Register) instruction • *MACRO,* 9–196
vector IPRs • *MACRO,* 10–3, 10–8, 10–32
VPSR • *MACRO,* 10–6, 10–31, 10–41
MFVP (Move from Vector Processor) instruction • *MACRO,* 10–19, 10–35
MicroVAX
See Workstation
MicroVAX/VAXstation 3100 systems
support for SCSI devices • *Device Support (A),* 1–18
MicroVAX 2000
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–5
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
MicroVAX 3500
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–5
requesting interrupt • *Delta/XDelta,* DELTA–7
MicroVAX 3600
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–5
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
MicroVAX I
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–5
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
MicroVAX II
adapter logic • *Device Support (A),* 14–1
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–5
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
"Middle_of_tab" string constant parameter to GET_ INFO • *VAXTPU,* 7–223
Minimal interface example • *VAXTPU,* 4–26
Minimum record length field
See also XAB$W_MRL field
in XABKEY • *RMS,* 13–12
Minimum value • *RTL Math,* 1–7
"Minimum_parameters" string constant parameter to GET_INFO • *VAXTPU,* 7–190
Minor ID • *Linker,* 3–7
Minor ID (Cont.)
of shareable image in map • *Linker,* 5–6
Miscellaneous data type • *Routines Intro,* 2–18
Mixed I/O
precautions listed • *RMS,* 4–24
MMG$GL_SBICONF • *Device Support (A),* 16–8
MMG$IOLOCK • *Device Support (B),* 3–33, 3–35, 3–41, 3–46, 3–55, 3–59
MMG$UNLOCK • *Device Support (B),* 1–43, 3–109
MMG spin lock • *Device Support (A),* 3–13; *Device Support (B),* 3–16, 3–107, 3–108, 3–109
MMS (Module Management System) • *Modular Procedures,* 1–12
MNEGB (Move Negated Byte) instruction • *MACRO,* 9–23
MNEGD (Move Negated D_floating) instruction • *MACRO,* 9–117
MNEGF (Move Negated F_floating) instruction • *MACRO,* 9–117
MNEGG (Move Negated G_floating) instruction • *MACRO,* 9–117
MNEGH (Move Negated H_floating) instruction • *MACRO,* 9–117
MNEGL (Move Negated Long) instruction • *MACRO,* 9–23
MNEGW (Move Negated Word) instruction • *MACRO,* 9–23
MNT (module name table) • *Librarian,* LIB–2
Mode
CANCEL MODE command • *Debugger,* CD–23
interactive • *File Applications,* 10–11
locate
performance • *File Applications,* 9–9
SET MODE [NO]DYNAMIC command • *Debugger,* 5–7, 5–14, CD–152
SET MODE [NO]G_FLOAT command • *Debugger,* CD–152
SET MODE [NO]INTERRUPT command • *Debugger,* CD–152
SET MODE [NO]KEYPAD command • *Debugger,* 8–8, CD–153
SET MODE [NO]LINE command • *Debugger,* CD–153
SET MODE [NO]OPERANDS command • *Debugger,* 4–20, CD–153
SET MODE [NO]SCREEN command • *Debugger,* 7–1, CD–154
SET MODE [NO]SCROLL command • *Debugger,* CD–154
SET MODE [NO]SEPARATE command • *Debugger,* 9–5, CD–154
SET MODE [NO]SYMBOLIC command • *Debugger,* 4–14, CD–154

Mode (Cont.)
  SHOW MODE • *Debugger,* CD–230
Mode card
  026 punch mode • *I/O User's I,* 2–2
  029 punch mode • *I/O User's I,* 2–2
Mode field in XABITM
  See XAB$L_MODE field
Modem signals
  input transitions of • *Device Support (A),* 18–15
  sending to device • *Device Support (A),* 18–13
Mode qualifier, PATCH command • *Patch,* PAT–15, PAT–76
"Mode" string constant parameter to GET_INFO • *VAXTPU,* 7–173
Mode switching
  when permitted • *RMS,* 4–24
Modifiability
  setting records • *VAXTPU,* 7–448
MODIFIABLE keyword • *VAXTPU,* 7–429
"Modifiable" string constant parameter to GET_INFO • *VAXTPU,* 7–173
MODIFICATIONS keyword
  using in collating sequence expression • *National Char Set,* NCS–14
  using in conversion function expression • *National Char Set,* NCS–16
MODIFICATIONS keyword clause • *National Char Set,* NCS–17
Modified page list
  displaying • *System Dump Analyzer,* SDA–115
/MODIFIED qualifier • *System Dump Analyzer,* SDA–115
"Modified" string constant parameter to GET_INFO • *VAXTPU,* 7–173
Modify access type • *MACRO,* 8–17
MODIFY command • *File Applications,* 10–28; *File Def Language,* FDL–64
  Edit/FDL Utility • *File Applications,* A–1
Modify-fault
  vector • *MACRO,* 10–47
Modify file function • *I/O User's I,* 1–28
Modify function
  FDT routine for • *Device Support (A),* 7–9
/MODIFY qualifier • *Debugger,* CD–132, CD–189; *VAXTPU,* 5–12
"Modify" string constant parameter to GET_INFO • *VAXTPU,* 7–177
MODIFY_RANGE built-in procedure • *VAXTPU,* 7–273 to 7–277
Modularity
  virtual displays • *Programming Resources,* 7–31
Modular programming • *Linker,* 2–1

Module • *Debugger,* 2–5
  See also Shareable image
  canceling • *Debugger,* 5–7, CD–24
  creating • *Librarian,* LIB–4
  finding a failing • *System Dump Analyzer,* SDA–24
  formatting • *Librarian,* LIB–5
  information about • *Debugger,* 5–7, CD–231
  key number in • *Librarian,* LIB–5
  replacing in the default NCS library • *National Char Set,* NCS–21
  setting • *Debugger,* 5–6, CD–156
    with DECwindows • *Debugger,* 1–27
  terminating • *Librarian,* LIB–5
  traceback information • *Debugger,* 5–3
  with DECwindows • *Debugger,* 1–3
Module declaration
  syntax • *VAXTPU,* 3–15
Module header • *Librarian,* LIB–2
Module Management System
  See MMS
Module name
  made available to debugger • *MACRO,* 6–23
Module name table
  See MNT
/MODULE qualifier • *Debugger,* CD–28, CD–171, CD–176; *Librarian,* LIB–32
  using with /INSERT • *Librarian,* LIB–32
MODULE statement • *Command Def,* CDU–14, CDU–37; *VAXTPU,* 3–14 to 3–15
Modules used with EVE$BUILD • *VAXTPU,* G–2
Monitoring procedures • *Modular Procedures,* 4–8, A–5
  in the Run-Time Library • *Modular Procedures,* 4–9
  timer • *Modular Procedures,* 4–8
MOUNT command • *I/O User's I,* 6–27
  and window size • *File Applications,* 9–8
Mount function • *I/O User's I,* 1–30
MOUNT privilege • *System Services Intro,* 7–4
Mount verification • *Device Support (B),* 1–40, 1–78
Mount verification routine • *Device Support (B),* 1–30, 1–31
Mouse
  determining support for • *VAXTPU,* 7–432
  determining where drag operation originated • *VAXTPU,* 7–188
Mouse button
  fetching information about • *VAXTPU,* 7–188
MOUSE keyword • *VAXTPU,* 7–432
  with POSITION • *VAXTPU,* 7–288, 7–289
Mouse pad
  implementing • *VAXTPU,* B–4

"Mouse" string constant parameter to GET_INFO • *VAXTPU,* 7–200
MOVAB (Move Address Byte) instruction • *MACRO,* 9–34
MOVAD (Move Address D_floating) instruction • *MACRO,* 9–34
MOVAF (Move Address F_floating) instruction • *MACRO,* 9–34
MOVAG (Move Address G_floating) instruction • *MACRO,* 9–34
MOVAH (Move Address H_floating) instruction • *MACRO,* 9–34
MOVAL (Move Address Long) instruction • *MACRO,* 9–34
MOVAO (Move Address Octa) instruction • *MACRO,* 9–34
MOVAQ (Move Address Quad) instruction • *MACRO,* 9–34
MOVAW (Move Address Word) instruction • *MACRO,* 9–34
MOVB (Move Byte) instruction • *MACRO,* 9–24
MOVC3 (Move Character 3 Operand) instruction • *MACRO,* 9–132
  RTL routine to access • *RTL Library,* LIB–275
MOVC5 (Move Character 5 Operand) instruction • *MACRO,* 9–132
  RTL routine to access • *RTL Library,* LIB–276
MOVD (Move D_floating) instruction • *MACRO,* 9–118
MOVE command • *Debugger,* 7–13, CD–110
MOVE_HORIZONTAL built-in procedure • *VAXTPU,* 7–278 to 7–279
MOVE_TEXT built-in procedure • *VAXTPU,* 7–280 to 7–281
MOVE_VERTICAL built-in procedure • *VAXTPU,* 7–282 to 7–283
MOVF (Move F_floating) instruction • *MACRO,* 9–118
MOVG (Move G_floating) instruction • *MACRO,* 9–118
MOVH (Move H_floating) instruction • *MACRO,* 9–118
MOVL (Move Long) instruction • *MACRO,* 9–24
MOVO (Move Octa) instruction • *MACRO,* 9–24
MOVP (Move Packed) instruction • *MACRO,* 9–165
MOVPSL (Move PSL) instruction • *MACRO,* 9–77
MOVQ (Move Quad) instruction • *MACRO,* 9–24
MOVTC (Move Translated Characters) instruction • *MACRO,* 9–134
MOVTUC (Move Translated Until Character) instruction • *MACRO,* 9–136
MOVW (Move Word) instruction • *MACRO,* 9–24
MOVZBL (Move Zero-Extended Byte to Long) instruction • *MACRO,* 9–25
MOVZBW (Move Zero-Extended Byte to Word) instruction • *MACRO,* 9–25
MOVZWL (Move Zero-Extended Word to Long) instruction • *MACRO,* 9–25
MSCP server
  code
    base address • *System Dump Analyzer,* SDA–14
MSCP symbol • *System Dump Analyzer,* SDA–14
MSE option • *File Def Language,* FDL–37
MSG$_CRUNSOLIC • *Device Support (A),* 9–7
MSG$_DEVOFFLIN • *Device Support (A),* 10–7
MSYNC (Memory Instruction Synchronization) instruction • *MACRO,* 10–35, 10–39, 10–42, 10–44, 10–88
MTH$ACOS • *RTL Math,* MTH–3
MTH$ACOSD • *RTL Math,* MTH–6
MTH$AIMAG • *RTL Math,* MTH–110
MTH$ALOG • *RTL Math,* MTH–112
MTH$ALOG10 • *RTL Math,* MTH–116
MTH$ALOG2 • *RTL Math,* MTH–114
MTH$ASIN • *RTL Math,* MTH–9
MTH$ASIND • *RTL Math,* MTH–11
MTH$ATAN • *RTL Math,* MTH–13
MTH$ATAN2 • *RTL Math,* MTH–17
MTH$ATAND • *RTL Math,* MTH–15
MTH$ATAND2 • *RTL Math,* MTH–19
MTH$ATANH • *RTL Math,* MTH–21
MTH$CABS • *RTL Math,* MTH–23
MTH$CCOS • *RTL Math,* MTH–26
MTH$CDABS • *RTL Math,* MTH–23
MTH$CDCOS • *RTL Math,* MTH–28
MTH$CDEXP • *RTL Math,* MTH–33
MTH$CDLOG • *RTL Math,* MTH–37
MTH$CDSIN • *RTL Math,* MTH–54
MTH$CDSQRT • *RTL Math,* MTH–59
MTH$CEXP • *RTL Math,* MTH–31
MTH$CGABS • *RTL Math,* MTH–23
MTH$CGCOS • *RTL Math,* MTH–28
MTH$CGEXP • *RTL Math,* MTH–33
MTH$CGLOG • *RTL Math,* MTH–37
MTH$CGSIN • *RTL Math,* MTH–54
MTH$CGSQRT • *RTL Math,* MTH–59
MTH$CLOG • *RTL Math,* MTH–35
MTH$CMPLX • *RTL Math,* MTH–40
MTH$CONJG • *RTL Math,* MTH–44
MTH$COS • *RTL Math,* MTH–47
MTH$COSD • *RTL Math,* MTH–49
MTH$COSH • *RTL Math,* MTH–51
MTH$CSIN • *RTL Math,* MTH–53
MTH$CSQRT • *RTL Math,* MTH–57
MTH$CVT_DA_GA • *RTL Math,* MTH–63
MTH$CVT_D_G • *RTL Math,* MTH–62

MTH$CVT_GA_DA • *RTL Math*, MTH–63
MTH$CVT_G_D • *RTL Math*, MTH–62
MTH$DACOS • *RTL Math*, MTH–3
MTH$DACOSD • *RTL Math*, MTH–6
MTH$DASIN • *RTL Math*, MTH–9
MTH$DASIND • *RTL Math*, MTH–11
MTH$DATAN • *RTL Math*, MTH–13
MTH$DATAN2 • *RTL Math*, MTH–17
MTH$DATAND • *RTL Math*, MTH–15
MTH$DATAND2 • *RTL Math*, MTH–19
MTH$DATANH • *RTL Math*, MTH–21
MTH$DCMPLX • *RTL Math*, MTH–42
MTH$DCONJG • *RTL Math*, MTH–45
MTH$DCOS • *RTL Math*, MTH–47
MTH$DCOSD • *RTL Math*, MTH–49
MTH$DCOSH • *RTL Math*, MTH–51
MTH$DEXP • *RTL Math*, MTH–65
MTH$DIMAG • *RTL Math*, MTH–110
MTH$DLOG • *RTL Math*, MTH–112
MTH$DLOG10 • *RTL Math*, MTH–116
MTH$DLOG2 • *RTL Math*, MTH–114
MTH$DREAL • *RTL Math*, MTH–120
MTH$DSIN • *RTL Math*, MTH–122
MTH$DSINCOS • *RTL Math*, MTH–124
MTH$DSINCOSD • *RTL Math*, MTH–127
MTH$DSIND • *RTL Math*, MTH–131
MTH$DSINH • *RTL Math*, MTH–133
MTH$DSQRT • *RTL Math*, MTH–136
MTH$DTAN • *RTL Math*, MTH–139
MTH$DTAND • *RTL Math*, MTH–141
MTH$DTANH • *RTL Math*, MTH–143
MTH$EXP • *RTL Math*, MTH–65
MTH$GACOS • *RTL Math*, MTH–3
MTH$GACOSD • *RTL Math*, MTH–6
MTH$GASIN • *RTL Math*, MTH–9
MTH$GASIND • *RTL Math*, MTH–11
MTH$GATAN • *RTL Math*, MTH–13
MTH$GATAN2 • *RTL Math*, MTH–17
MTH$GATAND • *RTL Math*, MTH–15
MTH$GATAND2 • *RTL Math*, MTH–19
MTH$GATANH • *RTL Math*, MTH–21
MTH$GCMPLX • *RTL Math*, MTH–42
MTH$GCONJG • *RTL Math*, MTH–45
MTH$GCOS • *RTL Math*, MTH–47
MTH$GCOSD • *RTL Math*, MTH–49
MTH$GCOSH • *RTL Math*, MTH–51
MTH$GEXP • *RTL Math*, MTH–65
MTH$GIMAG • *RTL Math*, MTH–110
MTH$GLOG • *RTL Math*, MTH–112
MTH$GLOG10 • *RTL Math*, MTH–116
MTH$GLOG2 • *RTL Math*, MTH–114
MTH$GREAL • *RTL Math*, MTH–120
MTH$GSIN • *RTL Math*, MTH–122
MTH$GSINCOS • *RTL Math*, MTH–124
MTH$GSINCOSD • *RTL Math*, MTH–127
MTH$GSIND • *RTL Math*, MTH–131
MTH$GSINH • *RTL Math*, MTH–133
MTH$GSQRT • *RTL Math*, MTH–136
MTH$GTAN • *RTL Math*, MTH–139
MTH$GTAND • *RTL Math*, MTH–141
MTH$GTANH • *RTL Math*, MTH–143
MTH$HACOS • *RTL Math*, MTH–68
MTH$HACOSD • *RTL Math*, MTH–70
MTH$HASIN • *RTL Math*, MTH–72
MTH$HASIND • *RTL Math*, MTH–74
MTH$HATAN • *RTL Math*, MTH–76
MTH$HATAN2 • *RTL Math*, MTH–80
MTH$HATAND • *RTL Math*, MTH–78
MTH$HATAND2 • *RTL Math*, MTH–82
MTH$HATANH • *RTL Math*, MTH–84
MTH$HCOS • *RTL Math*, MTH–86
MTH$HCOSD • *RTL Math*, MTH–87
MTH$HCOSH • *RTL Math*, MTH–88
MTH$HEXP • *RTL Math*, MTH–90
MTH$HLOG • *RTL Math*, MTH–92
MTH$HLOG10 • *RTL Math*, MTH–96
MTH$HLOG2 • *RTL Math*, MTH–94
MTH$HSIN • *RTL Math*, MTH–98
MTH$HSINCOS • *RTL Math*, MTH–124
MTH$HSINCOSD • *RTL Math*, MTH–127
MTH$HSIND • *RTL Math*, MTH–99
MTH$HSINH • *RTL Math*, MTH–100
MTH$HSQRT • *RTL Math*, MTH–102
MTH$HTAN • *RTL Math*, MTH–104
MTH$HTAND • *RTL Math*, MTH–106
MTH$HTANH • *RTL Math*, MTH–108
MTH$RANDOM • *RTL Math*, MTH–118
MTH$REAL • *RTL Math*, MTH–120
MTH$SIN • *RTL Math*, MTH–122
MTH$SINCOS • *RTL Math*, MTH–124
MTH$SINCOSD • *RTL Math*, MTH–127
MTH$SIND • *RTL Math*, MTH–131
MTH$SINH • *RTL Math*, MTH–133
MTH$SIN_R4 • *RTL Intro*, 3–5
MTH$SQRT • *RTL Math*, MTH–136
MTH$TAN • *RTL Math*, MTH–139
MTH$TAND • *RTL Math*, MTH–141
MTH$TANH • *RTL Math*, MTH–143
MTH$UMAX • *RTL Math*, MTH–145
MTH$UMIN • *RTL Math*, MTH–146
MTH$VxFOLRLy_MA_V5 • *RTL Math*, MTH–201
MTH$VxFOLRLy_z_V2 • *RTL Math*, MTH–205

MTH$VxFOLRy_MA_V15 • *RTL Math*, MTH–192
MTH$VxFOLRy_z_V8 • *RTL Math*, MTH–197
MTPR (Move to Processor Register) instruction • *MACRO*, 9–195, 10–47
  vector IPRs • *MACRO*, 10–8, 10–47
MTVP (Move to Vector Processor) instruction • *MACRO*, 10–90
MT_BLOCK_SIZE attribute • *File Def Language*, FDL–21
MT_BLOCK_SIZE secondary attribute • *File Applications*, 4–28
MT_CLOSE_REWIND attribute • *File Def Language*, FDL–21
MT_CURRENT_POSITION attribute • *File Def Language*, FDL–21
MT_NOT_EOF attribute • *File Def Language*, FDL–21
MT_OPEN_REWIND attribute • *File Def Language*, FDL–21
MT_PROTECTION attribute • *File Def Language*, FDL–22
MT_PROTECTION secondary attribute • *File Applications*, 4–28
MULB2 (Multiply Byte 2 Operand) instruction • *MACRO*, 9–26
MULB3 (Multiply Byte 3 Operand) instruction • *MACRO*, 9–26
MULD2 (Multiply D_floating 2 Operand) instruction • *MACRO*, 9–119
MULD3 (Multiply D_floating 3 Operand) instruction • *MACRO*, 9–119
MULF2 (Multiply F_floating 2 Operand) instruction • *MACRO*, 9–119
MULF3 (Multiply F_floating 3 Operand) instruction • *MACRO*, 9–119
MULG2 (Multiply G_floating 2 Operand) instruction • *MACRO*, 9–119
MULG3 (Multiply G_floating 3 Operand) instruction • *MACRO*, 9–119
MULH2 (Multiply H_floating 2 Operand) instruction • *MACRO*, 9–119
MULH3 (Multiply H_floating 3 Operand) instruction • *MACRO*, 9–119
MULL2 (Multiply Long 2 Operand) instruction • *MACRO*, 9–26
MULL3 (Multiply Long 3 Operand) instruction • *MACRO*, 9–26
MULP (Multiply Packed) instruction • *MACRO*, 9–166
Multiblock • *File Applications*, 3–11
  defined • *File Applications*, 2–1, 3–6
  restriction for use • *File Applications*, 3–6
Multiblock count field
  See RAB$B_MBC field
MULTIBLOCK_COUNT attribute • *File Def Language*, FDL–12
MULTIBLOCK_COUNT secondary attribute • *File Applications*, 7–18
Multibuffer count • *File Applications*, 3–11, 3–13, 3–26, 3–27
Multibuffer count field
  See RAB$B_MBF field
MULTIBUFFER_COUNT attribute • *File Def Language*, FDL–12
MULTIBUFFER_COUNT secondary attribute • *File Applications*, 7–17, 7–19
  and record access type • *File Applications*, 7–20
  for sequential file • *File Applications*, 7–18
Multilanguage program
  debugging • *Debugger*, 9–7
    with DECwindows • *Debugger*, 1–29
Multilevel device interrupt dispatching • *Device Support (A)*, 14–31, 14–33 to 14–36; *Device Support (B)*, 1–22
Multinational character set
  See DEC Multinational Character Set
Multinational Character Set
  See DEC Multinational Character Set
Multiple active signal • *Routines Intro*, 2–54
Multiple area
  See Area
Multiple areas • *File Def Language*, FDL–6, FDL–28
Multiple argument
  delimiting in control block fields • *RMS*, 3–5, 3–7
  specifying in control block fields • *RMS*, B–3
Multiple buffers • *VAXTPU*, 7–59
Multiple definition modules
  specifying with /DELETE qualifier • *National Char Set*, NCS–27, NCS–32
  specifying with /EXTRACT qualifier • *National Char Set*, NCS–28
  specifying with /ONLY qualifier • *National Char Set*, NCS–38
Multiple exception • *System Services Intro*, 10–17
Multiple input files • *Convert*, CONV–5
  specifying • *National Char Set*, NCS–21
Multiple-key indexed file
  creating • *RMS*, 4–5
Multiple keys • *Convert*, CONV–27
  example of use with Close service • *RMS*, 4–12
  performance cost of using • *RMS*, 13–14
  recommended number • *RMS*, 13–14
Multiple record stream
  with block I/O • *RMS*, 4–25
Multiple service
  for retrieving records • *File Applications*, 8–3

Multiplexer
  DMB32 device • *I/O User's I*, 8–1
  DMF32 device • *I/O User's I*, 8–1
  DZ11 device • *I/O User's I*, 8–1
  DZ32 device • *I/O User's I*, 8–1
Multiplication • *RTL Library*, LIB–128, LIB–130, LIB–132, LIB–134
  decimal strings • *RTL String Manipulation*, STR–58
  extended precision • *RTL Library*, LIB–136
  of complex number • *RTL General Purpose*, OTS–53
Multiplication operator (*) • *System Dump Analyzer*, SDA–12
Multiplying
  vector • *RTL Math*, MTH–155
Multiprocessing
  global symbols • *System Dump Analyzer*, SDA–61
Multiprocessing device driver
  analyzing crash dumps • *Device Support (A)*, E–19 to E–20
  incompatibility with uniprocessing driver • *Device Support (A)*, 12–13, E–3
  using XDELTA • *Device Support (A)*, 13–7, E–20
  writing • *Device Support (A)*, E–8 to E–20
Multiprocessing environment • *Programming Resources*, 4–18
  See also Synchronization
  contrasted with uniprocessing environment • *Device Support (A)*, 3–11, E–1
  debugging a driver designed for • *Device Support (A)*, 13–28 to 13–30
  initial XDELTA breakpoint • *Delta/XDelta*, DELTA–8
  scheduling • *Programming Resources*, 4–19
  XDELTA breakpoints • *Delta/XDelta*, DELTA–13, DELTA–29, DELTA–35
  XDELTA operation • *Delta/XDelta*, DELTA–13
MULTIPROCESSING parameter • *Device Support (A)*, 13–28, E–2 to E–3, E–4
Multiprocessing software model
  master/slave • *RTL Parallel Processing*, 1–3 to 1–4
  pipelining • *RTL Parallel Processing*, 1–4 to 1–5
  work queue processing • *RTL Parallel Processing*, 1–5
Multiprocessor
  analyzing crash dumps • *System Dump Analyzer*, SDA–9
  displaying synchronization structures • *System Dump Analyzer*, SDA–150
Multiprocessor state • *Device Support (B)*, 1–16
Multiprocess program
  CALL command • *Debugger*, CD–10

Multiprocess program (Cont.)
  CONNECT command • *Debugger*, 10–4, 10–16, CD–36
  controlling execution • *Debugger*, 10–6
  DBG$PROCESS • *Debugger*, 10–10
  debugging • *Debugger*, 10–1
    with DECwindows • *Debugger*, 1–8, 1–30
  DEFINE/PROCESS_GROUP command • *Debugger*, CD–54
  DO command • *Debugger*, 10–5, CD–76
  EXIT command • *Debugger*, 10–9, 10–10, CD–94
    with DECwindows • *Debugger*, 1–21
  global section watchpoint • *Debugger*, 10–17
  GO command • *Debugger*, 10–6, CD–105
  QUIT command • *Debugger*, 10–9, 10–10, CD–112
    with DECwindows • *Debugger*, 1–21
  screen mode features • *Debugger*, 10–16
  SET MODE [NO]INTERRUPT command • *Debugger*, 10–7, CD–152
  SET PROCESS command • *Debugger*, 10–7, 10–8, CD–161
  SHOW PROCESS command • *Debugger*, 10–3, CD–235
  Specifying processes • *Debugger*, 10–12
  STEP command • *Debugger*, 10–6, CD–265
  system requirements • *Debugger*, 10–20
  with DECwindows • *Debugger*, 1–8, 1–30
Multiprogramming • *RTL Parallel Processing*, 1–1
  timesharing • *RTL Parallel Processing*, 1–1
Multistream access option
  See FAB$V_MSE option
MULTISTREAM attribute • *File Def Language*, FDL–37
Multistreamed workload • *Programming Resources*, 4–18
MULTISTREAM secondary attribute • *File Applications*, 7–4
MULW2 (Multiply Word 2 Operand) instruction • *MACRO*, 9–26
MULW3 (Multiply Word 3 Operand) instruction • *MACRO*, 9–26
Must Be Zero
  See also MBZ
  See Field
Mutex
  for ACL • *Device Support (B)*, 1–45
  for I/O database • *Device Support (B)*, 4–6
  I/O database • *Device Support (A)*, 11–12
Mutual exclusion
  definition of • *RTL Parallel Processing*, 1–2
  semaphore • *RTL Parallel Processing*, 4–9
MXV option • *File Def Language*, FDL–21

N

NAM$B_BID field • *RMS*, 6–4
NAM$B_BLN field • *RMS*, 6–4
NAM$B_DEV descriptor • *RMS*, 6–3
NAM$B_DEV field • *RMS*, 6–4
NAM$B_DIR descriptor • *RMS*, 6–3
NAM$B_DIR field • *RMS*, 6–5
NAM$B_ESL field • *RMS*, 6–5
NAM$B_ESS field • *RMS*, 6–5
NAM$B_NAME descriptor • *RMS*, 6–3
NAM$B_NAME field • *RMS*, 6–7
NAM$B_NODE descriptor • *RMS*, 6–3
NAM$B_NODE field • *RMS*, 6–7
NAM$B_NOP field • *RMS*, 6–7
  options listed • *RMS*, 6–8
NAM$B_RSL field • *RMS*, 6–9, RMS–63
NAM$B_RSS field • *File Applications*, 6–9; *RMS*, 6–9
NAM$B_TYPE descriptor • *RMS*, 6–3
NAM$B_TYPE field • *RMS*, 6–9
NAM$B_VER descriptor • *RMS*, 6–3
NAM$B_VER field • *RMS*, 6–10
NAM$L_DEV descriptor • *RMS*, 6–3
NAM$L_DEV field • *RMS*, 6–4
NAM$L_DIR descriptor • *RMS*, 6–3
NAM$L_DIR field • *RMS*, 6–5
NAM$L_ESA field • *File Applications*, 6–4; *RMS*, 6–5
NAM$L_FNB field • *RMS*, 6–6, RMS–63, RMS–87
NAM$L_FNB status bit
  listing • *RMS*, 6–6
NAM$L_NAME descriptor • *RMS*, 6–3
NAM$L_NAME field • *RMS*, 6–7
NAM$L_NODE descriptor • *RMS*, 6–3
NAM$L_NODE field • *RMS*, 6–7
NAM$L_RLF field • *File Applications*, 6–4, 6–9, 9–7; *RMS*, 6–8
NAM$L_RSA field • *File Applications*, 6–4, 6–9; *RMS*, 6–9, RMS–63
NAM$L_TYPE descriptor • *RMS*, 6–3
NAM$L_TYPE field • *RMS*, 6–9
NAM$L_VER descriptor • *RMS*, 6–3
NAM$L_VER field • *RMS*, 6–10
NAM$L_WCC field • *RMS*, 6–10
  returned by Remove service • *RMS*, RMS–82
NAM$T_DVI field • *File Applications*, 6–5; *RMS*, 6–5
NAM$V_CNCL_DEV bit • *RMS*, 6–6
NAM$V_CONCEAL field • *RMS*, RMS–26, RMS–63
NAM$V_DIR_LVLS bit • *RMS*, 6–6
NAM$V_EXP_DEV bit • *RMS*, 6–6
NAM$V_EXP_DIR bit • *RMS*, 6–6
NAM$V_EXP_NAME bit • *RMS*, 6–6
NAM$V_EXP_TYPE bit • *RMS*, 6–6
NAM$V_EXP_VER bit • *RMS*, 6–6
NAM$V_GRP_MBR bit • *RMS*, 6–6
NAM$V_HIGHVER bit • *RMS*, 6–6
NAM$V_LOWVER bit • *RMS*, 6–6
NAM$V_NOCONCEAL option • *RMS*, 6–8, RMS–16, RMS–68
NAM$V_NODE bit • *RMS*, 6–6
NAM$V_PPF bit • *RMS*, 6–6
NAM$V_PWD field • *RMS*, RMS–26, RMS–63, RMS–68
NAM$V_PWD option • *RMS*, 6–8, RMS–16
NAM$V_QUOTED bit • *RMS*, 6–6
NAM$V_ROOT_DIR bit • *RMS*, 6–7
NAM$V_SEARCH_LIST bit • *RMS*, 6–7
NAM$V_SRCHXABS option • *RMS*, 6–8
NAM$V_SYNCHK option • *RMS*, 6–8, RMS–68
  use with Parse service • *RMS*, 5–7
  using for Parse service without I/O • *RMS*, RMS–67
NAM$V_WILDCARD bit • *RMS*, 6–7
NAM$V_WILD_GRP bit • *RMS*, 6–7
NAM$V_WILD_MBR bit • *RMS*, 6–7
NAM$V_WILD_NAME bit • *RMS*, 6–7
NAM$V_WILD_SFD1 bit • *RMS*, 6–7
NAM$V_WILD_TYPE bit • *RMS*, 6–7
NAM$V_WILD_UFD bit • *RMS*, 6–7
NAM$V_WILD_VER bit • *RMS*, 6–7
NAM$W_DID field • *File Applications*, 6–5; *RMS*, 6–4
NAM$W_FID field • *File Applications*, 6–5; *RMS*, 6–6
NAM (name block) • *Programming Resources*, 1–36; *System Dump Analyzer*, SDA–77
  address field • *File Applications*, 5–9
  and resulting file specification • *File Applications*, 5–8
  and Search service • *File Applications*, 5–8
  presence of a search list • *File Applications*, 5–9
  presence of a wildcard character • *File Applications*, 5–9
  summary of fields • *RMS*, 6–1
  support by FDL • *File Applications*, 5–10
  support by languages • *File Applications*, 5–10
  using • *File Applications*, 5–12 to 5–14
  using from higher-level language • *RMS*, 6–2
  using from VAX MACRO • *RMS*, 6–2
NAM (name block) option
  See FAB$V_NAM option
$NAMDEF • *File Applications*, 5–10

Name
  widget
    case sensitivity of • *VAXTPU,* 7–74
%NAME • *Debugger,* D–4
NAME attribute • *File Def Language,* FDL–19, FDL–22, FDL–29
Name block
  See NAM
Name block address field
  See FAB$L_NAM field
Name block options field
  See NAM$B_NOP field
NAME keyword
  with FILE_PARSE • *VAXTPU,* 7–141
  with FILE_SEARCH • *VAXTPU,* 7–144
/NAMES qualifier • *Librarian,* LIB–33
"Name" string constant parameter to GET_INFO • *VAXTPU,* 7–164, 7–173, 7–182
Naming
  application-wide • *RTL Parallel Processing,* 2–4
Naming conventions • *Modular Procedures,* 3–1, A–6
  FOLR routines • *RTL Math,* 2–7
  for facilities • *Modular Procedures,* 3–2
  for files • *Modular Procedures,* 3–4
  for modules • *Modular Procedures,* 3–4
  for procedures • *Modular Procedures,* 3–3
  for PSECTs • *Modular Procedures,* 3–5
  for VAXTPU procedures • *VAXTPU,* 3–16
  macros • *RMS,* 3–2
  services • *RMS,* 3–3
  vector routines • *RTL Math,* 2–9
Naming help modules • *Librarian,* LIB–4
Naming PPL$ components • *RTL Parallel Processing,* 5–5
$NAM macro • *RMS,* B–6
  argument categories • *RMS,* B–6
$NAM_STORE macro • *RMS,* B–7
  argument categories • *RMS,* B–7
  comparing with $NAM macro • *RMS,* B–7
  NAM$T_DVI argument • *RMS,* B–7
  NAM$W_DID argument • *RMS,* B–7
  NAM$W_FID argument • *RMS,* B–7
  requirements • *RMS,* B–7
.NARG directive • *MACRO,* 6–63
NARGS keyword • *System Services Intro,* 2–8
National Character Set (NCS) Routines
  See NCS routines
National Character Set Utility (NCS) • *Programming Resources,* 1–22; *National Char Set,* NCS–3
  DCL interface
    default function • *National Char Set,* NCS–3
National Character Set Utility (NCS)
  DCL interface (Cont.)
    library functions • *National Char Set,* NCS–3
  directing output from • *National Char Set,* NCS–21
  exiting • *National Char Set,* NCS–21
  functions • *National Char Set,* NCS–3
  implementation • *National Char Set,* NCS–3
Native language
  on VMS • *File Def Language,* FDL–41
NBI
  See Memory interconnect to VAXBI adapter
NBP (next block pointer)
  default for block transfer • *RMS,* 7–2
  for block I/O • *RMS,* 4–25
  functions listed • *RMS,* 4–25
.NCHR directive • *MACRO,* 6–64
NCR 5380 controller • *Device Support (A),* 1–18
NCS
  See National Character Set Utility
NCS$COMPARE routine • *Utility Routines,* NCS–7
NCS$CONVERT routine • *Utility Routines,* NCS–9
NCS$END_CF routine • *Utility Routines,* NCS–11
NCS$END_CS routine • *Utility Routines,* NCS–12
NCS$GET_CF routine • *Utility Routines,* NCS–13
NCS$GET_CS routine • *Utility Routines,* NCS–15
NCS$RESTORE_CF routine • *Utility Routines,* NCS–17
NCS$RESTORE_CS routine • *Utility Routines,* NCS–19
NCS$SAVE_Cf routine • *Utility Routines,* NCS–21
NCS$SAVE_CS routine • *Utility Routines,* NCS–23
NCS collating sequence end routine
  See NCS$END_CS routine
NCS command
  specifying input files for • *National Char Set,* NCS–21
NCS compare strings routine
  See NCS$COMPARE routine
NCS conversion function end routine
  See NCS$END_CF routine
NCS convert string routine
  See NCS$CONVERT routine
NCS get collating sequence routine
  See NCS$GET_CS routine
NCS get conversion function routine
  See NCS$GET_CF routine
NCS keyword
  for /FORMAT qualifier • *National Char Set,* NCS–29
NCS library
  creating • *National Char Set,* NCS–25

NCS library
  creating (Cont.)
    See also /CREATE qualifier
  deleting definition modules from • *National Char Set*, NCS–27
  extracting definition modules from • *National Char Set*, NCS–28
  generating MACRO-32 output from • *National Char Set*, NCS–36
  generating NCS definition files from • *National Char Set*, NCS–39
  inserting definition modules • *National Char Set*, NCS–32
  obtaining listing of • *National Char Set*, NCS–34
  replacing definition modules • *National Char Set*, NCS–40
  specifying an alternate • *National Char Set*, NCS–33
  specifying history records • *National Char Set*, NCS–24, NCS–25
  specifying MACRO-32 output format • *National Char Set*, NCS–29
  specifying maximum length of definition module names • *National Char Set*, NCS–24, NCS–25
  specifying maximum number of modules • *National Char Set*, NCS–24, NCS–25
  specifying size • *National Char Set*, NCS–24, NCS–25
  verifying operations • *National Char Set*, NCS–35
  with data-expanded format • *National Char Set*, NCS–26
  with data-reduced format • *National Char Set*, NCS–26
NCS restore collating sequence routine
  See NCS$RESTORE_CS routine
NCS restore conversion function routine
  See NCS$RESTORE_CF routine
NCS routines • *Utility Routines*, NCS–1
  example of use in FORTRAN program • *Utility Routines*, NCS–2
  example of use in MACRO-32 program • *Utility Routines*, NCS–4
  list of • *Utility Routines*, NCS–1
  typical application of • *Utility Routines*, NCS–2
NCS save collating sequence routine
  See NCS$SAVE_CS routine
NCS save conversion function routine
  See NCS$SAVE_CF routine
NEF option • *File Def Language*, FDL–21
NEGATABLE clause
  for DEFINE TYPE statement • *Command Def*, CDU–28
NEGATABLE clause (Cont.)
  for QUALIFIER clause • *Command Def*, CDU–25, CDU–34
Negative compression • *File Def Language*, FDL–4
Negative condition code (N) • *MACRO*, 8–15
Negative operator (–) • *System Dump Analyzer*, SDA–12
NETDEF.STB • *System Dump Analyzer*, SDA–60
Network
  completing connection • *Programming Resources*, 3–27
  connection request • *Programming Resources*, 3–26
  exchanging messages • *Programming Resources*, 3–28
  terminating connection • *Programming Resources*, 3–30
NETWORK attribute • *File Def Language*, FDL–32
Network device • *Device Support (B)*, 1–74
Network work area
  See NWA
NETWORK_BLOCK_COUNT qualifier
  for specifying maximum record size • *RMS*, 5–22
NETWORK_DATA_CHECKING attribute • *File Def Language*, FDL–32
/NEW_VERSION qualifier • *Patch*, PAT–30
Next block pointer
  See NBP
NEXT command • *File Applications*, 10–12, 10–16; *Analyze/RMS_File*, ARMS–29
%NEXTDISP • *Debugger*, C–7
%NEXTINST • *Debugger*, C–7
Next key
  See RAB$V_NXT option
Next key option • *File Applications*, 8–9, 8–10
%NEXTLOC • *Debugger*, 4–8, D–5
Next location
  See Logical successor
Next or equal key option
  See RAB$V_EQNXT option
%NEXTOUTPUT • *Debugger*, C–7
/NEXT qualifier • *Debugger*, 6–6, CD–121
Next-record position • *File Applications*, 8–16
  use with sequential access • *File Applications*, 8–16
%NEXTSCROLL • *Debugger*, C–7
%NEXTSOURCE • *Debugger*, C–7
"Next" string constant parameter to GET_INFO • *VAXTPU*, 7–166, 7–168, 7–169, 7–180, 7–181, 7–183, 7–184, 7–191, 7–218, 7–223
Next Volume service • *File Applications*, 8–5; *RMS*, RMS–55

Next Volume service (Cont.)
  condition values • *RMS,* RMS–57
  control block input and output fields • *RMS,* RMS–56
  flush logic • *RMS,* RMS–56
  input logic sequence • *RMS,* RMS–56
  output logic sequence • *RMS,* RMS–56
  requirements for using • *RMS,* RMS–56
"Next_marker" string constant parameter to GET_INFO • *VAXTPU,* 7–173
%NEXT_PROCESS • *Debugger,* 10–12
"Next_range" string constant parameter to GET_INFO • *VAXTPU,* 7–173
%NEXT_SCOPE_ENTRY • *Debugger,* D–11
%NEXT_TASK • *Debugger,* D–10
Nexus • *Device Support (A),* 12–5, 12–8, 12–9, 12–10, 12–11
Nexus ID • *Device Support (B),* 1–6
NFS option • *File Def Language,* FDL–22
NIL option • *File Def Language,* FDL–37
.NLIST directive • *MACRO,* 6–65
  See also .NOSHOW directive
NLK option • *File Def Language,* FDL–12
nnDRIVER symbol • *System Dump Analyzer,* SDA–13
/NOAPPEND qualifier • *Convert,* CONV–7
NOCONCATENATE clause
  for VALUE clause • *Command Def,* CDU–24, CDU–33
/NOCREATE qualifier • *Convert,* CONV–8
.NOCROSS directive • *MACRO,* 6–16, 6–66
Node • *Device Support (A),* 12–5, 12–8, 12–9, 12–10, 12–11
  See VAXBI node
  lock-mastering • *File Applications,* 3–29
  lock-requesting • *File Applications,* 3–29
Node ID • *Device Support (A),* 16–9; *Device Support (B),* 1–6
NODE keyword
  with FILE_PARSE • *VAXTPU,* 7–140
  with FILE_SEARCH • *VAXTPU,* 7–143
Node name address descriptor
  See NAM$L_NODE descriptor
Node name address field
  See NAM$L_NODE field
Node name length field
  See NAM$B_NODE field
Node name size descriptor
  See NAM$B_NODE descriptor
Node private space • *Device Support (A),* 16–5
Node space • *Device Support (A),* 16–5
Node space (Cont.)
  accessing BIIC registers within • *Device Support (A),* 16–5
  address • *Device Support (A),* 16–9
  mapped by VMS • *Device Support (A),* 16–8
NODISALLOW clause
  for DEFINE SYNTAX statement • *Command Def,* CDU–22
  for DEFINE VERB statement • *Command Def,* CDU–31
/NODISPLAY qualifier
  effect on LAST_KEY • *VAXTPU,* 7–242
  to disable screen manager • *VAXTPU,* 6–1
  with EVE$BUILD • *VAXTPU,* G–10
/NOEXCEPTIONS_FILE qualifier • *Convert,* CONV–9
/NOEXIT qualifier • *Convert,* CONV–10
/NOFAST_LOAD option
  compared with /FAST_LOAD option • *Convert,* CONV–11
/NOFAST_LOAD qualifier • *Convert,* CONV–11
/NOFILL_BUCKETS qualifier • *Convert,* CONV–14
/NOFIXED_CONTROL qualifier • *Convert,* CONV–15
/NOINTERACTIVE qualifier • *File Applications,* 10–29; *File Def Language,* FDL–42, FDL–52
/NOJOURNAL command qualifier • *VAXTPU,* 1–12
NOLOCK attribute • *File Def Language,* FDL–12
No lock option
  See RAB$V_NLK option
NOLOCK secondary attribute • *File Applications,* 7–11
NO logical value • *File Def Language,* FDL–2
/NOLOGICAL_NAMES qualifier • *System Dump Analyzer,* SDA–162
/NOLOG qualifier
  CREATE/FDL • *File Def Language,* FDL–45
"Nomodify" string constant parameter to GET_INFO • *VAXTPU,* 7–177
Noncontiguous array descriptor • *Routines Intro,* 2–31
Non-Digital-supplied SCSI class driver
  See Third-party SCSI class driver
Non-Digital terminal
  support for • *RTL Screen Management,* 5–1
Non-direct-vector interrupt • *Device Support (A),* 13–9, 14–3, 14–28, 14–29, 14–31; *Device Support (B),* 1–7, 1–25
NONE carriage control • *File Def Language,* FDL–34
NONE keyword
  with MARK • *VAXTPU,* 7–261
  with SELECT • *VAXTPU,* 7–337
  with SET (MESSAGE_ACTION_TYPE) • *VAXTPU,* 7–426
  with SET (PROMPT_AREA) • *VAXTPU,* 7–446
  with SET (STATUS_LINE) • *VAXTPU,* 7–476

NONE keyword (Cont.)
with SET (VIDEO) • *VAXTPU*, 7–492
Nonexecutable message file
creating • *Message*, MSG–4
Nonexistent record option
See RAB$V_NXR option
NONEXISTENT_RECORD attribute • *File Def Language*, FDL–12
NONEXISTENT_RECORD secondary attribute • *File Applications*, 7–15, 8–9
Non-file-structured option
See FAB$V_NFS option
NONNEGATABLE clause
for DEFINE TYPE statement • *Command Def*, CDU–28
for QUALIFIER clause • *Command Def*, CDU–25, CDU–34
Nonpaged dynamic storage pool
displaying contents • *System Dump Analyzer*, SDA–118
Nonpaged pool
allocating • *Device Support (B)*, 3–12 to 3–13, 3–14, 3–15, 3–22 to 3–23
allocating in initialization routine • *Device Support (A)*, 11–2
deallocating • *Device Support (B)*, 3–3, 3–19
lookaside list • *Device Support (A)*, E–14; *Device Support (B)*, 3–13, 3–14
synchronizing access to • *Device Support (A)*, 3–14
variable region • *Device Support (A)*, E–14; *Device Support (B)*, 3–15
/NONPAGED qualifier • *System Dump Analyzer*, SDA–118
Nonstandard file processing
run-time options • *File Applications*, 9–14
Nonstatic variable • *Debugger*, 3–19, 4–1
with DECwindows • *Debugger*, 1–25
/NOOPTIMIZE qualifier • *Debugger*, 2–5, 5–2, 9–1
with DECwindows • *Debugger*, 1–3
/NOOUTPUT qualifier • *Analyze/RMS_File*, ARMS–16
NOP (No Operation) instruction • *Debugger*, 4–23; *MACRO*, 9–78
/NOPAD qualifier • *Convert*, CONV–18
NOPARAMETERS clause
for DEFINE SYNTAX statement • *Command Def*, CDU–23
for DEFINE VERB statement • *Command Def*, CDU–32
NOP field
specifying multiple values • *RMS*, B–6
NOQUALIFIERS clause
for DEFINE SYNTAX statement • *Command Def*, CDU–24
for DEFINE VERB statement • *Command Def*, CDU–33
/NOREAD_CHECK qualifier • *Convert*, CONV–20
Norm
Euclidean
of a vector • *RTL Math*, MTH–170
Normal directory syntax • *File Applications*, 6–12 to 6–14
/NOSCRIPT qualifier • *File Def Language*, FDL–42, FDL–57
/NOSHARE qualifier • *Convert*, CONV–21
No sharing option
See FAB$V_NIL option
.NOSHOW directive • *MACRO*, 6–67, 6–89
/NOSKIP qualifier • *System Dump Analyzer*, SDA–52
/NOSORT qualifier • *Convert*, CONV–22
for avoiding unnecessary sort • *Convert*, CONV–11
/NOSTATISTICS qualifier
with CONVERT • *Convert*, CONV–24
with CONVERT/RECLAIM • *Convert*, CONV–5, CONV–24
/NOSUPPRESS qualifier • *System Dump Analyzer*, SDA–52
/NOSYMBOLS qualifier • *System Dump Analyzer*, SDA–162
NOTANY built-in procedure • *VAXTPU*, 7–284 to 7–285
Not end-of-file option
See FAB$V_NEF option
Notification
of abnormal exit • *RTL Parallel Processing*, 4–9
of normal exit • *RTL Parallel Processing*, 4–9
/NOTIFY qualifier • *System Dump Analyzer*, SDA–162
NOT operator • *VAXTPU*, 3–7
NOT operator (#) • *System Dump Analyzer*, SDA–12
/NOTRUNCATE qualifier • *Convert*, CONV–26
/NOWAIT qualifier • *System Dump Analyzer*, SDA–162
/NOWRITE_CHECK qualifier • *Convert*, CONV–28
NO_EXACT keyword
with LEARN_BEGIN • *VAXTPU*, 7–244
with SEARCH • *VAXTPU*, 7–328
with SEARCH_QUIETLY • *VAXTPU*, 7–333
NO_TRANSLATE keyword • *VAXTPU*, 7–483
"No_video" string constant parameter to GET_INFO • *VAXTPU*, 7–223
"No_video_status" string constant parameter to GET_INFO • *VAXTPU*, 7–223

"No_write" GET_INFO request_string • *VAXTPU*, 7–174

NO_WRITE keyword • *VAXTPU*, 7–434

NPR (Nonprocessor request)

  See DMA transfer

.NTYPE directive • *MACRO*, 6–68

Null

  key value • *File Def Language*, FDL–29

  string • *File Def Language*, FDL–2

Null arguments • *System Services Intro*, 1–5

Null character field

  See XAB$B_NUL field

Null device • *System Services Intro*, 7–28

Null key

  for improving performance • *File Applications*, 3–19

NULL pad character • *Convert*, CONV–18

Null parameters • *VAXTPU*, 3–18

null_arg data type • *Routines Intro*, A–10t

NULL_KEY attribute • *File Def Language*, FDL–29

NULL_VALUE attribute • *File Def Language*, FDL–29

Number

  See also Integer, Floating-point number, and Packed decimal string

  in source statement • *MACRO*, 3–2

Number of allocation areas field

  See XAB$B_NOA field

Number of arguments directive (.NARG) • *MACRO*, 6–63

Number of characters directive (.NCHR) • *MACRO*, 6–64

Number of files processed • *Convert*, CONV–24

Number of key segments field

  See XAB$B_NSG field

Number of keys field

  See XAB$B_NOK field

Number of modules

  in NCS library, specifying • *National Char Set*, NCS–24, NCS–25

Number sign (#)

  requirement for in control store macro • *RMS*, 3–8

Number value • *File Def Language*, FDL–2

/NUMBER_KEYS qualifier • *File Def Language*, FDL–42, FDL–53

Numeric constant

  specifying radix of • *VAXTPU*, 3–37

Numeric control operator • *MACRO*, 3–14

Numeric data

  entering • *Patch*, PAT–22

Numeric expression • *Delta/XDelta*, DELTA–9, DELTA–42

Numeric string

  leading separate • *MACRO*, 8–11

  trailing • *MACRO*, 8–8

Numeric time • *System Services Intro*, 9–7

NWA (network work area) • *System Dump Analyzer*, SDA–77

NXR option • *File Def Language*, FDL–12

O

Object

  definition of • *RTL Parallel Processing*, 1–2

  protection • *Device Support (B)*, 1–45

  retrieving information about • *RTL Parallel Processing*, 4–1

Object code • *Debugger*, 9–1

Object file

  input to linker • *Linker*, 1–4, 2–2

  processing of • *Linker*, 6–9, 6–12

  used as linker input • *Linker*, 1–4

Object language • *Linker*, 7–1 to 7–37

  See also Linker Utility

Object library • *Programming Resources*, 1–18, 5–1, 5–12; *Librarian*, LIB–1

  adding a module • *Programming Resources*, 5–2

  character case in • *Librarian*, LIB–2

  creating • *Programming Resources*, 5–2

  deleting a module • *Programming Resources*, 5–2

  extracting a module • *Programming Resources*, 5–2

  including message object module • *Programming Resources*, 9–9

  listing modules • *Programming Resources*, 5–2

  replacing a module • *Programming Resources*, 5–2

Object module • *Debugger*, 5–3, 6–1

  See also Message object module

  contents of • *Linker*, 2–2

  for command table • *Command Def*, CDU–4, CDU–16, CDU–41

  how to create • *Command Def*, CDU–46

  identifying • *MACRO*, 6–39

  input to linker • *Linker*, 6–3

  naming • *MACRO*, 6–95

  record contents of • *Linker*, 6–3

  statements for • *Command Def*, CDU–14

  title • *MACRO*, 6–95

Object module library

  contents of • *Linker*, 2–3

  creating • *Modular Procedures*, 5–2

Object module library (Cont.)
input to linker • *Linker*, 2–3
processing of • *Linker*, 6–13
updating • *Modular Procedures*, 6–5
/OBJECT qualifier • *Command Def*, CDU–41; *Librarian*, LIB–34; *Message*, MSG–12
Occlusion • *RTL Screen Management*, 2–5
O command • *Delta/XDelta*, DELTA–35
%OCT • *Debugger*, 4–12, D–5
.OCTA directive • *MACRO*, 6–70
OCTAL mode • *Patch*, PAT–17
/OCTAL qualifier • *Debugger*, 4–12, CD–81, CD–83, CD–87
/OCTAL qualifier
with DELETE command • *Patch*, PAT–52
with DEPOSIT command • *Patch*, PAT–55
with EVALUATE command • *Patch*, PAT–59
with EXAMINE command • *Patch*, PAT–62
with INSERT command • *Patch*, PAT–68
with REPLACE command • *Patch*, PAT–71
with SET MODE command • *Patch*, PAT–76
with VERIFY command • *Patch*, PAT–90
Octal text
converting to binary • *RTL Library*, LIB–76
Octaword data type • *MACRO*, 8–3
/OCTAWORD qualifier • *Debugger*, CD–63, CD–87
Octaword storage directive (.OCTA) • *MACRO*, 6–70
octaword_signed data type • *Routines Intro*, A–10t
octaword_unsigned data type • *Routines Intro*, A–10t
.ODD directive • *MACRO*, 6–71
OFF keyword
with CREATE_WINDOW • *VAXTPU*, 7–77
with HELP_TEXT • *VAXTPU*, 7–228
with QUIT • *VAXTPU*, 7–291
with SET (AUTO_REPEAT) • *VAXTPU*, 7–353
with SET (BELL) • *VAXTPU*, 7–355
with SET (COLUMN_MOVE_VERTICAL) • *VAXTPU*, 7–359
with SET (CROSS_WINDOW_BOUNDS) • *VAXTPU*, 7–361
with SET (DEBUG) • *VAXTPU*, 7–363, 7–364
with SET (INFORMATIONAL) • *VAXTPU*, 7–397
with SET (LINE_NUMBER) • *VAXTPU*, 7–416
with SET (MODIFIABLE) • *VAXTPU*, 7–429
with SET (MOUSE) • *VAXTPU*, 7–432
with SET (NO_WRITE) • *VAXTPU*, 7–434
with SET (PAD) • *VAXTPU*, 7–437
with SET (PAD_OVERSTRUCK_TABS) • *VAXTPU*, 7–439
with SET (SCREEN_UPDATE) • *VAXTPU*, 7–460
with SET (SCROLLING) • *VAXTPU*, 7–467
with SET (SELF_INSERT) • *VAXTPU*, 7–470
OFF keyword (Cont.)
with SET (SUCCESS) • *VAXTPU*, 7–479
with SET (TIMER) • *VAXTPU*, 7–486
with SET (TRACEBACK) • *VAXTPU*, 7–488
with SPAWN • *VAXTPU*, 7–515
“Offset” string constant parameter to GET_INFO • *VAXTPU*, 7–174, 7–186
“Offset_column” string constant parameter to GET_INFO • *VAXTPU*, 7–174, 7–186
OFP option • *File Def Language*, FDL–22
One's complement
of expression • *MACRO*, 3–14
ON keyword
with CREATE_WINDOW • *VAXTPU*, 7–77
with HELP_TEXT • *VAXTPU*, 7–228
with QUIT • *VAXTPU*, 7–291
with SET (AUTO_REPEAT) • *VAXTPU*, 7–353
with SET (BELL) • *VAXTPU*, 7–355
with SET (COLUMN_MOVE_VERTICAL) • *VAXTPU*, 7–359
with SET (CROSS_WINDOW_BOUNDS) • *VAXTPU*, 7–361
with SET (DEBUG) • *VAXTPU*, 7–363
with SET (INFORMATIONAL) • *VAXTPU*, 7–397
with SET (LINE_NUMBER) • *VAXTPU*, 7–416
with SET (MODIFIABLE) • *VAXTPU*, 7–429
with SET (MOUSE) • *VAXTPU*, 7–432
with SET (NO_WRITE) • *VAXTPU*, 7–434
with SET (PAD) • *VAXTPU*, 7–437
with SET (PAD_OVERSTRUCK_TABS) • *VAXTPU*, 7–439
with SET (SCREEN_UPDATE) • *VAXTPU*, 7–460
with SET (SCROLLING) • *VAXTPU*, 7–467
with SET (SELF_INSERT) • *VAXTPU*, 7–470
with SET (SUCCESS) • *VAXTPU*, 7–479
with SET (TIMER) • *VAXTPU*, 7–486
with SET (TRACEBACK) • *VAXTPU*, 7–488
with SPAWN • *VAXTPU*, 7–515
Online bit
See UCB$V_ONLINE
Online condition
on MASSBUS • *Device Support (A)*, 15–10
/ONLY qualifier • *Librarian*, LIB–35; *National Char Set*, NCS–38
ON_ERROR statement • *VAXTPU*, 3–25 to 3–31
location • *VAXTPU*, 3–25
ON_ERROR Statement • *VAXTPU*, 3–21
Opcode
creating • *MACRO*, 6–72
defining • *MACRO*, 6–83
format • *MACRO*, 8–16
illegal vector • *MACRO*, 10–17

Opcode (Cont.)
- redefining • *MACRO,* 6–58, 6–72
- summary • *MACRO,* D–1
  - alphabetic order • *MACRO,* D–1
  - numeric order • *MACRO,* D–12
- VAX MACRO instructions with same • *Patch,* PAT–21
- with the same name as a macro • *MACRO,* 6–58

Opcode definition directive (.OPDEF) • *MACRO,* 6–72

OPCOM process
- sending a message to • *Device Support (A),* 10–7; *Device Support (B),* 3–53, 3–61

.OPDEF directive • *MACRO,* 6–72

Open-by-name-block option • *File Applications,* 5–9, 6–5
- and performance • *File Applications,* 6–7

Open Location and Display Contents command • *Delta/XDelta,* DELTA–17

Open Location and Display Contents in Instruction Mode command • *Delta/XDelta,* DELTA–20

Open Location and Display Indirect Location command • *Delta/XDelta,* DELTA–24

Open Location and Display Previous Location command • *Delta/XDelta,* DELTA–23

$OPEN macro
- expansion of • *RMS,* 3–10
- for invoking the Open service • *RMS,* 4–1
- using in example • *RMS,* 3–10, 3–11

Open service • *File Applications,* 5–9; *RMS,* RMS–58
- condition values • *RMS,* RMS–64
- contrasted with Parse and Search services • *RMS,* 4–10
- control block input fields • *RMS,* RMS–59
- control block output fields • *RMS,* RMS–61
- for process-permanent files • *File Applications,* 6–21
- function • *RMS,* 4–1
- invoking • *RMS,* 4–4
- NAM input fields • *RMS,* RMS–63
- NAM output fields • *RMS,* RMS–63
- program example • *RMS,* 4–2
- requirements for using • *RMS,* RMS–59

Operand • *MACRO,* 2–3
- determining addressing mode of • *MACRO,* 6–68
- instruction • *Debugger,* 4–20, CD–87, CD–153
- primary • *MACRO,* 8–26
- reserved • *MACRO,* 9–102, 9–103, 9–145
- vector instruction • *Debugger,* 11–6, 11–9

Operand generation directive
- (.REF16) • *MACRO,* 6–83
- (.REF2) • *MACRO,* 6–83

Operand generation directive (Cont.)
- (.REF4) • *MACRO,* 6–83
- (.REF8) • *MACRO,* 6–83

Operand specifier • *MACRO,* 8–17
- access type notation • *MACRO,* 9–2
- access types • *MACRO,* 8–17
- base • *MACRO,* 8–26
- data type notation • *MACRO,* 9–2
- data types • *MACRO,* 8–17
- notation • *MACRO,* 9–2
- restrictions on usage for vector instructions • *MACRO,* 10–16

Operand specifier addressing mode formats • *MACRO,* 8–18
- autodecrement mode • *MACRO,* 8–21
- autoincrement deferred mode • *MACRO,* 8–20
- autoincrement mode • *MACRO,* 8–19
- branch mode • *MACRO,* 8–29
- displacement deferred mode • *MACRO,* 8–22
- displacement mode • *MACRO,* 8–21
- index mode • *MACRO,* 8–26
- literal mode • *MACRO,* 8–23
- register deferred mode • *MACRO,* 8–19
- register mode • *MACRO,* 8–19

/OPERANDS qualifier • *Debugger,* 4–20, 11–9, CD–87, CD–153

Operand type directive (.NTYPE) • *MACRO,* 6–68

Operational controls • *RTL Screen Management,* 2–16

Operation involving condition handler • *Routines Intro,* 2–46

Operator • *Patch,* PAT–23; *SUMSLP,* SUM–3; *System Dump Analyzer,* SDA–12; *MACRO,* 2–3
- address expression • *Debugger,* D–6
- AND • *MACRO,* 3–16
- arithmetic • *Delta/XDelta,* DELTA–10
- arithmetic shift • *MACRO,* 3–16
- ASCII • *MACRO,* 3–12
- binary • *MACRO,* 3–15, C–8
- complement • *MACRO,* 3–14
- exclusive OR • *MACRO,* 3–16
- floating-point • *MACRO,* 3–14
- for addressing locations • *Patch,* PAT–24
- for arithmetic expressions • *Patch,* PAT–23
- for DISALLOW clause • *Command Def,* CDU–13
- inclusive OR • *MACRO,* 3–16
- language expression • *Debugger,* E–1
- macro • *MACRO,* 4–8
- macro string • *MACRO,* C–8
- numeric control • *MACRO,* 3–14
- pattern • *MACRO,* 9–172

Operator (Cont.)
- precedence of • *System Dump Analyzer*, SDA–12, SDA–13
- radix control • *MACRO*, 3–11
- register • *MACRO*, 3–13
- sending message • *System Services*, SYS–495
- summary • *MACRO*, C–7
- textual • *MACRO*, 3–12
- unary • *MACRO*, 3–10, C–7

Operator device • *Device Support (B)*, 1–74

Operators • *VAXTPU*, 3–6 to 3–8
- partial pattern assignment (@) • *VAXTPU*, 2–17
- pattern alternation (|) • *VAXTPU*, 2–16
- pattern concatenation (+) • *VAXTPU*, 2–15
- pattern linking (&) • *VAXTPU*, 2–15
- precedence • *VAXTPU*, 3–7
- relational • *VAXTPU*, 2–18

Optimization
- Edit/FDL Utility • *File Applications*, A–1
- effect on debugging • *Debugger*, 2–5, 5–2, 7–8, 9–1
  - with DECwindows • *Debugger*, 1–3, 1–9, 1–11
- of indexed file • *File Applications*, 10–29

/OPTIMIZE qualifier • *Debugger*, 2–5, 5–2, 9–1
- with DECwindows • *Debugger*, 1–3

Optimize script • *File Def Language*, FDL–39, FDL–47

Option
- BASE= • *Linker*, 1–7, 3–5
- CLUSTER= • *Linker*, 1–7, 3–6
- COLLECT= • *Linker*, 1–8, 3–6
- creating with LBR$OPEN • *Programming Resources*, 8–36
- default values • *Linker*, 3–2
- DZROMIN= • *Linker*, 1–8, 3–7
- GSMATCH= • *Linker*, 1–8, 3–7
- IDENTIFICATION= • *Linker*, 1–8, 3–9
- IOSEGMENT= • *Linker*, 1–6, 1–8, 2–11, 3–9
- ISDMAX= • *Linker*, 1–8, 3–10
- NAME= • *Linker*, 1–8, 3–10
- PROTECT= • *Linker*, 1–8, 3–10
- PSECTATTR= • *Linker*, 1–9, 3–11
- specifying by symbolic bit offset • *RMS*, 2–3
- STACK= • *Linker*, 1–6, 1–9, 2–11, 3–11
- SYMBOL= • *Linker*, 1–9, 3–11
- UNIVERSAL= • *Linker*, 1–9, 3–12

Optional argument
- to service • *RMS*, 3–11

Options file • *Programming Resources*, 5–8
- See also Linker Utility
- content of • *Linker*, 2–5, 3–1
- creating • *Programming Resources*, 5–6; *Linker*, 1–7

Options file (Cont.)
- creation of • *Linker*, 3–4
- how used with linker • *Linker*, 1–6
- identification of • *Linker*, LINK–26
- in command procedure • *Linker*, 3–4
- input to linker • *Linker*, 1–5, 2–4
- processing of • *Linker*, 6–9
- rules for • *Linker*, 1–7, 3–4
- specification of clusters in • *Linker*, 6–10
- use for • *Linker*, 2–5, 3–1

/OPTIONS qualifier • *Debugger*, 5–12; *Linker*, 1–5, 2–4, LINK–26

ORB (object rights block) • *Device Support (B)*, 1–44 to 1–46
- address • *Device Support (B)*, 1–73
- cloned • *Device Support (A)*, 11–13; *Device Support (B)*, 4–7

Organization
- See File organization

ORGANIZATION attribute • *File Def Language*, FDL–22

ORGANIZATION secondary attribute • *File Applications*, 4–28

Organizing
- files and modules • *Modular Procedures*, 2–1
- procedures • *Modular Procedures*, 2–1

Organizing a file • *Convert*, CONV–1
- See also File organization

"Original_bottom" string constant parameter to GET_INFO • *VAXTPU*, 7–223

"Original_length" string constant parameter to GET_INFO • *VAXTPU*, 7–223

"Original_top" string constant parameter to GET_INFO • *VAXTPU*, 7–223

"Original_width" string constant parameter to GET_INFO • *VAXTPU*, 7–200

OR operator • *VAXTPU*, 3–7

OR operator (|) • *System Dump Analyzer*, SDA–12

OTS$CNVOUT • *RTL General Purpose*, OTS–3

OTS$CNVOUT_G • *RTL General Purpose*, OTS–3

OTS$CNVOUT_H • *RTL General Purpose*, OTS–3

OTS$CVT_L_TB • *RTL General Purpose*, OTS–5

OTS$CVT_L_TI • *RTL General Purpose*, OTS–7

OTS$CVT_L_TL • *RTL General Purpose*, OTS–9

OTS$CVT_L_TO • *RTL General Purpose*, OTS–11

OTS$CVT_L_TU • *RTL General Purpose*, OTS–13

OTS$CVT_L_TZ • *RTL General Purpose*, OTS–15

OTS$CVT_TB_L • *RTL General Purpose*, OTS–17

OTS$CVT_TI_L • *RTL General Purpose*, OTS–20

OTS$CVT_TL_L • *RTL General Purpose*, OTS–22

OTS$CVT_TO_L • *RTL General Purpose*, OTS–24

OTS$CVT_TU_L • *RTL General Purpose*, OTS–27

- OTS$CVT_TZ_L • *RTL General Purpose*, OTS–36
- OTS$CVT_T_z • *RTL General Purpose*, OTS–29, OTS–33
- OTS$DIVC • *RTL General Purpose*, OTS–39
- OTS$DIVCD_R3 • *RTL General Purpose*, OTS–39
- OTS$DIVCG_R3 • *RTL General Purpose*, OTS–39
- OTS$DIV_PK_LONG • *RTL General Purpose*, OTS–42
- OTS$DIV_PK_SHORT • *RTL General Purpose*, OTS–46
- OTS$MOVE3 • *RTL General Purpose*, OTS–49
- OTS$MOVE5 • *RTL General Purpose*, OTS–51
- OTS$MULCD_R3 • *RTL General Purpose*, OTS–53
- OTS$MULCG_R3 • *RTL General Purpose*, OTS–53
- OTS$POWCxCx • *RTL General Purpose*, OTS–55
- OTS$POWCxJ • *RTL General Purpose*, OTS–58
- OTS$POWDD • *RTL General Purpose*, OTS–61
- OTS$POWDJ • *RTL General Purpose*, OTS–65
- OTS$POWDLU • *RTL General Purpose*, OTS–79
- OTS$POWDR • *RTL General Purpose*, OTS–63
- OTS$POWGG • *RTL General Purpose*, OTS–67
- OTS$POWGJ • *RTL General Purpose*, OTS–70
- OTS$POWGLU • *RTL General Purpose*, OTS–79
- OTS$POWHH_R3 • *RTL General Purpose*, OTS–72
- OTS$POWHJ_R3 • *RTL General Purpose*, OTS–74
- OTS$POWHLU_R3 • *RTL General Purpose*, OTS–79
- OTS$POWII • *RTL General Purpose*, OTS–76
- OTS$POWJJ • *RTL General Purpose*, OTS–77
- OTS$POWLULU • *RTL General Purpose*, OTS–78
- OTS$POWRD • *RTL General Purpose*, OTS–81
- OTS$POWRJ • *RTL General Purpose*, OTS–84
- OTS$POWRLU • *RTL General Purpose*, OTS–79
- OTS$POWRR • *RTL General Purpose*, OTS–86
- OTS$SCOPY_DXDX • *RTL General Purpose*, OTS–89; *RTL String Manipulation*, 2–7
- OTS$SCOPY_R_DX • *RTL General Purpose*, OTS–91
- OTS$SFREE1_DD • *RTL General Purpose*, OTS–94
- OTS$SFREEN_DD • *RTL General Purpose*, OTS–95
- OTS$SGET1_DD • *RTL General Purpose*, OTS–96
- Out-of-band AST • *I/O User's I*, 8–13, 8–46
- Output
  - configuration, displaying • *Debugger*, 8–2, 8–6, CD–234
  - configuration, setting • *Debugger*, 8–2, 8–6, CD–159
  - debugger, DBG$DECW$DISPLAY
    - with DECwindows • *Debugger*, 1–36, D–1
  - debugger, DBG$OUTPUT • *Debugger*, 9–5, D–1
  - directing • *Librarian*, LIB–15; *Analyze/RMS_File*, ARMS–10
  - display (OUT) • *Debugger*, 7–7, C–4
- Output
  - display (OUT) (Cont.)
    - with DECwindows • *Debugger*, 1–10
  - display kind • *Debugger*, 7–18, C–1
  - formatting character string • *System Services*, SYS–165
  - from DELTA • *Delta/XDelta*, DELTA–14
  - from XDELTA • *Delta/XDelta*, DELTA–14
  - window (OUT), DECwindows • *Debugger*, 1–10
- Output data register
  - See DR11–W/DRV11–WA driver, ODR
- Output device • *Device Support (B)*, 1–75
- Output file • *SUMSLP*, SUM–3; *VAXTPU*, 5–12
  - creating • *Convert*, CONV–1
  - how effected by CONVERT • *Convert*, CONV–3
  - loading • *Convert*, CONV–1
- Output file parse option
  - See FAB$V_OFP option
- Output formatting control routine • *RTL Library*, 2–20
- Output image file • *Patch*, PAT–6
  - /OUTPUT qualifier • *Patch*, PAT–32
  - with UPDATE command • *Patch*, PAT–89
- Output operation
  - batching of • *RTL Screen Management*, 2–17
- OUTPUT parameter
  - SET built-in procedure • *VAXTPU*, 7–203
- /OUTPUT qualifier • *Debugger*, 7–21, CD–124, CD–168, CD–263; *Command Def*, CDU–42; *Librarian*, LIB–36; *Patch*, PAT–6, PAT–32; *SUMSLP*, SUM–17; *Analyze/RMS_File*, ARMS–16; *File Def Language*, FDL–42; *National Char Set*, NCS–39; *System Dump Analyzer*, SDA–162; *VAXTPU*, 5–12
  - EDIT/FDL • *File Def Language*, FDL–54
  - using with /COMPRESS • *Librarian*, LIB–15
  - using with /CROSS_REFERENCE • *Librarian*, LIB–19
  - using with /EXTRACT • *Librarian*, LIB–22
- Output record buffer address field
  - See RAB$L_RBF field
- "Output" string constant parameter to GET_INFO • *VAXTPU*, 7–177
- OUTPUT_FILE keyword • *VAXTPU*, 7–435
- "Output_file" string constant parameter to GET_INFO • *VAXTPU*, 7–174, 7–178
- OUTPUT_FILE_PARSE attribute • *File Def Language*, FDL–22
- OUTRANGE case constant • *VAXTPU*, 3–24
- Out swap
  - by suspension • *System Services Intro*, 8–14
- Overflow condition code (V) • *MACRO*, 8–15
- Overflow detection • *RTL Math*, 2–9

Overlapped vector instruction execution • *MACRO,* 10–21
/OVER qualifier • *Debugger,* CD–133, CD–189, CD–201, CD–266
/OVERRIDE=ACCESSIBILITY qualifier • *File Def Language,* FDL–22
/OVERRIDE qualifier • *Debugger,* 4–26, CD–26, CD–33, CD–168, CD–196, CD–240, CD–259
Override type • *Debugger,* 4–26
OVERSTRIKE keyword • *VAXTPU,* 7–436
Overstrike mode
  COPY_TEXT • *VAXTPU,* 7–53
  MOVE_TEXT • *VAXTPU,* 7–280
Overwrite tape file • *File Def Language,* FDL–16
OWNER attribute • *File Def Language,* FDL–22
OWNER protection code • *File Def Language,* FDL–23
OWNER secondary attribute • *File Applications,* 4–28
Ownership
  global selection
    determining • *VAXTPU,* 7–199
    losing • *VAXTPU,* 7–202
    requesting • *VAXTPU,* 7–380
  input focus
    determining • *VAXTPU,* 7–199
    losing • *VAXTPU,* 7–202
    requesting • *VAXTPU,* 7–398

P

P0BR register
  displaying • *System Dump Analyzer,* SDA–90
P0BR symbol • *System Dump Analyzer,* SDA–14
/P0IMAGE qualifier • *Linker,* LINK–13
P0LR register
  displaying • *System Dump Analyzer,* SDA–90
P0LR symbol • *System Dump Analyzer,* SDA–14
P0 page table
  displaying • *System Dump Analyzer,* SDA–127
/P0 qualifier • *System Dump Analyzer,* SDA–127
P0 region
  examining • *System Dump Analyzer,* SDA–52
  used for VMS RMS buffers • *File Applications,* 7–17
P1BR register
  displaying • *System Dump Analyzer,* SDA–90
P1BR symbol • *System Dump Analyzer,* SDA–14
P1LR register
  displaying • *System Dump Analyzer,* SDA–90
P1LR symbol • *System Dump Analyzer,* SDA–14
P1 page table
  displaying • *System Dump Analyzer,* SDA–127
/P1 qualifier • *System Dump Analyzer,* SDA–52, SDA–127
P1 region
  examining • *System Dump Analyzer,* SDA–52
Packed decimal byte
  structure for key type • *RMS,* 13–6
Packed decimal instructions • *MACRO,* 9–144
Packed decimal string • *MACRO,* 9–144
  as key type • *RMS,* 13–6
  data type • *MACRO,* 8–13
  format • *MACRO,* 3–4
  in source statement • *MACRO,* 3–4
  storing • *MACRO,* 6–74
Packed decimal string directive (.PACKED) • *MACRO,* 6–74
.PACKED directive • *MACRO,* 6–74
/PACKED qualifier • *Debugger,* CD–63, CD–88
Pad character • *Convert,* CONV–18
  how to select • *Convert,* CONV–3
  in collating sequence • *National Char Set,* NCS–10
Padding effects • *VAXTPU,* 6–11 to 6–12
  version differences • *VAXTPU,* 7–439
  with APPEND_LINE • *VAXTPU,* 7–28
  with ATTACH • *VAXTPU,* 7–35
  with COPY_TEXT • *VAXTPU,* 7–53
  with CURRENT_CHARACTER • *VAXTPU,* 7–81
  with CURRENT_LINE • *VAXTPU,* 7–86
  with CURRENT_OFFSET • *VAXTPU,* 7–88
  with ERASE_CHARACTER • *VAXTPU,* 7–119
  with ERASE_LINE • *VAXTPU,* 7–121
  with MARK • *VAXTPU,* 7–262
  with MOVE_HORIZONTAL • *VAXTPU,* 7–278
  with MOVE_TEXT • *VAXTPU,* 7–281
  with MOVE_VERTICAL • *VAXTPU,* 7–282
  with READ_FILE • *VAXTPU,* 7–297
  with SELECT • *VAXTPU,* 7–338
  with SELECT_RANGE • *VAXTPU,* 7–341
  with SET (PAD) • *VAXTPU,* 7–437
  with SPAWN • *VAXTPU,* 7–516
  with SPLIT_LINE • *VAXTPU,* 7–518
Padding records • *Convert,* CONV–3
PAD keyword • *VAXTPU,* 7–437
/PAD qualifier • *Convert,* CONV–3, CONV–18
"Pad" string constant parameter to GET_INFO • *VAXTPU,* 7–223
PAD_OVERSTRUCK_TABS keyword • *VAXTPU,* 7–439
"Pad_overstruck_tabs" string constant parameter to GET_INFO • *VAXTPU,* 7–207
Page • *System Services Intro,* 11–2

- Page (Cont.)
  - copy-on-reference • *System Services Intro*, 11–11
  - demand-zero • *System Services Intro*, 11–11
  - locking into memory • *System Services Intro*, 11–7; *System Services*, SYS–335
  - locking into working set • *System Services*, SYS–337
  - owner • *System Services Intro*, 11–5
  - ownership and protection • *System Services Intro*, 11–5
  - removing from working set • *System Services*, SYS–370
  - setting protection • *System Services*, SYS–414
  - unlocking from memory • *System Services*, SYS–526
  - unlocking from working set • *System Services*, SYS–528
- %PAGE • *Debugger*, C–6
- Page boundary • *Linker*, 3–5
- Paged dynamic storage pool
  - displaying contents • *System Dump Analyzer*, SDA–118
- Page directive (.PAGE)
  - in message source file • *Message*, MSG–25
- /PAGED qualifier • *System Dump Analyzer*, SDA–118
- Page ejection directive (.PAGE) • *MACRO*, 6–75
- Page fault • *Programming Resources*, 3–20; *Convert*, CONV–24
  - illegal • *System Dump Analyzer*, SDA–19
  - taken within driver code • *Device Support (A)*, 3–5
- Page fault cluster • *Linker*, 3–6, 5–5
- Page frame section • *System Services Intro*, 11–19
- /PAGE qualifier • *Debugger*, 7–23, CD–185
  - ALIGN command • *Patch*, PAT–38
- Page table
  - displaying • *System Dump Analyzer*, SDA–111, SDA–127
  - physical address of • *Device Support (A)*, 16–21
- Page table entry
  - allocating • *Device Support (B)*, 3–107
  - deallocating • *Device Support (B)*, 3–108
  - evaluating • *System Dump Analyzer*, SDA–48
  - examining • *System Dump Analyzer*, SDA–52
  - format • *Device Support (A)*, 16–20
  - modifying • *Device Support (A)*, E–15; *Device Support (B)*, 2–41
- PAGE_BREAK keyword • *VAXTPU*, 7–286
  - with SEARCH • *VAXTPU*, 7–327
  - with SEARCH_QUIETLY • *VAXTPU*, 7–332
- PAGE_MANAGEMENT.EXE
  - global symbols • *System Dump Analyzer*, SDA–61
- page_protection data type • *Routines Intro*, A–10t
- /PAGE_TABLES qualifier • *System Dump Analyzer*, SDA–127
- Paging file
  - See also SYS$SYSTEM:PAGEFILE.SYS
  - as system dump file • *System Dump Analyzer*, SDA–5
- Paging file section • *System Services Intro*, 11–16
  - global • *System Services Intro*, 11–15
- Paging I/O function • *Device Support (B)*, 1–40
- Parallel processing • *Programming Resources*, 4–15; *RTL Parallel Processing*, 1–1
  - initializing • *Programming Resources*, 4–16
  - subprocess
    - creating • *Programming Resources*, 4–16
    - deleting • *Programming Resources*, 4–16
  - terminating • *Programming Resources*, 4–16
  - using semaphores • *Programming Resources*, 4–17
  - using spin locks • *Programming Resources*, 4–16
- Parallel programming • *Programming Resources*, 4–18 to 4–19
- Parameter • *Librarian*, LIB–11
  - debugger command procedure • *Debugger*, 8–2, CD–44
  - for VMS RMS • *File Def Language*, FDL–2
  - how to define • *Command Def*, CDU–23, CDU–32
- PARAMETER clause
  - for DEFINE SYNTAX statement • *Command Def*, CDU–23
  - for DEFINE VERB statement • *Command Def*, CDU–32
- Parameters
  - for procedures • *VAXTPU*, 3–16 to 3–19
- “Parameter” string constant parameter to GET_INFO • *VAXTPU*, 7–180
- Parameter value
  - delimiting a • *Patch*, PAT–23
- %PARCNT • *Debugger*, 8–2, D–4
- Parent
  - of widget
    - fetching in VAXTPU • *VAXTPU*, 7–215
- “"parent"” string constant parameter to GET_INFO • *VAXTPU*, 7–215
- Parentheses
  - as precedence operator • *System Dump Analyzer*, SDA–13
  - in expressions • *VAXTPU*, 3–7
- Parent lock • *System Services Intro*, 12–11
- /PARENT qualifier • *System Dump Analyzer*, SDA–41
- Parity bit • *File Applications*, 1–8
- Parity flag • *I/O User's I*, 8–41

$PARSE macro
  for processing wildcard characters • *RMS,* 4–10
Parser
  maximum stack depth of • *VAXTPU,* 4–2
Parsers with EVE$BUILD • *VAXTPU,* G–3 to G–4
Parse service • *File Applications,* 5–8 to 5–12; *RMS,* RMS–66, RMS–67
  condition values • *RMS,* RMS–69
  control block input fields • *RMS,* RMS–67
  control block output fields • *RMS,* RMS–68
  preparing for file search • *RMS,* 4–9
  preparing for wildcard character processing • *RMS,* RMS–67
  program example • *RMS,* 4–9
  requirements for using • *RMS,* RMS–67
Parsing
  See File specification parsing
Parsing file specification
  See File specification parsing
Partial pattern assignment (@) • *VAXTPU,* 2–17
Participant
  definition of • *RTL Parallel Processing,* 1–2
Pascal
  See VAX Pascal
Pascal data type declaration • *Routines Intro,* A–38
Pascal implementation table • *Routines Intro,* A–38
Passall mode • *I/O User's I,* 5–4
Passing arguments • *System Services Intro,* 1–7
Passing mechanism • *Routines Intro,* 1–10; *System Services Intro,* 1–8; *RTL Intro,* 2–24
  by descriptor • *RTL Intro,* 3–7
  by reference • *RTL Intro,* 3–7
  by value • *RTL Intro,* 3–6
  descriptor
    code • *Routines Intro,* 1–11
    definition of • *Routines Intro,* 2–3
  for arrays • *RTL Intro,* 3–9
  for scalars • *RTL Intro,* 3–9
  for strings • *RTL Intro,* 3–10
  language extensions • *Routines Intro,* 2–6
  reference
    definition of • *Routines Intro,* 2–3
  value
    definition of • *Routines Intro,* 2–3
PASSWORD command
  in card reader batch job • *I/O User's I,* 2–2
Pasteboard • *Programming Resources,* 7–8; *Debugger,* 7–3; *RTL Screen Management,* 1–4
  creating • *Programming Resources,* 7–9
  deleting • *Programming Resources,* 7–9
Pasteboard (Cont.)
  ID • *Programming Resources,* 7–31
  sharing • *Programming Resources,* 7–31
Pasthru mode • *I/O User's I,* 8–9, 8–11, 8–24, 8–27
PAT$A_NONPAGED • *Device Support (A),* 13–20
PAT$A_NONPGD
  replaced by PAT$A_NONPAGED • *Device Support (A),* 13–20
Patch
  applying a • *Patch,* PAT–2
  sample session • *Patch,* PAT–92
PATCH
  See Patch Utility
Patch area • *Patch,* PAT–17
  allocate space • *Patch,* PAT–38
  commands that affect • *Patch,* PAT–20
  creating and accessing • *Patch,* PAT–19
  default • *Patch,* PAT–18
  depositing new data or instructions • *Patch,* PAT–55, PAT–57
  descriptor • *Patch,* PAT–18, PAT–79
  displaying size and starting address • *Patch,* PAT–87
  /INITIALIZE qualifier • *Patch,* PAT–79
  inserting new instructions • *Patch,* PAT–68
  patch area symbols • *Patch,* PAT–18, PAT–38
  resetting • *Patch,* PAT–19, PAT–43
  SET PATCH_AREA • *Patch,* PAT–79
  setting user-defined patch area • *Patch,* PAT–79
  starting address • *Patch,* PAT–38
  terminating • *Patch,* PAT–19
  used with device driver images • *Patch,* PAT–19
  used with shareable images • *Patch,* PAT–19
  user-defined • *Patch,* PAT–19, PAT–80
Patch area symbol • *Patch,* PAT–18
  created with ALIGN • *Patch,* PAT–18
  reserved by DIGITAL • *Patch,* PAT–18
PATCH command • *Patch,* PAT–25
  qualifiers • *Patch,* PAT–26
PATCH commands • *Patch,* PAT–38
  for expressing symbols and pathnames • *Patch,* PAT–14
  rules of syntax for • *Patch,* PAT–20
Patch space • *Device Support (A),* 13–20
Patch Utility (PATCH) • *Programming Resources,* 1–20
  applying patches • *Patch,* PAT–95
  commands • *Patch,* PAT–38
  DCL qualifiers • *Patch,* PAT–26
  directing output from • *Patch,* PAT–25
  examples
    interactive patch session • *Patch,* PAT–92

Patch Utility (PATCH) (Cont.)
exiting • *Patch,* PAT–25
input • *Programming Resources,* 1–20
invoking • *Patch,* PAT–25
rules of syntax • *Patch,* PAT–20
using entry and display modes • *Patch,* PAT–14
using PATCH • *Patch,* PAT–1
using patch area • *Patch,* PAT–17
using symbols • *Patch,* PAT–7
/PATCH_AREA qualifier • *Patch,* PAT–18
See also DEPOSIT command
with DEPOSIT command • *Patch,* PAT–56, PAT–57
Path block
See PB
Path name • *Patch,* PAT–12
abbreviating • *Debugger,* 5–10
commands that affect • *Patch,* PAT–14
determining value of • *Patch,* PAT–60
numeric • *Debugger,* 5–10
relation to symbol • *Debugger,* 5–9
with DECwindows • *Debugger,* 1–10
syntax • *Debugger,* 5–9
to specify scope • *Debugger,* 5–8, 5–9
with DECwindows • *Debugger,* 1–28
Path to file
file specification string address • *RMS,* 4–9
file specification string size • *RMS,* 4–9
Pattern
alternation (|) • *VAXTPU,* 2–16
anchoring • *VAXTPU,* 7–24
built-in procedures • *VAXTPU,* 2–13
compilation • *VAXTPU,* 2–18
concatenation (+) • *VAXTPU,* 2–15
execution • *VAXTPU,* 2–18
expression • *VAXTPU,* 3–11
linking (&) • *VAXTPU,* 2–15
operators • *VAXTPU,* 2–15
searching • *VAXTPU,* 2–11
Pattern assignment
partial (@) • *VAXTPU,* 2–17
PATTERN data type • *VAXTPU,* 2–11 to 2–20
Pattern matching
built-in procedures
ANCHOR • *VAXTPU,* 7–24
ANY • *VAXTPU,* 7–26
ARB • *VAXTPU,* 7–30
LINE_BEGIN • *VAXTPU,* 7–249
LINE_END • *VAXTPU,* 7–251
MATCH • *VAXTPU,* 7–264
NOTANY • *VAXTPU,* 7–284
PAGE_BREAK • *VAXTPU,* 7–286
Pattern matching
built-in procedures (Cont.)
REMAIN • *VAXTPU,* 7–312
SCAN • *VAXTPU,* 7–319
SCANL • *VAXTPU,* 7–322
SPAN • *VAXTPU,* 7–510
SPANL • *VAXTPU,* 7–512
UNANCHOR • *VAXTPU,* 7–530
Pattern operator • *MACRO,* 9–170, 9–172
PB (path block) • *System Dump Analyzer,* SDA–99
PBI
See Memory interconnect to VAXBI adapter
%PC
See PC
PC (program counter) • *System Dump Analyzer,* SDA–14
built-in symbol (%PC) • *Debugger,* 4–23, D–3
content of • *Debugger,* 2–12, 4–20
EXAMINE/INSTRUCTION command • *Debugger,* 7–9, 7–17
EXAMINE/OPERANDS command • *Debugger,* 4–20, 11–9
EXAMINE/SOURCE command • *Debugger,* 6–4, 7–6, 7–19, 7–22
examining • *Debugger,* 4–20, 11–9
with DECwindows • *Debugger,* 1–26
in a crash dump • *System Dump Analyzer,* SDA–15
scope • *Debugger,* 5–8
SHOW CALLS display • *Debugger,* 2–14, CD–214
PCA (Performance and Test Coverage Analyzer) • *Modular Procedures,* 1–12
PCB$L_ASTQFL • *Device Support (A),* E–14
PCB$L_JIB • *Device Support (A),* 7–6
PCB$L_PID • *Device Support (A),* 11–8; *Device Support (B),* 3–68, 4–5
PCB$V_SSRWAIT • *Device Support (A),* 4–9; *Device Support (B),* 3–12, 3–20, 3–22
PCB$W_ASTCNT • *Device Support (B),* 3–4, 3–6, 3–10
modifying with ADAWI instruction • *Device Support (A),* E–13
PCB$W_BIOCNT • *Device Support (A),* 2–7
PCB (process control block) • *System Dump Analyzer,* SDA–160; *Device Support (A),* 3–4, 3–5, 13–13
displaying • *System Dump Analyzer,* SDA–127
hardware • *System Dump Analyzer,* SDA–129
referring to current • *Device Support (A),* E–6
synchronizing access to • *Device Support (A),* 3–14
PCB address location • *Delta/XDelta,* DELTA–9

PCBB register

displaying • *System Dump Analyzer*, SDA–90

/PCB qualifier • *System Dump Analyzer*, SDA–127

PCB vector start symbolic address • *Delta/XDelta*, DELTA–9

;P command • *Delta/XDelta*, DELTA–32

PC symbol • *System Dump Analyzer*, SDA–14

PDT (port descriptor table) • *System Dump Analyzer*, SDA–123; *Device Support (B)*, 1–80

Pending-I/O queue • *Device Support (A)*, 3–23, 4–13, 8–1, 11–7, E–14; *Device Support (B)*, 1–38, 1–76, 3–27, 3–28, 3–37, 3–38, 3–73, 3–95

bypassing • *Device Support (A)*, 7–5; *Device Support (B)*, 3–17

length • *Device Support (B)*, 1–79, 3–28

synchronizing with driver internal queue • *Device Support (A)*, 7–5

Per-CPU database

See CPU

PERFMON spin lock • *Device Support (A)*, 3–14

Performance • *Linker*, 3–7, 4–4, 4–5, 6–8; *File Applications*, 3–1, 9–7 to 9–10

and asynchronous processing • *File Applications*, 9–9

and extension size • *File Applications*, 9–8

and fast-delete option • *File Applications*, 9–9

and global buffer count • *File Applications*, 9–9

and locate mode • *File Applications*, 9–9

and window size • *File Applications*, 9–8

buffers • *File Applications*, 9–9

deferred-write option • *File Applications*, 3–28, 9–9

effect of compression • *File Applications*, 3–16

extension size • *File Applications*, 9–9

I/O in VAXcluster • *File Applications*, 3–29

improving with null keys • *File Applications*, 3–19

improving with SHR argument • *RMS*, 4–14

in a VAXcluster • *File Applications*, 3–28

multiblock count • *File Applications*, 9–9

read-ahead option • *File Applications*, 9–9

recommendations for a VAXcluster • *File Applications*, 3–30

sequential access • *File Applications*, 9–10

stack time • *Device Support (B)*, 1–17

using Prolog 3 • *File Applications*, 3–16

window size • *File Applications*, 9–10

write-behind option • *File Applications*, 9–10

Performance analysis • *Modular Procedures*, 4–8

Performance and Test Coverage Analyzer

See PCA

Performance measurement • *RTL Parallel Processing*, 5–10

Performance measurement (Cont.)

geometric model • *RTL Parallel Processing*, 5–10 to 5–13

Performance measurement routine • *RTL Library*, 2–18

Period (.)

contents-of operator • *Debugger*, 4–7, 4–20, D–7

current entity • *Debugger*, 4–8, D–5

current location counter • *MACRO*, 3–17

PERMANENT keyword • *VAXTPU*, 7–441

Permanent mailbox

See Mailbox

"Permanent" string constant parameter to GET_ INFO • *VAXTPU*, 7–174

Permanent symbol • *MACRO*, 3–5, 3–6

Permanent symbol table • *MACRO*, D–1

Per-process common blocks • *Programming Resources*, 3–6

Per-Process page

locking in memory • *Device Support (A)*, E–16

PFN (page frame number) database • *System Dump Analyzer*, SDA–111

displaying • *System Dump Analyzer*, SDA–115

PFN (physical page number) • *Delta/XDelta*, DELTA–38

PFN database

examining with XDELTA • *Device Support (A)*, 13–13 to 13–14

PFN mapping • *Device Support (A)*, 19–5 to 19–7

deleting a page designated for • *Device Support (A)*, 19–7

modifying a page designated for • *Device Support (A)*, 19–5

PGFIPLHI bugcheck • *System Dump Analyzer*, SDA–19

PHD$L_BIOCNT • *Device Support (A)*, 2–7

PHD (process header) • *System Dump Analyzer*, SDA–160

displaying • *System Dump Analyzer*, SDA–127

/PHD qualifier • *System Dump Analyzer*, SDA–127

Phonemic text

defined • *RTL DECtalk*, 1–1

speaking • *RTL DECtalk*, DTK–35

Physical address

format • *Device Support (A)*, 19–4

Physical device name • *Routines Intro*, A–5t

Physical I/O

access checks • *System Services Intro*, 7–8

operations • *System Services Intro*, 7–7

privilege • *System Services Intro*, 7–4, 7–7

Physical I/O function • *Device Support (B)*, 1–40, 3–72

Physical name • *System Services Intro,* 7–26
Physical page number
  See PFN
PID (process identification) number • *System Services Intro,* 8–8; *Device Support (B),* 1–74
"Pid" string constant parameter to GET_INFO • *VAXTPU,* 7–192
PIO transfer • *Device Support (A),* 1–21
  example • *Device Support (A),* 2–1 to 2–7
  using buffered I/O in • *Device Support (A),* 6–8
  using I/O adapter resources in • *Device Support (A),* 14–2
Pipelining software model • *RTL Parallel Processing,* 1–4 to 1–5
Pixmap
  use of to implent icon in DECwindows VAXTPU • *VAXTPU,* 7–393, 7–395
PL/I
  See VAX PL/I
PL/I data type declaration • *Routines Intro,* A–42
PL/I implementation table • *Routines Intro,* A–42
PLACEMENT clause
  for QUALIFIER clause • *Command Def,* CDU–25, CDU–34
Plane rotation
  applying Givens plane rotation to a vector • *RTL Math,* MTH–173
  generating the elements for a Givens plane rotation • *RTL Math,* MTH–178
PMT option • *File Def Language,* FDL–14
Pn symbol • *Delta/XDelta,* DELTA–9
Pointer
  See also Message pointer
  retrieval • *File Applications,* 9–8
  structure • *Analyze/RMS_File,* ARMS–21
Pointer position • *VAXTPU,* 7–252
Pointer type • *Debugger,* 4–19
POLYD (Polynomial Evaluation D_floating) instruction • *MACRO,* 9–120
POLYF (Polynomial Evaluation F_floating) instruction • *MACRO,* 9–120
POLYG (Polynomial Evaluation G_floating) instruction • *MACRO,* 9–120
POLYH (Polynomial Evaluation H_floating) instruction • *MACRO,* 9–120
Polynomial
  evaluating • *RTL Library,* LIB–300, LIB–302, LIB–305, LIB–307
Pool checking mechanism • *Device Support (A),* 13–23 to 13–27
POOLCHECK parameter • *Device Support (A),* 13–23
POOL spin lock • *Device Support (A),* 3–14; *Device Support (B),* 3–14, 3–15, 3–19
Poor man's lockdown • *Device Support (A),* E–16 to E–17; *Device Support (B),* 2–49 to 2–50, 2–97
POPL instruction • *MACRO,* 9–27
/POP qualifier • *Debugger,* CD–71, CD–166
POPR (Pop Registers) instruction • *MACRO,* 9–79
Pop-up menu
  with DECwindows • *Debugger,* 1–12
Port • *Device Support (A),* 17–1
  displaying SDA information • *System Dump Analyzer,* SDA–123
  DMA buffer • *Device Support (A),* 17–2, 17–16, 17–27; *Device Support (B),* 2–77 to 2–79
  examining status of • *Device Support (A),* 17–17 to 17–18
  resetting • *Device Support (B),* 2–82
Port access mode • *I/O User's I,* 3–12
Port capabilities longword • *Device Support (A),* 17–13
Port command buffer
  allocating • *Device Support (A),* 17–11, 17–27; *Device Support (B),* 2–69
  deallocating • *Device Support (A),* 17–11, 17–28; *Device Support (B),* 2–72
Port driver • *Device Support (A),* 17–3
  See Terminal port driver
  displaying SDA information • *System Dump Analyzer,* SDA–82
Port driver entry vector table • *Device Support (B),* 1–34
Port driver vector table • *Device Support (A),* 18–4 to 18–5; *Device Support (B),* 1–89
  address • *Device Support (A),* 18–9; *Device Support (B),* 2–8
  creating • *Device Support (A),* 18–6; *Device Support (B),* 2–99, 2–100
  defining entry in • *Device Support (B),* 2–98
  relocating • *Device Support (B),* 2–7
Port selection • *I/O User's I,* 3–12
PORT_ABORT service routine • *Device Support (A),* 18–16
PORT_CANCEL service routine • *Device Support (A),* 18–17
PORT_DISCONNECT initiate routine • *Device Support (A),* 18–13
PORT_DS_SET initiate routine • *Device Support (A),* 18–13
PORT_FDT initiate routine • *Device Support (A),* 18–14
PORT_FORKRET initiate routine • *Device Support (A),* 18–14, 18–20

- PORT_MAINT initiate routine • *Device Support (A),* 18–15; *Device Support (B),* 1–90
- PORT_RESUME service routine • *Device Support (A),* 18–17
- PORT_SET_LINE initiate routine • *Device Support (A),* 18–15
- PORT_SET_MODEM initiate routine • *Device Support (A),* 18–15
- PORT_STARTIO initiate routine • *Device Support (A),* 18–16
- PORT_STOP service routine • *Device Support (A),* 18–17
- PORT_XOFF service routine • *Device Support (A),* 18–17
- PORT_XON service routine • *Device Support (A),* 18–18
- Positional argument • *MACRO,* 4–3
- POSITIONAL clause
  - for PLACEMENT clause • *Command Def,* CDU–25, CDU–34
- Positional qualifier
  - /INCLUDE • *Linker,* 2–4, 2–10, LINK–24
  - incompatibility among • *Linker,* LINK–23
  - /LIBRARY • *Linker,* 2–4, LINK–25
  - /OPTIONS • *Linker,* 2–4, LINK–26
  - /SELECTIVE_SEARCH • *Linker,* LINK–27
  - /SHAREABLE • *Linker,* LINK–28
- POSITION attribute • *File Applications,* 4–31; *File Def Language,* FDL–7, FDL–28, FDL–29
- POSITION/BUCKET command • *Analyze/RMS_File,* ARMS–30
- POSITION built-in procedure • *VAXTPU,* 7–287 to 7–290
  - example of use • *VAXTPU,* B–25 to B–27
- Position independence • *Modular Procedures,* 3–1, A–3
  - coding guidelines for • *Linker,* 4–5
  - desirability of • *Linker,* 4–4
  - in shareable image • *Linker,* 1–10, 4–4
- Position independent code • *Device Support (A),* 5–1
- POSITION/RECORD command • *Analyze/RMS_File,* ARMS–32
- Positive operator (+) • *System Dump Analyzer,* SDA–12
- POS option • *File Def Language,* FDL–21
- Postprocessing
  - See I/O postprocessing
- POST_KEY_PROCEDURE keyword • *VAXTPU,* 7–442
- "Post_key_procedure" string constant parameter to GET_INFO • *VAXTPU,* 7–204
- Power bit
  - See UCB$V_POWER
- Power failure • *MACRO,* 10–43
  - blocking • *Device Support (A),* 3–7
  - determining the occurrence of • *Device Support (A),* 8–5
  - occurring when device is busy • *Device Support (B),* 1–78
  - on I/O bus • *Device Support (A),* 19–7
  - recovery procedure
    - setting AST for • *System Services,* SYS–409
  - servicing in an initialization routine • *Device Support (A),* 11–1, 11–5
  - servicing in port driver unit initialization routine • *Device Support (A),* 18–13, 18–22
- Power failure recovery procedure • *Device Support (B),* 1–25, 1–26, 1–74
  - device timeout forced by • *Device Support (A),* 10–5
  - initialization performed by • *Device Support (A),* 11–5
- PPL$ADJUST_QUORUM • *RTL Parallel Processing,* 4–4, PPL–3
- PPL$ADJUST_SEMAPHORE_MAXIMUM • *RTL Parallel Processing,* 4–13, PPL–5
- PPL$AWAIT_EVENT • *RTL Parallel Processing,* 4–7, PPL–7
- PPL$CREATE_APPLICATION • *RTL Parallel Processing,* 2–1, PPL–9
- PPL$CREATE_BARRIER • *RTL Parallel Processing,* 4–2, PPL–14
- PPL$CREATE_EVENT • *RTL Parallel Processing,* 4–5, PPL–16
- PPL$CREATE_PROCESS • *Programming Resources,* 4–16
- PPL$CREATE_SEMAPHORE • *RTL Parallel Processing,* 4–11, PPL–20
- PPL$CREATE_SHARED_MEMORY • *RTL Parallel Processing,* 3–1, PPL–23
- PPL$CREATE_SPIN_LOCK • *RTL Parallel Processing,* 4–14, PPL–27
- PPL$CREATE_VM_ZONE • *RTL Parallel Processing,* 3–4, PPL–29
- PPL$CREATE_WORK_QUEUE • *RTL Parallel Processing,* 4–16, PPL–34
- PPL$DECREMENT_SEMAPHORE • *RTL Parallel Processing,* 4–12, PPL–36
- PPL$DELETE_APPLICATION • *RTL Parallel Processing,* 2–2, PPL–38
- PPL$DELETE_BARRIER • *RTL Parallel Processing,* 4–3, PPL–39
- PPL$DELETE_EVENT • *RTL Parallel Processing,* 4–6, PPL–41

PPL$DELETE_SEMAPHORE • *RTL Parallel Processing*, 4–12, PPL–43
PPL$DELETE_SHARED_MEMORY • *RTL Parallel Processing*, 3–3, PPL–45
PPL$DELETE_SPIN_LOCK • *RTL Parallel Processing*, 4–15, PPL–47
PPL$DELETE_VM_ZONE • *RTL Parallel Processing*, 3–4, PPL–49
PPL$DELETE_WORK_ITEM • *RTL Parallel Processing*, 4–18, PPL–51
PPL$DELETE_WORK_QUEUE • *RTL Parallel Processing*, 4–17, PPL–53
PPL$DISABLE_EVENT • *RTL Parallel Processing*, 4–7, PPL–55
PPL$ENABLE_EVENT_AST • *RTL Parallel Processing*, 4–6, PPL–56
PPL$ENABLE_EVENT_SIGNAL • *RTL Parallel Processing*, 4–7, PPL–59
PPL$FIND_OBJECT_ID • *RTL Parallel Processing*, 4–1, PPL–63
PPL$FLUSH_SHARED_MEMORY • *RTL Parallel Processing*, 3–3, PPL–65
PPL$GET_INDEX • *RTL Parallel Processing*, 2–4, PPL–67
PPL$INCREMENT_SEMAPHORE • *RTL Parallel Processing*, 4–13, PPL–68
PPL$INDEX_TO_PID • *RTL Parallel Processing*, 2–4, PPL–69
PPL$INSERT_WORK_ITEM • *RTL Parallel Processing*, 4–17, PPL–71
PPL$PID_TO_INDEX • *RTL Parallel Processing*, 2–4, PPL–73
PPL$READ_BARRIER • *RTL Parallel Processing*, 4–3, PPL–75
PPL$READ_EVENT • *RTL Parallel Processing*, 4–8, PPL–77
PPL$READ_SEMAPHORE • *RTL Parallel Processing*, 4–13, PPL–79
PPL$READ_SPIN_LOCK • *RTL Parallel Processing*, 4–16, PPL–81
PPL$READ_WORK_QUEUE • *RTL Parallel Processing*, 4–17, PPL–83
PPL$RELEASE_SPIN_LOCK • *RTL Parallel Processing*, 4–15, PPL–85
PPL$REMOVE_WORK_ITEM • *RTL Parallel Processing*, 4–18, PPL–86
PPL$RESET_EVENT • *RTL Parallel Processing*, 4–8, PPL–88
PPL$ routines • *Programming Resources*, 4–15
PPL$SEIZE_SPIN_LOCK • *RTL Parallel Processing*, 4–15, PPL–89
PPL$SET_QUORUM • *RTL Parallel Processing*, 4–4, PPL–91
PPL$SET_SEMAPHORE_MAXIMUM • *RTL Parallel Processing*, 4–14, PPL–93
PPL$SPAWN • *RTL Parallel Processing*, 2–3, PPL–95
PPL$STOP • *RTL Parallel Processing*, 2–3, PPL–99
PPL$TERMINATE • *RTL Parallel Processing*, 2–2, PPL–100
PPL$TRIGGER_EVENT • *RTL Parallel Processing*, 4–8, PPL–101
PPL$UNIQUE_NAME • *RTL Parallel Processing*, 2–4, PPL–103
PPL$WAIT_AT_BARRIER • *RTL Parallel Processing*, 4–3, PPL–105
PPL$_INSVIRMEM
  reasons for error • *RTL Parallel Processing*, PPL–11
PR$_ASTLVL processor register • *Device Support (A)*, 3–4
PR$_SID processor register • *Device Support (B)*, 1–17
PR$_SIRR processor register • *Device Support (A)*, 3–9; *Device Support (B)*, 2–67
PR$_TBIA processor register • *Device Support (A)*, E–15
PR$_TBIS processor register • *Device Support (A)*, E–15
Precedence of operators • *System Dump Analyzer*, SDA–12
Precedence operator • *System Dump Analyzer*, SDA–13
Predecessor
  See Logical predecessor
Predefined constants
  names • *VAXTPU*, 3–13
Predefined logical name
  LNM$FILE_DEV • *System Services Intro*, 6–12
/PREDEFINED qualifier • *Debugger*, CD–15, CD–18, CD–31, CD–212, CD–257
Prefetch function of UNIBUS adapter • *Device Support (A)*, 14–3, 14–12, 14–13
/PREFIX qualifier
  in .FACILITY directive • *Message*, MSG–18
Preprocessing
  See I/O preprocessing
Preprocessing routine
  See FDT routine
Previous location
  See Logical predecessor
"Previous" string constant parameter to GET_INFO • *VAXTPU*, 7–166, 7–168, 7–169, 7–180, 7–181, 7–183, 7–184, 7–191, 7–218, 7–223
%PREVIOUS_PROCESS • *Debugger*, 10–12
%PREVIOUS_SCOPE_ENTRY • *Debugger*, D–11

%PREVLOC • *Debugger*, 4–8, D–5
PRE_KEY_PROCEDURE keyword • *VAXTPU*, 7–444
"Pre_key_procedure" string constant parameter to GET_INFO • *VAXTPU*, 7–204
Primary attribute • *File Applications*, 4–9; *File Def Language*, FDL–1
Primary data record • *Analyze/RMS_File*, ARMS–6
Primary exception vector • *Programming Resources*, 9–13
Primary handler • *Debugger*, 3–22, 9–13
Primary index structure • *Analyze/RMS_File*, ARMS–6
Primary key • *Convert*, CONV–16
Primary operand • *MACRO*, 8–26
Primary processor • *Device Support (A)*, E–2
Primary record structure • *File Applications*, 10–20
PRIMITIVE_IO.EXE
  global symbols • *System Dump Analyzer*, SDA–61
PRINT carriage control • *Convert*, CONV–2; *File Def Language*, FDL–34
.PRINT directive • *MACRO*, 6–76
Printer
  See Line printer
Printer device width • *Programming Resources*, 7–6
Printer driver
  description • *Device Support (A)*, 2–1 to 2–7
Print format option
  See FAB$V_PRN option
Print format options for VFC records with 2-byte control area • *RMS*, 5–25
Print queue • *File Def Language*, FDL–23
Print symbiont
  See Symbiont
  invoking • *Utility Routines*, PSM–23
Print Symbiont Modification routines
  See PSM routines
PRINT_ON_CLOSE attribute • *File Def Language*, FDL–23
Priority
  of work queue • *RTL Parallel Processing*, 4–16
  setting • *System Services*, SYS–411
/PRIORITY qualifier • *Debugger*, CD–183, CD–254
Private section
  defining • *System Services Intro*, 11–8
Privilege • *System Services Intro*, 6–6
  BYPASS • *System Services Intro*, 7–6
  defined by access mode • *System Services Intro*, 2–2
  DELTA • *Delta/XDelta*, DELTA–14
  I/O operations • *System Services Intro*, 7–2
  logical I/O • *System Services Intro*, 7–4, 7–7
  MOUNT • *System Services Intro*, 7–4
Privilege (Cont.)
  physical I/O • *System Services Intro*, 7–4, 7–7
  PRMGBL • *RTL Parallel Processing*, 1–6
  required to analyze VAX RMS Journaling files • *Analyze/RMS_File*, ARMS–11
  setting for process • *System Services*, SYS–417
  SS$_NOPRIV • *Programming Resources*, 9–3
  SYSGBL • *RTL Parallel Processing*, 1–6
  SYSLCK • *RTL Parallel Processing*, 1–6
  SYSTEM • *System Services Intro*, 7–6
  user • *System Services Intro*, 2–2
  XDELTA • *Delta/XDelta*, DELTA–14
Privileged image
  installing • *Programming Resources*, 6–2
Privileged shareable image
  See also User-written system service
  creation of • *Linker*, 1–11, 4–11
  definition of • *Linker*, 1–11, 4–11
PROBER (Probe Read) instruction • *MACRO*, 9–188
PROBEW (Probe Write) instruction • *MACRO*, 9–188
Procedural error handler • *VAXTPU*, 3–26 to 3–28
Procedure
  definition of • *Routines Intro*, 2–3
  entry mask • *Modular Procedures*, 3–11
  entry point names • *Modular Procedures*, 3–3
  executing • *VAXTPU*, 4–21
  grouping • *Modular Procedures*, 5–1
  interface • *Modular Procedures*, 2–3, A–2
  language support
    definition of • *Routines Intro*, 2–4
    use of • *Routines Intro*, 2–4
  libraries • *Modular Procedures*, 5–1
  library
    definition of • *Routines Intro*, 2–4
    use of • *Routines Intro*, 2–4
  name • *VAXTPU*, 3–16
  operation • *Routines Intro*, A–7t
  parameter • *VAXTPU*, 3–16 to 3–19
  recommended naming conventions • *VAXTPU*, 4–31
  recommended size for • *VAXTPU*, 4–2
  recursive • *VAXTPU*, 3–19
  returning result • *VAXTPU*, 2–8, 3–19, 7–101
  samples using EVE • *VAXTPU*, B–1 to B–33
  using LEARN_ABORT in • *VAXTPU*, 7–243
Procedure call format • *Routines Intro*, 1–3
Procedure call instructions • *MACRO*, 9–63
procedure data type • *Routines Intro*, A–11t
Procedure descriptor • *Routines Intro*, 2–29
PROCEDURES keyword
  with EXPAND_NAME • *VAXTPU*, 7–135

PROCEDURE statement • *VAXTPU,* 3–15 to 3–21
"Procedure" string constant parameter to GET_INFO • *VAXTPU,* 7–180
Proceed from Breakpoint command • *Delta/XDelta,* DELTA–32
Process
  See also Process quota
  activation tracepoint, predefined • *Debugger,* 10–14
  channel • *System Dump Analyzer,* SDA–126
  communicating between • *Programming Resources,* 3–7
  communicating within • *Programming Resources,* 3–1
    using logical names • *Programming Resources,* 3–2
    using symbols • *Programming Resources,* 3–5
  connecting debugger to • *Debugger,* 10–4, 10–16, CD–36
  creating • *Programming Resources,* 2–1; *System Services Intro,* 8–2; *System Services,* SYS–88
  creation restriction • *System Services Intro,* 8–7
  current • *Device Support (B),* 1–15
  deadlock • *RTL Parallel Processing,* 5–4
  definition of • *RTL Parallel Processing,* 1–2
  deleting • *Programming Resources,* 2–15; *System Services Intro,* 8–17; *System Services,* SYS–132; *VAXTPU,* 7–108
  detached • *Programming Resources,* 2–7; *System Services Intro,* 8–2, 8–7
  disabling swap mode • *System Services Intro,* 11–7
  disallowing swapping • *System Services Intro,* 11–7
  displaying SDA information • *System Dump Analyzer,* SDA–126, SDA–159
  examining a hung • *System Dump Analyzer,* SDA–8
  execution • *Programming Resources,* 2–14
  getting information about
    asynchronously • *System Services,* SYS–222
    synchronously • *System Services,* SYS–238
  hibernating • *System Services Intro,* 8–11; *System Services,* SYS–330
  how to set writable • *Delta/XDelta,* DELTA–43
  identification • *System Services Intro,* 8–8
  image • *System Dump Analyzer,* SDA–159
  information • *System Services Intro,* 8–10
  listening • *System Dump Analyzer,* SDA–83
  lock • *System Dump Analyzer,* SDA–127
  modes of execution • *Programming Resources,* 2–1

Process (Cont.)
  modifying name • *Programming Resources,* 2–13
  multiple
    built-in procedures
      ATTACH • *VAXTPU,* 7–35
      CREATE_PROCESS • *VAXTPU,* 7–67
      RECOVER_BUFFER • *VAXTPU,* 7–307
      SEND • *VAXTPU,* 7–342
      SEND_EOF • *VAXTPU,* 7–346
      SPAWN • *VAXTPU,* 7–515
  multiprocess debugging • *Debugger,* 10–1
    with DECwindows • *Debugger,* 1–8, 1–30
  name • *System Services Intro,* 8–8
  name within group • *System Services Intro,* 8–9
  obtaining information • *Programming Resources,* 2–9
    using LIB$GETJPI • *Programming Resources,* 2–9
    using SYS$GETJPI • *Programming Resources,* 2–9
    using SYS$GETJPIW • *Programming Resources,* 2–9
  priority
    modifying • *Programming Resources,* 2–12
  privilege mask • *Device Support (B),* 1–42
  privileges
    setting • *Programming Resources,* 2–12
  quantum end event • *Device Support (A),* 3–8
  resource limits • *File Applications,* 1–16
  resuming after suspension • *System Services,* SYS–391
  returning control from driver to • *Device Support (A),* 4–16
  scheduling • *Programming Resources,* 2–12
  scheduling state • *System Dump Analyzer,* SDA–129, SDA–159
  scheduling wakeup for • *System Services,* SYS–397
  setting name of • *System Services,* SYS–413
  setting priority of • *System Services,* SYS–411
  setting privilege • *System Services,* SYS–417
  setting swap mode for • *System Services,* SYS–429
  spawning a subprocess • *System Dump Analyzer,* SDA–162
  subprocess • *System Services Intro,* 8–2
  suspending • *System Services Intro,* 8–11, 8–14; *System Services,* SYS–509
  swapping • *System Services Intro,* 11–7
  swapping by suspension • *System Services Intro,* 8–14
  termination mailbox • *System Services Intro,* 7–34, 8–18

Process (Cont.)
termination tracepoint, predefined • *Debugger*, 10–14
types of resources • *File Applications*, 1–15
asynchronous system trap limit (ASTLM) • *File Applications*, 1–17
buffered I/O limit (BIOLM) • *File Applications*, 1–17
I/O limit (DIOLM) • *File Applications*, 1–17
waiting for entire set of event flags • *System Services*, SYS–540
waiting for event flag to be set • *System Services*, SYS–537
waiting for one of set of event flags • *System Services*, SYS–542
waking • *System Services*, SYS–538
Process command table • *Command Def*, CDU–2
adding commands to • *Command Def*, CDU–3, CDU–45
deleting commands from • *Command Def*, CDU–39
Process context • *Device Support (A)*, 1–8, 2–4, 4–13, 7–1
changing • *System Dump Analyzer*, SDA–68, SDA–73, SDA–93, SDA–126
returning to • *Device Support (A)*, 4–20
Process control block
See PCB
Process control region • *System Dump Analyzer*, SDA–14
Process control region operator (H) • *System Dump Analyzer*, SDA–12
Process control services • *System Services Intro*, 1–2
PROCESS data type • *VAXTPU*, 2–20 to 2–21
Process default • *File Applications*, 4–14; *File Def Language*, FDL–30
batch queue • *File Def Language*, FDL–24
print queue • *File Def Language*, FDL–23
Process directory table • *System Services Intro*, 6–3
Process header
See PHD
Process I/O channel • *Device Support (A)*, 11–6; *Device Support (B)*, 1–11, 1–40
assigning • *Device Support (A)*, 4–5
assigning to template device • *Device Support (A)*, 11–12
deassigning • *Device Support (A)*, 11–7, 11–8, 18–13; *Device Support (B)*, 4–4
reference count • *Device Support (B)*, 1–77, 1–78
validating • *Device Support (A)*, 2–3, 4–5; *Device Support (B)*, 3–103
Process I/O segment • *File Applications*, 1–16
Process identification • *System Dump Analyzer*, SDA–126
See also PID number
Process index • *System Dump Analyzer*, SDA–126
Process index number • *System Services*, SYS–230
Processing
deferred-write option • *File Applications*, 3–15, 3–27
options for improving file performance • *File Applications*, 3–7
read-ahead option • *File Applications*, 3–11, 3–12
write-behind option • *File Applications*, 3–11, 3–12
Process logical name table • *System Services Intro*, 6–4
Process management • *Programming Resources*, 2–8
Process name • *System Dump Analyzer*, SDA–126
Processor
synchronization • *Programming Resources*, 4–18
Processor context
changing • *System Dump Analyzer*, SDA–68, SDA–74, SDA–89, SDA–93, SDA–126
Processor register symbol • *Delta/XDelta*, DELTA–9
Processor-specific loadable code
base address • *System Dump Analyzer*, SDA–14
Processor state
See Multiprocessor state
Processor status longword
See PSL
Processor status longword symbol • *Delta/XDelta*, DELTA–9, DELTA–13
See also PSL
Processor status word
See PSW
Processor subtype • *Device Support (B)*, 2–9
Processor type • *Device Support (B)*, 2–9
displaying • *System Dump Analyzer*, SDA–90
Process-permanent files • *File Applications*, 1–16, 6–20
access to • *File Applications*, 6–20
implications for indirect access • *File Applications*, 6–21
Process-permanent I/O structures • *System Dump Analyzer*, SDA–77
/PROCESS qualifier • *Debugger*, 10–6, 10–16, CD–71, CD–76; *System Dump Analyzer*, SDA–163
Process quota
adjusting • *Device Support (A)*, 4–20
buffered I/O • *Device Support (A)*, 2–3, 2–7, 4–9
byte count • *Device Support (A)*, 7–8
charging • *Device Support (A)*, 4–9, 4–12; *Device Support (B)*, 1–41, 4–17

Process quota (Cont.)
  direct I/O • *Device Support (A)*, 4–9
  symbolic names for (PQL$_xxxx) • *System Services*, SYS–91
Process rights list • *Programming Resources*, 6–1; *System Services Intro*, 3–2
Process section table
  See PST
/PROCESS_GROUP qualifier • *Debugger*, 10–14, CD–54
process_id data type • *Routines Intro*, A–11t
PROCESS_MANAGEMENT.EXE
  global symbols • *System Dump Analyzer*, SDA–61
%PROCESS_NAME • *Debugger*, 10–12
process_name data type • *Routines Intro*, A–11t
%PROCESS_NUMBER • *Debugger*, 10–12
%PROCESS_PID • *Debugger*, 10–12
/PROCESS_SECTION_TABLE qualifier • *System Dump Analyzer*, SDA–127
Product
  of a vector • *RTL Math*, MTH–165
Program
  add to section file • *VAXTPU*, 4–25
  calling VAXTPU from • *VAXTPU*, 4–1, 7–41
  compiling • *VAXTPU*, 4–18 to 4–19
  complex • *VAXTPU*, 4–2
  creating • *Message*, MSG–4
  debugging • *VAXTPU*, 4–33 to 4–37
  deleting • *VAXTPU*, 7–108
  display kind • *Debugger*, 7–19, C–1
  executing • *Message*, MSG–4; *VAXTPU*, 4–19 to 4–21
  interrupting • *VAXTPU*, 4–20
  order • *VAXTPU*, 4–3
  simple • *VAXTPU*, 4–2
  syntax • *VAXTPU*, 4–3
    example • *VAXTPU*, 4–4
  using wildcard characters • *RMS*, 4–12
  writing • *VAXTPU*, 4–1 to 4–14
Program counter
  See PC
Program counter mode • *MACRO*, 5–12
  summary • *MACRO*, 8–29
PROGRAM data type • *VAXTPU*, 2–21
Program decomposition • *Programming Resources*, 4–18
Program execution
  See also Synchronization
  built-in procedures
    COMPILE • *VAXTPU*, 7–47
    SAVE • *VAXTPU*, 7–316
  continuing • *Delta/XDelta*, DELTA–33
Program execution (Cont.)
  proceeding from breakpoint • *Delta/XDelta*, DELTA–32
  specifying a time • *Programming Resources*, 4–8, 4–9
  step execution • *Delta/XDelta*, DELTA–34
  step over subroutine execution • *Delta/XDelta*, DELTA–35
  timed intervals • *Programming Resources*, 4–10
Program execution mode
  using to call services • *RMS*, 2–7
Program execution time
  delaying • *MACRO*, 9–78
Program interface • *RMS*, 2–1
  to VMS RMS • *RMS*, 2–1
PROGRAM keyword • *VAXTPU*, 7–362
  with LOOK_UP_KEY • *VAXTPU*, 7–254
Programmed I/O
  See PIO transfer
Programming examples
  interpreting • *System Services Intro*, 2–18
Programming language
  using control blocks with • *RMS*, 2–1
Programming rules • *RMS*, 3–6
/PROGRAM qualifier • *Debugger*, 7–21, CD–124
Program region • *System Services Intro*, 11–2
  adding page to • *System Services*, SYS–163
  base register • *System Dump Analyzer*, SDA–14
  examining • *System Dump Analyzer*, SDA–52
  length register • *System Dump Analyzer*, SDA–14
Program region page table
  displaying • *System Dump Analyzer*, SDA–127
Program section
  See PSECT
  absolute • *MACRO*, 6–80
  alignment • *MACRO*, 6–80
  attributes • *MACRO*, 6–77, 6–80
  defining • *MACRO*, 6–77
  directive
    (.PSECT) • *MACRO*, 6–77
    (.RESTORE_PSECT) • *MACRO*, 6–86
    (.SAVE_PSECT) • *MACRO*, 6–87
  name • *MACRO*, 6–77, 6–80
  restoring context of • *MACRO*, 6–86
  saving context of • *MACRO*, 6–87
  saving local label • *MACRO*, 6–87
  unnamed • *MACRO*, 6–80
PROHIBIT attribute • *File Def Language*, FDL–37
PROHIBIT secondary attribute • *File Applications*, 7–4
Prolog • *File Applications*, 3–12, 3–15, 3–16, 3–19
Prolog 1 • *File Applications*, 3–16

Prolog 2 • *File Applications*, 3–16
Prolog 3 • *File Applications*, 3–16, 10–30
Prolog 3 file • *Convert*, CONV–1; *File Def Language*, FDL–27
  compression • *File Def Language*, FDL–27, FDL–28
  creating with CONV routines • *Utility Routines*, CONV–15
  key segment length • *File Def Language*, FDL–30
  key segment position • *File Def Language*, FDL–30
Prolog 3 indexed files
  reclaiming • *Utility Routines*, CONV–18
  with Convert/Reclaim Utility • *Utility Routines*, CONV–1
PROLOG attribute • *Convert*, CONV–19; *File Def Language*, FDL–27, FDL–28, FDL–29
Prolog field
  See XAB$B_PROLOG field
Prolog files
  with CONV routines • *Utility Routines*, CONV–15
Prolog level • *RMS*, RMS–18
/PROLOG qualifier • *Convert*, CONV–19
PROLOG structure • *File Applications*, 10–16, 10–19
Prolog version number field
  See XAB$W_PVN field
Prompt
  COMMAND box, DECwindows • *Debugger*, 1–28
  debugger (DBG>) • *Debugger*, 2–7, 10–2, CD–165
    with DECwindows • *Debugger*, 1–28
  display (PROMPT) • *Debugger*, 7–7, C–4
  ECO level • *Patch*, PAT–45, PAT–47
  ending repetitive • *Patch*, PAT–65
  multiprocess program • *Debugger*, 10–2
Prompt buffer address field
  See RAB$L_PBF field
Prompt buffer size field
  See RAB$B_PSZ field
PROMPT clause
  for PARAMETER clause • *Command Def*, CDU–23, CDU–32
Prompt for input
  with LIB$GET_INPUT • *Programming Resources*, 7–4
/PROMPTING qualifier • *File Def Language*, FDL–42, FDL–55
Prompt option
  See RAB$V_PMT option
/PROMPT qualifier • *Debugger*, 7–21, CD–124
Prompt string
  setting with CLI$DCL_PARSE • *Utility Routines*, CLI–8
PROMPT_AREA
  video attributes • *VAXTPU*, 7–446
PROMPT_AREA keyword • *VAXTPU*, 7–446
"Prompt_length" string constant parameter to GET_INFO • *VAXTPU*, 7–200
"Prompt_row" string constant parameter to GET_INFO • *VAXTPU*, 7–201
Properties of condition handler • *Routines Intro*, 2–49
Protected shareable image
  See also User-written system service
Protection
  See also Mailbox
  access category • *File Applications*, 4–21
  ACL-based • *File Applications*, 1–10, 4–21
  by access mode • *System Services Intro*, 2–2
  cluster • *Linker*, 1–8, 3–10, LINK–14
  device • *System Services Intro*, 7–6
  directory entry • *I/O User's I*, 1–9
  disk and tape volumes • *File Applications*, 1–10
  I/O operations • *System Services Intro*, 7–2
  image section • *Linker*, 5–6
  mailbox • *System Services Intro*, 7–4, 7–5
  page • *System Services Intro*, 11–5
  queue • *System Services*, SYS–488
  setting for page • *System Services*, SYS–414
  shareable image • *Linker*, LINK–14
  UIC-based • *File Applications*, 1–10, 4–21
  volume • *System Services Intro*, 7–4
PROTECTION attribute • *File Def Language*, FDL–23
Protection code • *File Def Language*, FDL–23
Protection extended address block
  See XABPRO block
Protection mask • *System Services Intro*, 7–4
PROTECTION secondary attribute • *File Applications*, 4–28
/PROTECT qualifier • *Linker*, LINK–14
Protocol
  DMC11/DMR11 driver • *I/O User's II*, 1–1, 1–8
  DMP11/DMF32 driver • *I/O User's II*, 2–1
$PRTCTEND macro • *Device Support (A)*, 16–13, 16–14
$PRTCTINI macro • *Device Support (A)*, 16–13, 16–14
$PRTDEF macro • *Routines Intro*, A–10t
PSECT (program section) • *Modular Procedures*, 2–13, 3–5, A–3
  absolute • *Linker*, 1–12, 6–4
  alignment • *Linker*, 1–12, 6–4
    in map • *Linker*, 5–6
  attributes • *Linker*, 1–9, 3–11, 4–3, 6–3, 6–4, 6–5, 6–6

PSECT (program section) (Cont.)
- base address of, in map • *Linker*, 5–6
- DIGITAL-written • *Modular Procedures*, 3–5
- executable • *Linker*, 6–5
- global • *Linker*, 6–5, 6–12
- in image section generation • *Linker*, 6–3
- length of, in map • *Linker*, 5–6
- LIB$INITIALIZE • *Modular Procedures*, 3–17
- local • *Linker*, 6–5, 6–12
- location controls • *Linker*, 1–13
- modification of attributes • *Linker*, 1–12, 6–3
- module contribution to • *Linker*, 6–4
- module contribution to, in map • *Linker*, 5–6
- name • *Linker*, 1–12, 6–4
- name of, in map • *Linker*, 5–6
- nonexecutable • *Linker*, 6–5
- nonposition-independent • *Linker*, 6–6
- nonshareable • *Linker*, 6–6
- nonwritable • *Linker*, 6–6
- ordering of, in image section • *Linker*, 6–16
- position-independent • *Linker*, 6–6
- relocatable • *Linker*, 1–12, 6–4
- shareable • *Linker*, 6–6
- significant attributes of • *Linker*, 6–15, 6–16
- size • *Linker*, 1–12, 6–4
- summary • *Linker*, 1–12
- user-written • *Modular Procedures*, 3–5
- writable • *Linker*, 6–6

.PSECT directive • *MACRO*, 6–77

Pseudoterminal
- canceling request • *I/O User's I*, 9–2
- control connection routines • *I/O User's I*, C–1
- creating • *I/O User's I*, 9–1
- deleting • *I/O User's I*, 9–2
- device characteristics • *I/O User's I*, 9–3
- driver • *I/O User's I*, 9–1
- event notification • *I/O User's I*, 9–6
- features • *I/O User's I*, 9–3
- flow control • *I/O User's I*, 9–6
- I/O buffers • *I/O User's I*, 9–4
- programming example • *I/O User's I*, 9–8
- reading data • *I/O User's I*, 9–5
- using write with echo • *I/O User's I*, 9–5
- writing data • *I/O User's I*, 9–5

%PSL • *Debugger*, 4–23, D–3

PSL (processor status longword) • *Debugger*, 4–24; *System Dump Analyzer*, SDA–14; *MACRO*, 8–14
- evaluating • *System Dump Analyzer*, SDA–22, SDA–48
- examining • *System Dump Analyzer*, SDA–52

PSL (processor status longword) (Cont.)
- examining with XDELTA • *Device Support (A)*, 13–10
- symbol • *System Dump Analyzer*, SDA–14
- Z condition code • *Device Support (B)*, 3–27

/PSL qualifier • *Debugger*, CD–88; *System Dump Analyzer*, SDA–52

PSM$PRINT routine • *Utility Routines*, PSM–23

PSM$READ_ITEM_DX routine • *Utility Routines*, PSM–25

PSM$REPLACE routine • *Utility Routines*, PSM–27

PSM$REPORT routine • *Utility Routines*, PSM–32

PSM$_FUNNOTSUP • *Utility Routines*, PSM–36

PSM routines
- examples • *Utility Routines*, PSM–18 to PSM–22
- introduction • *Utility Routines*, PSM–1
- user-written
  - USER-FORMAT-ROUTINE • *Utility Routines*, PSM–35
  - USER-INPUT-ROUTINE • *Utility Routines*, PSM–40
  - USER-OUTPUT-ROUTINE • *Utility Routines*, PSM–46

PST (process section table)
- displaying • *System Dump Analyzer*, SDA–127

PSW (processor status word) • *MACRO*, 8–14
- condition codes • *MACRO*, 8–14
- decimal overflow enable (DV) • *MACRO*, 8–16
- floating underflow enable (FU) • *MACRO*, 8–16
- integer overflow enable (IV) • *MACRO*, 8–15
- trace trap enable (T) • *MACRO*, 8–15

/PSW qualifier • *Debugger*, CD–88

PTA option • *File Def Language*, FDL–14

PTD$CANCEL control connection routine • *I/O User's I*, C–2

PTD$CREATE control connection routine • *I/O User's I*, C–3

PTD$DELETE control connection routine • *I/O User's I*, C–6

PTD$READ control connection routine • *I/O User's I*, C–7

PTD$SET_EVENT_NOTIFICATION control connection routine • *I/O User's I*, C–9

PTD$WRITE control connection routine • *I/O User's I*, C–12

/PTE qualifier • *System Dump Analyzer*, SDA–48, SDA–52

PURDPR macro • *Device Support (A)*, 14–24; *Device Support (B)*, 2–51, 3–82
- detecting memory errors using • *Device Support (A)*, 14–25

Purge type-ahead option
- See RAB$V_PTA option

PUSHAB (Push Address Byte) instruction • *MACRO*, 9–35
PUSHAD (Push Address D_floating) instruction • *MACRO*, 9–35
PUSHAF (Push Address F_floating) instruction • *MACRO*, 9–35
PUSHAG (Push Address G_floating) instruction • *MACRO*, 9–35
PUSHAH (Push Address H_floating) instruction • *MACRO*, 9–35
PUSHAL (Push Address Long) instruction • *MACRO*, 9–35
PUSHAQ (Push Address Quad) instruction • *MACRO*, 9–35
PUSHAW (Push Address Word) instruction • *MACRO*, 9–35
PUSHL (Push Long) instruction • *MACRO*, 9–27
/PUSH qualifier • *Debugger*, CD–73
PUSHR (Push Registers) instruction • *MACRO*, 9–80
PUT attribute • *File Def Language*, FDL–3, FDL–37
$PUT macro
  program example • *RMS*, 4–16
$PUTMSG • *Message*, MSG–2; *RTL Library*, 4–4, 4–13, 4–16, 4–27
PUT option • *File Def Language*, FDL–3, FDL–37
PUT secondary attribute • *File Applications*, 7–3, 7–4
Put service • *File Applications*, 8–1, 8–3 to 8–4; *RMS*, RMS–70
  and next record • *File Applications*, 8–16
  condition values • *RMS*, RMS–74
    See also Completion status code
  control block input fields • *RMS*, RMS–73
  control block output fields • *RMS*, RMS–74
  effect on next-record position • *File Applications*, 8–16
  high-level language equivalents • *File Applications*, 8–1
  inserting records by sort order • *RMS*, RMS–72
  inserting records into indexed files • *RMS*, RMS–71
  inserting records into relative files • *RMS*, RMS–71
  inserting records into sequential files • *RMS*, RMS–71
  inserting records with duplicate keys • *RMS*, RMS–72
  record-locking caution • *RMS*, RMS–72
  record-processing options • *RMS*, 7–16
  requirements for using • *RMS*, RMS–72
  run-time options • *File Applications*, 9–17 to 9–19
  update-if logic • *RMS*, RMS–72
  using RAB$V_TPT option • *RMS*, RMS–71
  using RAB$V_UIF option • *RMS*, RMS–71
Put service (Cont.)
  using with mailboxes • *RMS*, RMS–71
  using with stream format files • *RMS*, RMS–71
Put service option
  See FAB$V_PUT option
Put sharing option
  See FAB$V_PUT option

Q

Q22 bus • *Device Support (A)*, 1–16; *Device Support (B)*, 2–3
  accomplishing a DMA transfer on • *Device Support (A)*, 14–15 to 14–16, 14–19 to 14–26
  address size • *Device Support (A)*, 14–6
  device interrupt dispatching • *Device Support (A)*, 14–33 to 14–36; *Device Support (B)*, 1–22
  example of driver designed for • *Device Support (A)*, C–1 to C–29, D–1 to D–26
  I/O address space • *Device Support (A)*, 19–1, 19–4, 19–7
  I/O space • *Device Support (A)*, 14–4
  power failure • *Device Support (A)*, 19–7
  rules for configuring • *Device Support (A)*, 1–16, 14–34 to 14–35
  scatter-gather map • *Device Support (A)*, 14–4 to 14–7
Q22 bus interface
  functions • *Device Support (A)*, 14–1 to 14–15
  obtaining resources of • *Device Support (A)*, 14–16
QBUS_MULT_INTR parameter • *Device Support (A)*, 14–34
Q symbol • *Delta/XDelta*, DELTA–9
.QUAD directive • *MACRO*, 6–82
/QUAD qualifier
  ALIGN command • *Patch*, PAT–38
Quadword • *MACRO*, 8–2
/QUADWORD qualifier • *Debugger*, 11–7, 11–8, CD–63, CD–88
Quadword storage directive (.QUAD) • *MACRO*, 6–82
quadword_signed data type • *Routines Intro*, A–11t
quadword_unsigned data type • *Routines Intro*, A–11t
Qualifier • *Librarian*, LIB–13 to LIB–45; *Message*, MSG–9; *SUMSLP*, SUM–15 to SUM–20; *Convert*, CONV–5 to CONV–28
  for DCL command • *Patch*, PAT–26
  for SET COMMAND command • *Command Def*, CDU–38 to CDU–44
  how to define • *Command Def*, CDU–24, CDU–33
  mode, PATCH command • *Patch*, PAT–15

Qualifier (Cont.)
to LINK command • *Linker*, 1–2
QUALIFIER clause
for DEFINE SYNTAX statement • *Command Def*, CDU–24
for DEFINE VERB statement • *Command Def*, CDU–33
Qualifier lines
help files • *Librarian*, LIB–6
Quantum end event • *Device Support (A)*, 3–8
Queue • *RTL Library*, 2–12, LIB–251; *MACRO*, 9–82
See Work queue
absolute • *MACRO*, 9–82
creating and managing
asynchronously • *System Services*, SYS–441
synchronously • *System Services*, SYS–493
entry insertion • *RTL Library*, LIB–248
execution • *Utility Routines*, PSM–4
generic • *Utility Routines*, PSM–4
getting information about
asynchronously • *System Services*, SYS–257
synchronously • *System Services*, SYS–297
header • *MACRO*, 9–82, 9–85
inserting entries • *MACRO*, 9–82, 9–85
lock management • *System Services Intro*, 12–4
protection • *System Services*, SYS–488
removing entries • *MACRO*, 9–84, 9–87
self-relative • *RTL Library*, 2–13; *MACRO*, 9–85
stepping through • *System Dump Analyzer*, SDA–64
types of • *System Services*, SYS–485
validating • *System Dump Analyzer*, SDA–164
Queue access routine • *RTL Library*, 2–13
QUEUEAST spin lock • *Device Support (A)*, 3–13; *Device Support (B)*, 3–7
Queue I/O Request system service • *File Applications*, 7–4, 9–14
Queue information, obtaining • *Programming Resources*, 3–22
Queue instructions • *MACRO*, 9–82
Queue operations
in multiprocessing environment • *Device Support (A)*, E–13 to E–14
QUIT built-in procedure • *VAXTPU*, 7–291 to 7–292
QUIT command • *Debugger*, 3–5, CD–112; *File Def Language*, FDL–65
multiprocess program • *Debugger*, 10–9, 10–10
with DECwindows • *Debugger*, 1–21
Quorum • *System Dump Analyzer*, SDA–82
adjusting • *RTL Parallel Processing*, 4–4
setting • *RTL Parallel Processing*, 4–4
Quota • *Routines Intro*, A–9t

Quota (Cont.)
See also Process quota, Job quota
AST • *System Services Intro*, 7–4; *I/O User's I*, 3–24, 4–14, 6–13, 7–5, 7–9, 8–43
AST limit • *RTL Parallel Processing*, 1–6
buffered I/O • *System Services Intro*, 7–3; *I/O User's I*, 3–24, 6–13, 7–5; *I/O User's II*, 1–3, 2–3, 5–1
buffered I/O byte count • *System Services Intro*, 7–3; *I/O User's II*, 1–3, 1–9, 2–3, 5–1
BYTELIM • *I/O User's I*, 1–11
direct I/O • *System Services Intro*, 7–3; *I/O User's I*, 3–24, 6–13; *I/O User's II*, 1–3, 2–3
disk • *I/O User's I*, 1–33 to 1–34
enqueue • *RTL Parallel Processing*, 1–6
establishing • *System Services Intro*, 6–9
global section • *RTL Parallel Processing*, 1–7
I/O operations • *System Services Intro*, 7–2
mailbox buffer • *I/O User's I*, 7–2, 7–3, 7–5
resource • *System Services Intro*, 2–2
SS$_EXQUOTA • *Programming Resources*, 9–3
subprocess • *RTL Parallel Processing*, 1–6
Quota file transfer block • *I/O User's I*, 1–33
Quotation mark (")
ASCII string delimiter • *Debugger*, 4–17
instruction delimiter • *Debugger*, 4–22
Quote characters • *VAXTPU*, 7–112, 7–113

R

R0
use by control block store macros • *RMS*, 3–8
use in asynchronous operations • *RMS*, 2–5
RA60 disk • *I/O User's I*, 3–5
RA70 disk • *I/O User's I*, 3–5
RA90 disk • *I/O User's I*, 3–5
RAB$B_BID field • *RMS*, 7–2
RAB$B_BLN field • *RMS*, 7–3
RAB$B_KRF field • *File Applications*, 9–13, 9–15; *File Def Language*, FDL–11; *RMS*, 7–4
for selecting key path • *RMS*, 4–12
RAB$B_KSZ field • *File Applications*, 8–8, 8–9, 8–12, 9–13, 9–15, 9–18; *RMS*, 7–4
use with limit option • *RMS*, 7–13
use with search key • *RMS*, 7–12, 7–14
RAB$B_MBC field • *File Applications*, 3–11, 7–18, 9–9; *File Def Language*, FDL–12; *RMS*, 7–5
default logic • *RMS*, 7–5
performance benefit • *RMS*, 7–6
use restriction • *RMS*, 7–5, 7–6

RAB$B_MBF field • *File Applications*, 3–11, 3–26, 7–17, 7–19, 7–20, 9–9; *File Def Language*, FDL–12; *RMS*, 7–6

  use with read-ahead option • *RMS*, 7–16

  use with write-behind option • *RMS*, 7–16

RAB$B_PSZ field • *RMS*, 7–7

RAB$B_RAC field • *RMS*, 7–7

  RAB$C_KEY option • *File Applications*, 8–6, 9–10, 9–16, 9–18

  RAB$C_RFA option • *File Applications*, 8–6, 9–10, 9–16, 9–18

  RAB$C_SEQ option • *File Applications*, 8–6, 9–10, 9–16, 9–18

RAB$B_TMO field • *File Applications*, 7–12, 7–13, 9–17; *File Def Language*, FDL–13; *RMS*, 7–21

  use with RAB$V_TMO option for mailbox service • *RMS*, 7–14

  use with timeout option for terminal operation • *RMS*, 7–19

RAB$C_KEY option • *RMS*, 7–8

RAB$C_RFA option • *RMS*, 7–8

RAB$C_SEQ option • *RMS*, 7–7

RAB$L_BKT field

  as output • *RMS*, 7–2

  use with block I/O • *RMS*, 7–2

RAB$L_CKT field • *File Def Language*, FDL–9

RAB$L_CTX field • *File Def Language*, FDL–10; *RMS*, 7–3

RAB$L_FAB field • *RMS*, 7–3

RAB$L_FOP field • *File Def Language*, FDL–14

RAB$L_KBF field • *File Applications*, 8–8, 8–9, 8–12, 9–13, 9–15, 9–18; *RMS*, 7–3

  use with limit option • *RMS*, 7–13

  use with RAB$B_KSZ field • *RMS*, 7–4

  use with search key • *RMS*, 7–12, 7–14

RAB$L_PBF field • *RMS*, 7–7

RAB$L_RBF field • *File Applications*, 9–18, 9–20; *RMS*, 4–4, 7–8

RAB$L_RBZ field • *File Applications*, 9–18

RAB$L_RHB field • *File Applications*, 9–17, 9–18, 9–20; *RMS*, 7–9

RAB$L_ROP field • *File Applications*, 9–7; *File Def Language*, FDL–9, FDL–10, FDL–11, FDL–12, FDL–13, FDL–14, FDL–15; *RMS*, 7–10

  RAB$V_ASY option • *File Applications*, 8–17, 8–18, 9–9, 9–15, 9–18, 9–19, 9–20

  RAB$V_EOF option • *File Applications*, 8–14, 8–16, 9–10

  RAB$V_EQNXT option • *File Applications*, 9–12, 9–15

  RAB$V_FDL option • *File Applications*, 9–9, 9–12, 9–20

  RAB$V_KGE option • *File Applications*, 8–9, 8–10

RAB$L_ROP field (Cont.)

  RAB$V_KGT option • *File Applications*, 8–9, 8–10

  RAB$V_LIM option • *File Applications*, 9–13, 9–16

  RAB$V_LOA option • *File Applications*, 9–13, 9–18

  RAB$V_LOC option • *File Applications*, 9–9, 9–16

  RAB$V_NLK option • *File Applications*, 7–12, 9–15

  RAB$V_NXR option • *File Applications*, 7–15, 8–9, 9–16

  RAB$V_NXT option • *File Applications*, 9–13, 9–15

  RAB$V_RAH option • *File Applications*, 3–12, 9–9, 9–16

  RAB$V_REA option • *File Applications*, 7–12, 9–16

  RAB$V_RLK option • *File Applications*, 7–12, 9–16, 9–18

  RAB$V_RRL option • *File Applications*, 7–12, 9–16

  RAB$V_TMO option • *File Applications*, 7–12, 7–13, 9–17, 9–19

  RAB$V_TPT option • *File Applications*, 9–11, 9–19

  RAB$V_UIF option • *File Applications*, 8–4, 8–8, 9–11, 9–19

  RAB$V_ULK option • *File Applications*, 7–15, 9–16

  RAB$V_WAT option • *File Applications*, 7–12, 9–17

  RAB$V_WBH option • *File Applications*, 3–12, 9–10, 9–19

  specifying key match method • *RMS*, 7–5

RAB$L_STS field • *RMS*, 7–20

RAB$L_STV0 field

  for returning terminating character • *RMS*, RMS–49

RAB$L_STV field • *RMS*, 7–20

  for returning I/O status block • *RMS*, RMS–49

  for returning I/O status block from Put service • *RMS*, RMS–74

  for returning PID from Put service • *RMS*, RMS–71

  for returning process identification (PID) • *RMS*, RMS–50

  for returning record length • *RMS*, RMS–53

  using with Get service • *RMS*, RMS–50

RAB$L_UBF field • *File Applications*, 9–17; *RMS*, 7–21

RAB$L_USZ field • *File Applications*, 9–17

RAB$L_XAB field • *RMS*, 7–22

  requirement for using XABTRM • *RMS*, 18–1

RAB$V_ASY option • *RMS*, 7–11, 7–14

  use restriction • *RMS*, 7–15

RAB$V_BIO option • *RMS*, 7–11

RAB$V_CCO option • *RMS*, 7–18

RAB$V_CVT option • *RMS*, 7–19

RAB$V_EOF option • *RMS*, 7–12

RAB$V_EQNXT option • *RMS*, 7–12

  examples • *RMS*, 7–13

  specifying key match method • *RMS*, 7–5

RAB$V_ETO option
requirement for using XABTRM • *RMS*, 18–1
RAB$V_FDL option • *RMS*, 7–15
RAB$V_KGE option
See RAB$V_EQNXT option
See RAB$V_NXT option
RAB$V_LIM option • *RMS*, 7–13
RAB$V_LOA option • *RMS*, 7–13
determining fill size • *RMS*, 13–10
example of use • *RMS*, 4–8
use restriction • *RMS*, 13–4, 13–11
RAB$V_LOC option • *RMS*, 7–15
RAB$V_NLK option • *RMS*, 7–17
RAB$V_NXR option • *RMS*, 7–17
RAB$V_NXT option • *RMS*, 7–14
specifying key match method • *RMS*, 7–5
RAB$V_PMT option • *RMS*, 7–19
RAB$V_PTA option • *RMS*, 7–19
RAB$V_RAH option • *RMS*, 7–12, 7–15
default logic • *RMS*, 7–16
use restriction • *RMS*, 7–16
RAB$V_REA option • *RMS*, 7–17
use restriction • *RMS*, 7–17
RAB$V_RLK option • *RMS*, 7–18
RAB$V_RNE option • *RMS*, 7–19
RAB$V_RNF option • *RMS*, 7–19
RAB$V_RRL option • *RMS*, 7–18
RAB$V_SYNCSTS option • *RMS*, 7–16
RAB$V_TMO
for immediate mailbox service • *RMS*, 7–14
RAB$V_TMO option • *RMS*, 7–14, 7–18, 7–19
RAB$V_TPT option • *RMS*, 7–16
using with Put service • *RMS*, RMS–71
RAB$V_UIF option • *RMS*, 7–17
using with Put service • *RMS*, RMS–71
RAB$V_ULK option • *RMS*, 7–18
RAB$V_WAT option • *RMS*, 7–18, 7–19
RAB$V_WBH option • *RMS*, 7–12, 7–16
RAB$W_ISI field • *RMS*, 7–3
RAB$W_RBF • *File Applications*, 8–3
RAB$W_RFA field • *File Applications*, 8–12, 8–15, 9–17; *RMS*, 7–9
as argument to $RAB_STORE macro • *RMS*, B–12
RAB$W_RSZ field • *File Applications*, 8–3, 9–20; *RMS*, 4–4, 7–20
RAB$W_STV0 offset
alternate access to RAB$L_STV • *RMS*, 7–20
RAB$W_STV2 field
for returning length of escape sequence • *RMS*, RMS–49
RAB$W_STV2 offset
alternate access to RAB$L_STV • *RMS*, 7–20
RAB$W_USZ field • *RMS*, 7–21
use with block I/O • *RMS*, 7–22
RAB$_V_WAT option
use with timeout option for record locking • *RMS*, 7–18
RAB (record access block) • *Programming Resources*, 1–36, 8–58; *File Applications*, 1–11; *System Dump Analyzer*, SDA–77
arguments • *RMS*, 1–4
described in context of example • *RMS*, 4–4
description • *RMS*, 1–4
general description • *RMS*, 7–1
summary of fields • *RMS*, 7–1
rab data type • *Routines Intro*, A–12t
$RABDEF • *File Applications*, 5–10
$RAB macro • *RMS*, B–9
argument categories • *RMS*, B–10
$RAB_STORE macro • *RMS*, B–11
argument categories • *RMS*, B–12
requirements • *RMS*, B–12
RFA argument • *RMS*, B–12
Race condition
avoiding at AST level • *Modular Procedures*, 3–21
elimination of • *Modular Procedures*, 3–21
Radix • *Message*, MSG–7
canceling • *Debugger*, CD–26
conversion • *Debugger*, 4–11, D–5
current • *Debugger*, 4–11, CD–168
default • *System Dump Analyzer*, SDA–12
displaying • *Debugger*, CD–240
multilanguage program • *Debugger*, 9–8
of numeric constant
specifying • *VAXTPU*, 3–37
specifying • *Debugger*, 4–11, CD–168
Radix control operator • *MACRO*, 3–11
Radix modes • *Patch*, PAT–17
See also Entry and display modes
Radix operator • *Linker*, 1–7, 3–5; *Message*, MSG–7; *Patch*, PAT–17; *System Dump Analyzer*, SDA–12
RAH option • *File Def Language*, FDL–13
Random access
by key value • *File Applications*, 2–5 to 2–7, 8–6, 8–11 to 8–12
by relative record number • *File Applications*, 2–5 to 2–7, 8–6, 8–8, 8–9
by RFA (record file address) • *File Applications*, 2–7, 8–6, 8–12 to 8–13
to indexed files • *File Applications*, 2–6, 8–11 to 8–12, 8–12 to 8–13

- Random access (Cont.)
  - to relative files • *File Applications*, 2–6, 8–9, 8–12 to 8–13
  - to sequential files • *File Applications*, 2–6, 8–8, 8–12 to 8–13
  - with multibuffer count • *File Applications*, 3–26
- Random access device • *Device Support (B)*, 1–75
- Random access mode • *File Applications*, 1–2
- Random number generator • *RTL Math*, MTH–118
- Range
  - colon (:) • *Debugger*, 4–18, 11–4, 11–6, 11–7, CD–85
  - converting contents of to string format using STR • *VAXTPU*, 7–520
  - deleting • *VAXTPU*, 2–22, 7–70, 7–108
  - determining if unmodifiable records are present in • *VAXTPU*, 7–193
  - erasing • *VAXTPU*, 2–22, 7–70, 7–117
  - moving delimiters of • *VAXTPU*, 7–273
  - syntax • *MACRO*, 7–1
  - video attributes • *VAXTPU*, 2–22
- RANGE data type • *VAXTPU*, 2–21 to 2–22
- Rank
  - of spin lock • *Device Support (A)*, 3–15
- RAZ field • *MACRO*, 7–2
- RB02 disk • *I/O User's I*, 3–6
- RC25 disk • *I/O User's I*, 3–6
- RCK option • *File Def Language*, FDL–23
- RD53 disk • *I/O User's I*, 3–6
- RD54 disk • *I/O User's I*, 3–6
- RDT (response descriptor table) • *System Dump Analyzer*, SDA–148
- RDT (revision-date-time) argument • *RMS*, B–16
- READ access • *File Def Language*, FDL–23
- Read access type • *MACRO*, 8–17
- Read ahead option
  - See RAB$V_RAH option
- Read As Zero
  - See RAZ field
- Read attention AST function • *I/O User's I*, 7–9
- Read check
  - enabling • *Device Support (B)*, 1–75
- Read check option
  - See FAB$V_RCK option
- READ command • *System Dump Analyzer*, SDA–59
  - SYS$DISK • *System Dump Analyzer*, SDA–60
- READ/EXECUTIVE command • *System Dump Analyzer*, SDA–16
- Read function • *Device Support (B)*, 1–40, 1–41
  - FDT routine for • *Device Support (A)*, 7–9
  - postprocessing for • *Device Support (B)*, 3–72
- Read-no-echo option
- Read-no-echo option (Cont.)
  - See RAB$V_RNE option
- Read no filter option
  - See RAB$V_RNF option
- Read regardless of lock option
  - See RAB$V_RRL option
- Read request
  - fetching • *VAXTPU*, 7–199
- Read routine
  - fetching • *VAXTPU*, 7–174, 7–201
  - specifying • *VAXTPU*, 7–385
- Read service • *RMS*, RMS–76
  - condition values • *RMS*, RMS–78
  - control block input fields • *RMS*, RMS–77
  - control block output fields • *RMS*, RMS–77
  - requirements for using • *RMS*, RMS–77
- Read/write attributes
  - ACP-QIO interface • *I/O User's I*, 1–14
- Read/write attributes subfunction • *I/O User's I*, 1–14
- READ_AHEAD attribute • *File Def Language*, FDL–12
- READ_CHAR built-in procedure • *VAXTPU*, 7–293 to 7–294
- READ_CHECK attribute • *File Def Language*, FDL–23
- /READ_CHECK qualifier • *Convert*, CONV–20
- READ_CLIPBOARD built-in procedure • *VAXTPU*, 7–295
- READ_FILE built-in procedure • *VAXTPU*, 7–297 to 7–298
- READ_GLOBAL_SELECT built-in procedure • *VAXTPU*, 7–299
  - example of use • *VAXTPU*, B–28 to B–30, B–30 to B–31
- READ_KEY built-in procedure • *VAXTPU*, 7–301 to 7–302
- READ_LINE built-in procedure • *VAXTPU*, 7–303 to 7–305
- /READ_ONLY qualifier • *VAXTPU*, 5–13
- "Read_only" string constant parameter to GET_ INFO • *VAXTPU*, 7–178
- READ_REGARDLESS attribute • *File Def Language*, FDL–13
- READ_REGARDLESS secondary attribute • *File Applications*, 7–12
- READ_SYSTIME macro • *Device Support (A)*, E–15; *Device Support (B)*, 2–52
  - example • *Device Support (B)*, 2–52
- REALIZE_WIDGET built-in procedure • *VAXTPU*, 7–306
- Realizing widgets in VAXTPU • *VAXTPU*, 7–306
- Real time device • *Device Support (B)*, 1–75, 1–76
- REALTIME_SPTS parameter • *Device Support (A)*, 19–9

- Real type • *Debugger*, 4–15
- REA option • *File Def Language*, FDL–11
- RECLAIMED_SPACE attribute • *File Def Language*, FDL–3
- Reclaiming buckets • *Convert*, CONV–1
- Reclamation statistics • *Convert*, CONV–24
- Record • *File Applications*, 1–1; *Analyze/RMS_File*, ARMS–6
  - See Data record
  - adding • *File Applications*, 9–10 to 9–11
  - blocking • *File Applications*, 1–8
  - compressing • *Programming Resources*, 8–26
  - contents • *File Applications*, 2–1
  - deleting • *File Applications*, 8–5, 9–20
  - determining if unmodifiable is present • *VAXTPU*, 7–175, 7–186, 7–193
  - erasing unmodifiable
    - preventing or allowing • *VAXTPU*, 7–375
  - expanding • *Programming Resources*, 8–32
  - fetching display value of • *VAXTPU*, 7–186
  - fixed-length format • *File Applications*, 1–2, 2–8, 2–9, 3–9, 3–12; *Convert*, CONV–18
  - format • *File Applications*, 2–7; *Convert*, CONV–1; *RMS*, 1–1
  - I/O • *Programming Resources*, 8–10
  - inserting • *File Applications*, 8–3 to 8–4, 9–17 to 9–19
    - VMS RMS program example • *RMS*, 4–16
  - locating • *File Applications*, 8–2 to 8–3
  - maximum length • *Convert*, CONV–26; *File Def Language*, FDL–35
  - maximum number • *File Def Language*, FDL–20
  - maximum size • *File Def Language*, FDL–35
  - merging • *Programming Resources*, 8–21
  - requirements for reading or writing in a file • *RMS*, 4–12
  - retrieving • *File Applications*, 8–2 to 8–3, 9–14 to 9–17
    - VMS RMS program example • *RMS*, 4–16
  - sensing unmodifiable erasable state • *VAXTPU*, 7–169
  - setting attribute • *VAXTPU*, 7–448
  - sorting • *Programming Resources*, 8–16
  - source line correlation • *Debugger*, 6–1
  - stream format • *File Applications*, 1–2, 3–9
  - undefined format • *File Applications*, 3–9, 3–10
  - updating • *File Applications*, 8–4, 9–19 to 9–20
  - variable format • *File Applications*, 1–2
  - variable-length format • *File Applications*, 2–9, 3–9, 3–10, 3–12
  - variable-length with fixed-length control field (VFC) format • *File Applications*, 1–2, 3–12
- Record access • *File Applications*, 9–6, 9–10
  - in stream context • *File Applications*, 8–14
  - options • *File Applications*, 7–3
- Record access block • *Routines Intro*, A–12t
  - See RAB
- Record access field
  - See RAB$B_RAC field
- Record access mode • *File Applications*, 1–2, 2–2
  - for indexed files • *File Applications*, 8–9 to 8–12
  - for relative files • *File Applications*, 8–8 to 8–9
  - for sequential files • *File Applications*, 8–7 to 8–8
  - sequential • *File Applications*, 2–2, 8–6, 8–9, 8–10
  - specifying • *File Applications*, 8–6 to 8–7, 9–10, 9–16, 9–18
- Record attribute • *VAXTPU*, F–1
- RECORD attribute • *File Def Language*, FDL–2, FDL–33
- Record attribute field
  - See FAB$B_RAT field
- Record attributes field in XABFHC
  - See XAB$B_ATR field
- Record attributes option • *File Applications*, 4–29
- Record attributes value • *I/O User's I*, 1–20
- Record buffer • *File Applications*, 9–18, 9–20
  - size • *File Applications*, 9–18, 9–20
- Record buffer field
  - See RAB$L_RBF field
- Record buffering
  - See Buffering technique
- RECORD CONTROL_FIELD_SIZE attribute • *File Def Language*, FDL–35
- Record deleting • *VAXTPU*, 6–5
- Record file address
  - See RFA
- Record file address field
  - See RAB$W_RFA field
- Record format • *File Applications*, 1–1, 1–2, 3–12; *VAXTPU*, F–1
  - fixed-length • *File Applications*, 3–19
  - selecting • *File Applications*, 2–1
  - variable-length • *File Applications*, 3–19
- Record format field
  - See FAB$B_RFM field
- Record format option • *File Applications*, 4–30
- Record header buffer • *File Applications*, 9–17, 9–18, 9–20
- Record header buffer field
  - See RAB$L_RHB field
- Record I/O
  - how to execute • *RMS*, 4–24
- Record insertion • *VAXTPU*, 6–5

Record lock block
  See RLB
Record locking • *File Applications*, 9–6
  deadlock • *File Applications*, 7–16
  use with update operation • *File Applications*, 8–3
Record locking record-processing options • *RMS*, 7–17
Record management • *Programming Resources*, 1–23
Record Management Services
  See VMS RMS
Record operation • *File Applications*, 8–1 to 8–6
Record-oriented device • *Device Support (B)*, 1–74
RECORD primary attribute
  BLOCK_SPAN secondary attribute • *File Applications*, 3–10, 4–29
  CARRIAGE_CONTROL secondary attribute • *File Applications*, 4–29
  FORMAT secondary attribute • *File Applications*, 4–30
  SIZE secondary attribute • *File Applications*, 4–29
Record processing
  VMS RMS services listed • *RMS*, 3–3
Record-processing macro
  format example • *RMS*, 3–12
Record-processing option
  for Connect service • *RMS*, 7–10
Record-processing options field
  See RAB$L_ROP field
Record processing run-time option
  deleting • *File Applications*, 9–20
  inserting • *File Applications*, 9–17 to 9–19
  retrieving • *File Applications*, 9–14 to 9–17
  updating • *File Applications*, 9–19 to 9–20
Record-processing services
  list of • *File Applications*, 8–5
Record reference vector
  See RRV
Record size field
  See RAB$W_RSZ field
Record stream
  connecting to a file • *File Applications*, 7–2
  defined • *File Applications*, 7–2
  in the context of a RAB • *RMS*, 7–1
Record stream connection option
  See File opening option
Record structure • *Analyze/RMS_File*, ARMS–6
Record transfer mode
  locate • *File Applications*, 7–16
  move • *File Applications*, 7–16
Record type • *Debugger*, 4–19
RECORD_ATTRIBUTE parameter to SET built-in procedure • *VAXTPU*, 7–448
“Record_count” string constant parameter to GET_INFO • *VAXTPU*, 7–175
RECORD_IO attribute • *File Def Language*, FDL–3
RECORD_IO secondary attribute • *File Applications*, 7–3
“Record_number” string constant parameter to GET_INFO • *VAXTPU*, 7–175
“Record_size” string constant parameter to GET_INFO • *VAXTPU*, 7–175
/RECOVER command qualifier • *VAXTPU*, 1–11, 7–307
“Recover” GET_INFO request_string • *VAXTPU*, 7–178
/RECOVER qualifier • *VAXTPU*, 5–11, 5–14
  controlling errors related to • *VAXTPU*, 7–408
Recovery
  of buffer contents • *VAXTPU*, 1–11, 7–307
  role of source file • *VAXTPU*, 7–308
  using buffer change journaling • *VAXTPU*, 7–307
  using keystroke journal file
    enabling and disabling • *VAXTPU*, 7–408
Recovery unit block
  See RUB
Recovery unit extended address block
  See XABRU block
Recovery unit file block
  See RUFB
Recovery unit stream block
  See RUSB
Recovery unit system services
  global symbols • *System Dump Analyzer*, SDA–61
Recovery unit XAB
  See XABRU block
RECOVERY_UNIT_SERVICES.EXE
  global symbols • *System Dump Analyzer*, SDA–61
RECOVER_BUFFER built-in procedure • *VAXTPU*, 7–307 to 7–309
Recurrence
  linear
    definition of • *RTL Math*, 2–7
Recursive procedure • *VAXTPU*, 3–19
Redirecting output
  DELTA • *Delta/XDelta*, DELTA–14
  XDELTA • *Delta/XDelta*, DELTA–14
REDUCE keyword
  for /DATA qualifier • *National Char Set*, NCS–26
Reentrancy • *Linker*, 4–3
  AST • *Modular Procedures*, 3–19
  full • *Modular Procedures*, 3–19

- Reentrant code • *Device Support (A),* 5–1
- .REFn directive • *MACRO,* 6–83
- Reformatting libraries
  - with /COMPRESS qualifier • *Librarian,* LIB–15
  - with /DATA qualifier • *Librarian,* LIB–20
- REFRESH built-in procedure • *VAXTPU,* 6–10, 7–310 to 7–311
  - compared with UPDATE (ALL) • *VAXTPU,* 7–538
- /REFRESH qualifier • *Debugger,* CD–73
- Register
  - See also BIIC registers
  - See also Device registers
  - See also General purpose registers
  - See also Map registers
  - See also Vector register
  - built-in symbol • *Debugger,* 4–23, D–3
  - data • *Routines Intro,* 1–6
  - depositing into • *Debugger,* 4–23
    - with DECwindows • *Debugger,* 1–26
  - display (REG) • *Debugger,* 7–10, C–5
    - with DECwindows • *Debugger,* 1–11
  - display contents • *Delta/XDelta,* DELTA–17
  - displaying • *System Dump Analyzer,* SDA–89, SDA–127
  - display kind • *Debugger,* 7–18, C–1
  - examining • *Debugger,* 4–23
    - with DECwindows • *Debugger,* 1–26
  - for returns • *Routines Intro,* 1–5, 1–15, 2–12
  - general • *System Dump Analyzer,* SDA–14
  - loading base • *Delta/XDelta,* DELTA–40
  - PC
    - See PC
  - PSL • *Debugger,* 4–24
  - saving when making call • *RMS,* 2–4
  - symbol • *Debugger,* D–3
  - symbol for base • *Delta/XDelta,* DELTA–9
  - symbol for general • *Delta/XDelta,* DELTA–13
  - symbol for processor • *Delta/XDelta,* DELTA–9
  - symbolizing • *Debugger,* 4–14, CD–271
    - with DECwindows • *Debugger,* 1–26
  - usage • *Routines Intro,* 2–12
  - variable • *Debugger,* 3–19, 4–1
    - with DECwindows • *Debugger,* 1–25
  - vector • *Routines Intro,* 2–12; *MACRO,* 10–1
    - control registers • *MACRO,* 10–2
    - internal processor registers • *MACRO,* 10–3
  - watchpoint • *Debugger,* 3–19
  - window (REG), DECwindows • *Debugger,* 1–11
- Register 0
  - See R0
- Register conflict
  - vector • *MACRO,* 10–23
- Register deferred mode • *MACRO,* 5–5
  - operand specifier format • *MACRO,* 8–19
- Register dumping routine • *Device Support (A),* 1–4, 11–10, 11–11; *Device Support (B),* 1–30, 1–83, 2–51, 3–9, 3–69, 3–82
  - address • *Device Support (A),* 6–4; *Device Support (B),* 4–15
  - context • *Device Support (B),* 4–15
  - entry point • *Device Support (B),* 4–15
  - exit method • *Device Support (B),* 4–15
  - for generic VAXBI device • *Device Support (A),* 16–22
  - functions • *Device Support (B),* 4–16
  - input • *Device Support (B),* 4–15
  - of SCSI third-party class driver • *Device Support (A),* 17–21, 17–28
  - register usage • *Device Support (B),* 4–15
  - synchronization requirements • *Device Support (B),* 4–15
- Register mask operator • *MACRO,* 3–13, 6–29
- Register mode • *MACRO,* 5–4
  - operand specifier format • *MACRO,* 8–19
- Register name • *MACRO,* 3–5, 3–6
- Register save mask • *MACRO,* 6–29, 6–59
- Register save mask directive (.MASK) • *MACRO,* 6–59
- /REGISTERS qualifier • *System Dump Analyzer,* SDA–127
- Regression testing • *Modular Procedures,* 6–1
- REI (Return from Exception or Interrupt) instruction • *MACRO,* 9–192
- REI instruction
  - role in AST delivery • *Device Support (A),* 3–4
- Reinitialization table • *Device Support (A),* 6–2, 12–8; *Device Support (B),* 1–34, 2–25
- RELALT macro • *Device Support (A),* 14–26; *Device Support (B),* 2–53, 3–84
- Related file identification field
  - See XAB$W_RFI field
- Related file identification field in XABALL
  - See XAB$W_RFI field
- Related file NAM block address field
  - See NAM$L_RLF field
- Related-file-position option • *File Applications,* 4–31
- /RELATED qualifier • *Debugger,* CD–24, CD–156, CD–231
- Relational expression • *VAXTPU,* 3–10
- Relational operators • *VAXTPU,* 2–18
- RELATIVE attribute • *File Def Language,* FDL–22
- Relative deferred mode • *MACRO,* 5–13

Relative deferred mode (Cont.)
  setting default displacement length • *MACRO*, 6–19
Relative file • *File Applications*, 2–16, 3–12
  advantages and disadvantages of using • *File Applications*, 2–18
  allocating • *File Applications*, A–1
  bucket size • *File Applications*, 3–6, 3–13, 7–19, A–1
  buffering • *File Applications*, 7–19
  buffer requirement • *RMS*, 7–6
  deferred-write option with • *File Applications*, 3–8
  defining cell size • *RMS*, 5–21
  description of relative record number • *RMS*, 7–5
  designing • *File Applications*, 3–12 to 3–15
  determining record length • *RMS*, 5–21
  establishing highest record number • *RMS*, 5–21
  examining • *File Applications*, 10–16
  maximum record size • *File Applications*, 3–12
  nonexistent record processing • *RMS*, 7–17
  omitting initial prezeroing • *RMS*, 4–23
  optimizing performance • *File Applications*, 3–12 to 3–15
  random access • *RMS*, 7–3
  record access • *File Applications*, 8–8 to 8–9, 8–12 to 8–13
  record size limit • *RMS*, 5–21
  RFA value • *RMS*, 7–9
  specifying bucket size • *RMS*, 8–5
  specifying cell size • *RMS*, 10–5
  structure • *Analyze/RMS_File*, ARMS–1, ARMS–2
  tuning • *File Applications*, 3–12 to 3–15
  with global buffers • *File Applications*, 3–14
Relative file field
  record access • *RMS*, 7–2
Relative file organization • *File Applications*, 1–2
Relative file record limit • *File Def Language*, FDL–20
Relative mode • *MACRO*, 5–12
  assembled as absolute mode • *MACRO*, 6–22
  setting default displacement length • *MACRO*, 6–19
/RELATIVE qualifier • *File Applications*, 7–19
Relative record number • *File Applications*, 1–2, 3–12
Relative volume number field
  See XAB$W_VOL field
RELCHAN macro • *Device Support (A)*, 10–2, 15–15; *Device Support (B)*, 2–54, 3–86
RELDPR macro • *Device Support (A)*, 14–25; *Device Support (B)*, 2–55, 3–87
/RELEASE qualifier • *System Dump Analyzer*, SDA–3
Release service • *File Applications*, 8–5; *RMS*, RMS–79, RMS–80
Release service (Cont.)
  condition values • *RMS*, RMS–80
  control block input and output fields • *RMS*, RMS–80
RELMPR macro • *Device Support (A)*, 14–26; *Device Support (B)*, 2–56, 3–89
Relocatable expression • *MACRO*, 3–9
/RELOCATE qualifier • *System Dump Analyzer*, SDA–59
RELSCHAN macro • *Device Support (B)*, 2–57, 3–91
Remainder • *RTL Math*, 1–7
REMAIN keyword • *VAXTPU*, 7–312
  with SEARCH • *VAXTPU*, 7–327
  with SEARCH_QUIETLY • *VAXTPU*, 7–332
Remote file access
  See also File specification
  FORTRAN program example • *File Applications*, 5–6
Remote node
  establishing logical link with • *System Services*, SYS–23
Remote terminal UCB extension • *Device Support (B)*, 1–75
Removal of key map
  built-in procedures
    REMOVE_KEY_MAP • *VAXTPU*, 7–313
Removal of window • *VAXTPU*, 2–28
/REMOVE qualifier • *Debugger*, CD–73; *Librarian*, LIB–38
Remove service • *RMS*, RMS–81, RMS–82
  caution against mixing with Search service • *RMS*, RMS–82
  comparing with Erase service • *RMS*, RMS–82
  condition values • *RMS*, RMS–84
  control block input fields • *RMS*, RMS–82
  control block output fields • *RMS*, RMS–83
  improving performance • *RMS*, RMS–82
  requirements for using • *RMS*, RMS–82
  use with wildcard characters and search lists • *RMS*, RMS–82
REMOVE_KEY_MAP built-in procedure • *VAXTPU*, 7–313 to 7–314
REMQHI (Remove Entry from Queue at Head, Interlocked) instruction • *MACRO*, 9–95
REMQTI (Remove Entry from Queue at Tail, Interlocked) instruction • *MACRO*, 9–97
REMQUE (Remove Entry from Queue) instruction • *MACRO*, 9–99
Rename service • *File Applications*, 5–9; *RMS*, RMS–85, RMS–86
  alternative to specifying arguments to $RENAME macro • *RMS*, RMS–86
  condition values • *RMS*, RMS–88

Rename service (Cont.)
control block input fields • *RMS,* RMS–86
control block output fields • *RMS,* RMS–87
exception in argument list • *RMS,* 2–5
format • *RMS,* 3–11
indicating successful completion • *RMS,* 4–16
program example • *RMS,* 4–14
requirements for using • *RMS,* RMS–86
Reorganizing a file • *Convert,* CONV–4
Repeat block
argument substitution • *MACRO,* 6–47
character substitution • *MACRO,* 6–49
end • *MACRO,* 6–28
listing range definitions of • *MACRO,* 6–89
listing range expansions of • *MACRO,* 6–89
listing specifiers • *MACRO,* 6–89
terminating repetition • *MACRO,* 6–62
Repeat block directive (.REPEAT) • *MACRO,* 6–84
REPEAT command • *Debugger,* 8–11, CD–115; *System Dump Analyzer,* SDA–64
.REPEAT directive • *MACRO,* 6–84
Repeating characters • *File Def Language,* FDL–27, FDL–28
in compression • *File Applications,* 3–16
Repeat range end directive (.ENDR) • *MACRO,* 6–28
Repetitive statements • *VAXTPU,* 3–21 to 3–22
REPLACE command • *Patch,* PAT–71
with /INSTRUCTION qualifier • *Patch,* PAT–72, PAT–73
/REPLACE qualifier • *Command Def,* CDU–43; *Librarian,* LIB–12, LIB–39; *National Char Set,* NCS–40
LIBRARY command • *Programming Resources,* 5–2
Report system event
global symbols • *System Dump Analyzer,* SDA–61
REQALT macro • *Device Support (A),* 14–10, 14–19; *Device Support (B),* 3–92
REQCOM macro • *Device Support (A),* 10–3, 17–28; *Device Support (B),* 2–59, 3–94
required for error logging • *Device Support (A),* 11–10
REQDPR macro • *Device Support (A),* 14–11, 14–17; *Device Support (B),* 2–60, 3–96
REQMPR macro • *Device Support (A),* 14–10, 14–11, 14–19; *Device Support (B),* 2–61, 3–98
REQPCHAN macro • *Device Support (A),* 3–27, 8–2 to 8–4, 15–6, 15–14; *Device Support (B),* 2–62, 3–100
REQSCHAN macro • *Device Support (A),* 15–6, 15–14; *Device Support (B),* 2–63, 3–100
Request sense key • *Device Support (A),* 17–18
Request to unwind • *Routines Intro,* 2–52
REQUIRED clause
specifying keyword in a VALUE clause • *Command Def,* CDU–29
specifying parameter in a VALUE clause • *Command Def,* CDU–24
specifying qualifier in a VALUE clause • *Command Def,* CDU–26
Required values
for /DATA qualifier • *National Char Set,* NCS–26
Reserved data type code • *Routines Intro,* 2–20
Reserved descriptor class code • *Routines Intro,* 2–44
Reserved event flag
use of • *RMS,* 2–7
Reserved operand • *MACRO,* 9–102, 9–103, 9–145
fix floating-point fault • *RTL Library,* LIB–165
Reserved word
built-in procedures • *VAXTPU,* 3–12
keywords • *VAXTPU,* 3–12
language elements • *VAXTPU,* 3–13 to 3–14
predefined constants • *VAXTPU,* 3–13
Resizing
of screen in VAXTPU • *VAXTPU,* 7–391, 7–501
Resource
controlling • *System Services Intro,* 8–6
displaying SDA information • *System Dump Analyzer,* SDA–143
lock management concept • *System Services Intro,* 12–1
name • *System Services Intro,* 12–2
of widget
fetching class and data type of • *VAXTPU,* 7–215
quota • *System Services Intro,* 2–2
supported data types for • *VAXTPU,* 4–12
RESOURCE attribute • *System Services Intro,* 3–5
Resource block
See RSB
"resources" string constant parameter to GET_INFO • *VAXTPU,* 7–215
Resource wait flag
See PCB$V_SSRWAIT
Resource wait mode • *System Services Intro,* 2–2; *Device Support (A),* 4–9; *Device Support (B),* 3–12, 3–20, 3–22
setting • *System Services,* SYS–421
Resource wait queue • *Device Support (A),* 3–25 to 3–27, E–14
See also Alternate map register wait queue
See also Data path wait queue
See also Device controller data channel wait queue

Resource wait queue (Cont.)
  See also Map register wait queue
  See also Secondary data channel wait queue
  buffered data path • *Device Support (B)*, 3–88
Response descriptor table
  See RDT
Response ID
  See RSPID
/RESPONSES qualifier • *File Def Language*, FDL–42, FDL–56
REST command • *File Applications*, 10–12, 10–16; *Analyze/RMS_File*, ARMS–33
/RESTORE qualifier • *Debugger*, CD–183
.RESTORE_PSECT directive • *MACRO*, 6–86
Restoring terminal width
  example • *VAXTPU*, A–5
Restriction • *Librarian*, LIB–11; *Analyze/RMS_File*, ARMS–11; *Convert*, CONV–5; *File Def Language*, FDL–43
  in help file keys • *Librarian*, LIB–4
  to calling services • *RMS*, 2–7
  VAXTPU
    virtual address space • *VAXTPU*, 5–1
Restrictions
  for subprocess • *VAXTPU*, 2–20
Resultant string
  requesting • *RMS*, 6–2
Resultant string area address field
  See NAM$L_RSA field
Resultant string area size field
  See NAM$B_RSS field
Resultant string length field
  See NAM$B_RSL field
RET (Return from Procedure) instruction • *MACRO*, 9–69
Retrieval pointer • *File Applications*, 9–8
Retrieval window size field
  See FAB$B_RTV field
Retrieving record
  program example • *RMS*, 4–16
Retry count • *Device Support (A)*, 10–6
Return
  I/O status • *Routines Intro*, A–7t
  object • *Routines Intro*, A–7t
Return address array • *System Services Intro*, 11–4
Return condition
  special • *System Services Intro*, 2–13
Return condition value • *System Services Intro*, 2–15
  high-level language • *System Services Intro*, 2–18
Returning condition values • *Modular Procedures*, 2–23
Returning from condition handler • *Routines Intro*, 2–52
Return key • *I/O User's I*, 8–6
  interactive mode • *File Applications*, 10–12
  logical successor • *Debugger*, 4–8, D–5
Return key command • *Delta/XDelta*, DELTA–27
/RETURN qualifier • *Debugger*, CD–133, CD–190, CD–266
Returns • *Routines Intro*, 1–14
  condition value • *Routines Intro*, 2–8
  function value • *Routines Intro*, 2–7
  in I/O status block • *Routines Intro*, 1–14
  in mailbox • *Routines Intro*, 1–14
  signaled in register • *Routines Intro*, 1–15
Returns heading • *Routines Intro*, 1–5
RETURN statement • *VAXTPU*, 3–26, 3–31 to 3–33, 7–315
Return status • *Programming Resources*, 9–3
  from signal • *Programming Resources*, 9–6
REVERSE keyword • *VAXTPU*, 7–85, 7–453
  with MARK • *VAXTPU*, 7–261
  with SEARCH • *VAXTPU*, 7–328
  with SEARCH_QUIETLY • *VAXTPU*, 7–333
  with SELECT • *VAXTPU*, 7–337
  with SET (MESSAGE_ACTION_TYPE) • *VAXTPU*, 7–426
  with SET (PROMPT_AREA) • *VAXTPU*, 7–446
  with SET (STATUS_LINE) • *VAXTPU*, 7–476
  with SET (VIDEO) • *VAXTPU*, 7–492
"Reverse_status" string constant parameter to GET_INFO • *VAXTPU*, 7–224
"Reverse_video" string constant parameter to GET_INFO • *VAXTPU*, 7–224
Revert to the caller's handling • *Routines Intro*, 2–47
REVISION attribute • *File Def Language*, FDL–16, FDL–24
Revision data • *File Applications*, 9–10
Revision date and time extended address block
  See XABRDT block
Revision date and time field
  See XAB$Q_RDT field
Revision number • *File Def Language*, FDL–24
Revision number field
  See XAB$W_RVN field
REVISION secondary attribute • *File Applications*, 4–28
Rewind offline function • *I/O User's I*, 6–21
Rewind on close option
  See FAB$V_RWC option
Rewind on open option
  See FAB$V_RWO option

Rewind service • *File Applications*, 8–5; *RMS*, RMS–89, RMS–90

  condition values • *RMS*, RMS–90

  control block input fields • *RMS*; RMS–90

  control block output fields • *RMS*, RMS–90

  effect on next-record position • *File Applications*, 8–16

  use restriction • *RMS*, RMS–90

RF30 disk • *I/O User's I*, 3–7

RF71 disk • *I/O User's I*, 3–7

RFA (record file address) • *File Applications*, 1–2, 8–12 to 8–13, 9–17, 10–31; *Convert*, CONV–1, CONV–4

  access • *File Applications*, 10–30; *Convert*, CONV–4

  created by CONVERT • *File Applications*, 3–16

  use of table for rapid access • *File Applications*, 8–3

/RIGHT qualifier • *Debugger*, CD–98, CD–110, CD–118

Rights database • *Programming Resources*, 6–1; *System Services Intro*, 3–2, 3–5, 3–15

  adding to • *System Services Intro*, 3–9

  default protection • *System Services Intro*, 3–6

  elements of • *System Services Intro*, 3–7

  holder record • *System Services Intro*, 3–5

  identifier record • *System Services Intro*, 3–5

  initializing • *System Services Intro*, 3–6

  keys • *System Services Intro*, 3–6

  modifying • *System Services Intro*, 3–12, 3–13, 3–15

Rights identifier • *Routines Intro*, A–12t

Rights list • *System Services Intro*, 3–30

rights_holder data type • *Routines Intro*, A–11t

rights_id data type • *Routines Intro*, A–12t

RIGHT_MARGIN keyword • *VAXTPU*, 7–454

"Right_margin" string constant parameter to GET_ INFO • *VAXTPU*, 7–175, 7–186

RIGHT_MARGIN_ACTION keyword • *VAXTPU*, 7–456

"Right_margin_action" string constant parameter to GET_INFO • *VAXTPU*, 7–175

RK06 cartridge disk • *I/O User's I*, 3–7

RK07 cartridge disk • *I/O User's I*, 3–7

RL01 driver • *Device Support (A)*, C–1 to C–29

RL02 driver • *Device Support (A)*, C–1 to C–29

RL11 driver • *Device Support (A)*, C–1 to C–29

RLB (record lock block) • *System Dump Analyzer*, SDA–77

RLK option • *File Def Language*, FDL–11

RM03 device • *File Def Language*, FDL–38

RM03 disk • *I/O User's I*, 3–7

RM05 disk • *I/O User's I*, 3–7

RMS$_OK_LIM success status code • *RMS*, 7–13

RMS (Record Management Services)

  See VMS RMS

RMS.EXE • *System Dump Analyzer*, SDA–61

RMS–11

  block identifier field limitation • *RMS*, 5–3

  stream files • *File Def Language*, FDL–35

  Version 1.8 • *File Def Language*, FDL–30

RMS control blocks

  with FDL routines • *Utility Routines*, FDL–14, FDL–17

RMSDEF.STB • *System Dump Analyzer*, SDA–60

$RMSDEF macro

  See also VMS RMS

  access to symbolic offset names • *RMS*, 2–2

RMS image

  base address • *System Dump Analyzer*, SDA–14

/RMS qualifier • *System Dump Analyzer*, SDA–127

RMS structures • *Programming Resources*, 8–58

RMS symbol • *System Dump Analyzer*, SDA–14

RMS utilities

  See VMS RMS

RMS_DEFAULT command • *File Def Language*, FDL–30

RMS_DFNBC system parameter

  for specifying default network block count • *RMS*, 5–22

RMS_GBLBUFQUO system parameter • *File Applications*, 1–16

RNE option • *File Def Language*, FDL–14

RNF option • *File Def Language*, FDL–14

Rn symbol • *Delta/XDelta*, DELTA–9

Rooted-device logical name • *File Applications*, 6–15

Rooted-directory logical name

  for additional nesting • *File Applications*, 6–18

Rooted-directory specification

  concatenated • *File Applications*, 6–17 to 6–19

  syntax • *File Applications*, 6–15 to 6–20

Root index bucket virtual block field

  See XAB$L_RVB field

Root level • *File Applications*, 3–17

Rotation

  applying to a vector • *RTL Math*, MTH–173

Rotational latency • *File Applications*, 1–5

ROTL (Rotate Long) instruction • *MACRO*, 9–28

Routine

  See also DECtalk routine

  See also Entry point

  See also Mathematics routine

  See also String manipulation routine

Routine (Cont.)
- calling • *Debugger,* 8–11, 11–23, CD–10
- calling from a program • *Convert,* CONV–1
- call stack • *Debugger,* 2–14, 7–6, 7–10, CD–170, CD–214
  - with DECwindows • *Debugger,* 1–21, 1–22, 1–24, 1–28
- definition of • *RTL Intro,* 1–1
- displaying instructions for, on call stack • *Debugger,* 7–10, CD–170
  - with DECwindows • *Debugger,* 1–22
- displaying source code for, on call stack • *Debugger,* 7–6, CD–170
  - with DECwindows • *Debugger,* 1–21
- EXAMINE/SOURCE command • *Debugger,* 6–4
- how to call • *RTL Intro,* 1–19, 3–1, 3–2
- library • *File Def Language,* FDL–41, FDL–42
- multiple invocations of • *Debugger,* 5–10, CD–170
  - with DECwindows • *Debugger,* 1–28
- processwide resource allocation • *RTL Library,* 2–16, 2–17
- selecting from DECwindows window • *Debugger,* 1–23
- SET BREAK command • *Debugger,* 3–11
- SET SCOPE command • *Debugger,* CD–170
- SET TRACE command • *Debugger,* 3–11
- SHOW CALLS command • *Debugger,* 2–14
- traceback information • *Debugger,* 5–3
  - with DECwindows • *Debugger,* 1–24
- variable-length bit field • *RTL Library,* 2–10

ROUTINE clause
- for DEFINE SYNTAX statement • *Command Def,* CDU–26
- for DEFINE VERB statement • *Command Def,* CDU–35

Routine name
- made available to debugger • *MACRO,* 6–23

Routine name heading • *Routines Intro,* 1–1

Routine overview heading • *Routines Intro,* 1–1

Routines • *Librarian,* LIB–10

RP05 disk • *I/O User's I,* 3–7

RP06 device • *File Def Language,* FDL–38

RP06 disk • *I/O User's I,* 3–7

RP07 disk • *I/O User's I,* 3–7

RPG II
- See VAX RPG II

RPG II data type declaration • *Routines Intro,* A–48

RPG II implementation table • *Routines Intro,* A–48

RQDX3 disk controller • *I/O User's I,* 3–5

RRL option • *File Def Language,* FDL–13

RRV (record reference vector) • *File Applications,* 3–6, 3–22; *Analyze/RMS_File,* ARMS–6

RSB (resource block) • *System Dump Analyzer,* SDA–109, SDA–143

RSB (Return from Subroutine) instruction • *MACRO,* 9–60

RSB instruction • *Device Support (A),* 7–4

RSPID (response ID)
- displaying SDA information • *System Dump Analyzer,* SDA–148

RST (run-time symbol table) • *Debugger,* 5–6
- and symbol search • *Debugger,* 5–8
- deleting symbol records in • *Debugger,* 5–7, CD–24
- displaying modules in • *Debugger,* 5–7, CD–231
- displaying symbols in • *Debugger,* 5–9, CD–250
- inserting symbol records in • *Debugger,* 5–6, CD–156
- shareable image • *Debugger,* 5–14
- with DECwindows • *Debugger,* 1–27

RSTS/E • *File Def Language,* FDL–38

RSX–11M • *File Def Language,* FDL–38

RSX–11M–PLUS • *File Def Language,* FDL–38

RSX–11M/M–PLUS
- differences from VMS • *I/O User's I,* 4–35

RT–11 • *File Def Language,* FDL–38

RTL (Run-Time Library)
- capabilities of • *RTL Intro,* 1–1
- condition handling • *RTL Library,* 4–1
- described • *RTL Intro,* 1–1
- organization of • *RTL Intro,* 1–19
- queue access • *RTL Library,* 2–12

RTPAD • *I/O User's I,* 8–11

RUB (recovery unit block) • *System Dump Analyzer,* SDA–77

RUFB (recovery unit file block) • *System Dump Analyzer,* SDA–77

Rules
- for FDL validity • *File Def Language,* FDL–39

RUN command • *Debugger,* 3–1, 3–3, 5–4; *Linker,* 2–5
- See also Execution
- shareable image • *Debugger,* 5–14
- with DECwindows • *Debugger,* 1–4

Running VAXTPU from subprocess
- example • *VAXTPU,* A–5

RUN processor state • *Device Support (B),* 1–16

Run-time
- access options • *RMS,* 1–2
- access options under VMS RMS • *RMS,* 1–2
- implementation of services • *RMS,* 4–1
- implementation of VMS RMS services • *RMS,* 4–1
- information • *RMS,* 1–4
- information to VMS RMS listed • *RMS,* 1–4

Run-time (Cont.)
processing environment • *RMS,* 2–1
Run-Time Library
See RTL
Run-Time Library procedures • *Modular Procedures,* 1–6
Run-Time Library routine • *Programming Resources,* 1–24 to 1–29
capabilities of • *RTL Intro,* 1–18
DECtalk • *RTL DECtalk,* 1–1
defined • *RTL Intro,* 1–1
entry point • *RTL Intro,* 3–3, 3–4, 3–5
general purpose • *RTL General Purpose,* 1–1
how to call • *RTL Intro,* 1–19, 3–1, 3–2
integer and floating-point • *RTL Library,* 2–12
interaction with operating system • *RTL Library,* 2–1
jacket routine • *RTL Library,* 2–1
library • *RTL Library,* 1–1
linking with • *RTL Intro,* 1–19
output formatting control • *RTL Library,* 2–20
performance measurement • *RTL Library,* 2–18
return status • *Programming Resources,* 9–3
string manipulation • *RTL String Manipulation,* 2–1
system service access • *RTL Library,* 2–1
to access command language interpreter • *RTL Library,* 2–2
to access VAX instruction set • *RTL Library,* 2–9
to access VMS system components • *RTL Library,* 2–1
to manipulate character string • *RTL Library,* 2–14
variable-length bit field instruction • *RTL Library,* 2–10
Run-time option
example • *File Applications,* 9–20 to 9–22
specifying • *File Applications,* 9–1 to 9–6
Run-time symbol table
See RST
RUSB (recovery unit stream block) • *System Dump Analyzer,* SDA–77
/RU_JOURNAL qualifier
description • *Analyze/RMS_File,* ARMS–18
format • *Analyze/RMS_File,* ARMS–18
overview • *Analyze/RMS_File,* ARMS–18
using with /OUTPUT qualifier • *Analyze/RMS_File,* ARMS–16
RWC option • *File Def Language,* FDL–21
RWO option • *File Def Language,* FDL–22
RX01 console disk • *I/O User's I,* 3–8
RX02 Diskette • *I/O User's I,* 3–8
RX23 diskette • *I/O User's I,* 3–9
RX33 diskette • *I/O User's I,* 3–10
RX50 diskette • *I/O User's I,* 3–10
RX-series • *I/O User's I,* 3–9
RZ22 disk • *I/O User's I,* 3–10
RZ23 disk • *I/O User's I,* 3–10
RZ55 disk • *I/O User's I,* 3–10

S

S0 region
examining • *System Dump Analyzer,* SDA–52
"safe_for_journaling" string constant parameter
GET_INFO built-in • *VAXTPU,* 7–175
Sample procedures using DECwindows VAXTPU built-in procedures • *VAXTPU,* B–1 to B–33
Sample program • *System Services Intro,* 13–1
invoked by user-defined command • *Command Def,* CDU–45
to parse and execute commands • *Command Def,* CDU–46
Sample VAXTPU procedures
debugon • *VAXTPU,* 7–365
delete_all_definitions • *VAXTPU,* 7–533
init_help_key_map_list • *VAXTPU,* 7–66
init_sample_key_map • *VAXTPU,* 7–64
line_number_example • *VAXTPU,* 7–417
mail_sub • *VAXTPU,* 7–343
my_call_user • *VAXTPU,* 7–43
remove_comments • *VAXTPU,* 7–312
SAVE • *VAXTPU,* 7–318
shift_key_handler • *VAXTPU,* 7–257
show_key_maps_in_list • *VAXTPU,* 7–161
show_key_map_lists • *VAXTPU,* 7–160
show_self_insert • *VAXTPU,* 7–161
strip_blanks • *VAXTPU,* 7–124, 7–126, 7–128
strip_eight • *VAXTPU,* 7–528
toggle_self_insert • *VAXTPU,* 7–471
traceback_example • *VAXTPU,* 7–489
user_change_mode • *VAXTPU,* 7–103
user_change_windows • *VAXTPU,* 7–290
user_clear_key • *VAXTPU,* 7–533
user_collect_rnos • *VAXTPU,* 7–145
user_dcl_process • *VAXTPU,* 7–68
user_define_edtkey • *VAXTPU,* 7–240
user_define_key • *VAXTPU,* 7–103
user_delete • *VAXTPU,* 7–89
user_delete_char • *VAXTPU,* 7–29
user_delete_extra • *VAXTPU,* 7–109
user_delete_key • *VAXTPU,* 7–120
user_display_current_character • *VAXTPU,* 7–82

Sample VAXTPU procedures (Cont.)
user_display_help • *VAXTPU*, 7–23
user_display_key_map_list • *VAXTPU*, 7–160
user_display_position • *VAXTPU*, 7–522
user_do • *VAXTPU*, 7–131
user_double_parens • *VAXTPU*, 7–265
user_edit_string • *VAXTPU*, 7–114
user_emphasize_message • *VAXTPU*, 7–509
user_end_of_line • *VAXTPU*, 7–251
user_erase_message_buffer • *VAXTPU*, 7–315
user_erase_to_eob • *VAXTPU*, 7–71
user_error_messsage • *VAXTPU*, 7–139
user_fao_conversion • *VAXTPU*, 7–139
user_find_chap • *VAXTPU*, 7–330, 7–335
user_find_mark_twain • *VAXTPU*, 7–514
user_find_parens • *VAXTPU*, 7–320
user_find_procedure • *VAXTPU*, 7–27
user_find_string • *VAXTPU*, 7–315
user_free-cursor_up • *VAXTPU*, 7–98
user_free_cursor_down • *VAXTPU*, 7–98
user_free_cursor_left • *VAXTPU*, 7–95
user_free_cursor_right • *VAXTPU*, 7–95
user_get_info • *VAXTPU*, 7–160
user_get_key_info • *VAXTPU*, 7–256
user_go_down • *VAXTPU*, 7–91
user_go_up • *VAXTPU*, 7–91
user_help • *VAXTPU*, 7–229
user_help_buffer • *VAXTPU*, 7–62
user_help_on_key • *VAXTPU*, 7–302
user_include_file • *VAXTPU*, 7–38
user_initial_cap • *VAXTPU*, 7–524
user_is_character • *VAXTPU*, 7–231
user_lowercase_line • *VAXTPU*, 7–46
user_make_window • *VAXTPU*, 7–79
user_mark • *VAXTPU*, 7–248
user_message_window • *VAXTPU*, 7–260
user_move_8_lines • *VAXTPU*, 7–283
user_move_by_lines • *VAXTPU*, 7–279
user_move_text • *VAXTPU*, 7–281
user_move_to_mouse • *VAXTPU*, 7–253
user_next_page • *VAXTPU*, 7–286
user_next_screen • *VAXTPU*, 7–93
user_not_quite_working • *VAXTPU*, 7–39
user_one_window_to_two • *VAXTPU*, 7–537
user_on_eol • *VAXTPU*, 7–269
user_paste • *VAXTPU*, 7–116, 7–263
user_print • *VAXTPU*, 7–485
user_prompt_number • *VAXTPU*, 7–233, 7–305
user_quick_parse • *VAXTPU*, 7–137
user_quit • *VAXTPU*, 7–292
user_quote • *VAXTPU*, 7–294

Sample VAXTPU procedures (Cont.)
user_remove_blank_lines • *VAXTPU*, 7–514
user_remove_comments • *VAXTPU*, 7–25
user_remove_crlfs • *VAXTPU*, 7–118
user_remove_dsrlines • *VAXTPU*, 7–250
user_remove_non_numbers • *VAXTPU*, 7–323
user_remove_numbers • *VAXTPU*, 7–514
user_remove_odd_characters • *VAXTPU*, 7–321
user_remove_paren_text • *VAXTPU*, 7–531
user_repaint • *VAXTPU*, 7–311
user_replace_prefix • *VAXTPU*, 7–31
user_ring_bell • *VAXTPU*, 7–356
user_runoff_line • *VAXTPU*, 7–87
user_scroll_buffer • *VAXTPU*, 7–326
user_search_for_nonalpha • *VAXTPU*, 7–285
user_search_range • *VAXTPU*, 7–331, 7–336
user_select • *VAXTPU*, 7–341
user_show_direction • *VAXTPU*, 7–85
user_show_first_line • *VAXTPU*, 7–539
user_simple_insert • *VAXTPU*, 7–54
user_slow_down_arrow • *VAXTPU*, 7–354
user_slow_up_arrow • *VAXTPU*, 7–354
user_split_line • *VAXTPU*, 7–84, 7–519
user_start_journal • *VAXTPU*, 7–142
user_start_select • *VAXTPU*, 7–339
user_tab • *VAXTPU*, 7–33
user_test_key • *VAXTPU*, 7–34
user_toggle_direction • *VAXTPU*, 7–80
user_top • *VAXTPU*, 7–38
user_tpu • *VAXTPU*, 7–132
user_trans_text • *VAXTPU*, 7–528
user_two_window • *VAXTPU*, 7–298
user_upcase_item • *VAXTPU*, 7–46
user_what_is_comment • *VAXTPU*, 7–256
user_write_file • *VAXTPU*, 7–545
SAVE built-in procedure • *VAXTPU*, 7–316 to 7–318
SAVE command • *Debugger*, 7–22, CD–116
SAVEDUMP parameter • *System Dump Analyzer*, SDA–3, SDA–28
Save set (BACKUP) • *File Applications*, 10–31
.SAVE_PSECT directive • *MACRO*, 6–87
/SAVE_VECTOR_STATE qualifier • *Debugger*, 11–23, CD–11
SAVIPL macro • *Device Support (A)*, 3–10; *Device Support (B)*, 2–64
SB (system block) • *System Dump Analyzer*, SDA–83, SDA–99
SBI (synchronous backplane interconnect) • *Device Support (A)*, 1–11
UNIBUS interlock sequence to • *Device Support (A)*, 14–10
SBICONF array • *Device Support (A)*, 16–8

SBR register
  displaying • *System Dump Analyzer*, SDA–90
SBWC (Subtract with Carry) instruction • *MACRO*, 9–29
SBZ field • *MACRO*, 7–2
SCA (Source Code Analyzer) • *Modular Procedures*, 1–13
Scalar
  processor synchronization • *Routines Intro*, 2–13
Scalar type • *Debugger*, 4–15
Scalar/vector memory synchronization • *MACRO*, 10–38
Scaling
  vector • *RTL Math*, MTH–183
SCAN
  See VAX SCAN
SCAN built-in procedure • *VAXTPU*, 7–319 to 7–321
SCANC (Scan Characters) instruction • *MACRO*, 9–138
  RTL routine to access • *RTL Library*, LIB–334
SCAN data type declaration • *Routines Intro*, A–51
SCAN implementation table • *Routines Intro*, A–51
SCANL built-in procedure • *VAXTPU*, 7–322 to 7–323
Scatter-gather map • *Device Support (A)*, 14–4
  See also Map registers
SCB (system control block) • *Device Support (A)*, 16–10; *Device Support (B)*, 1–7
SCBB register
  displaying • *System Dump Analyzer*, SDA–90
SCDRP$L_ABCNT • *Device Support (A)*, 17–15
SCDRP$L_BCNT • *Device Support (A)*, 17–15, 17–19; *Device Support (B)*, 2–78, 2–85
SCDRP$L_CMD_PTR • *Device Support (A)*, 17–11; *Device Support (B)*, 2–85
SCDRP$L_DISCON_TIMEOUT • *Device Support (A)*, 17–11, 17–12
SCDRP$L_DMA_TIMEOUT • *Device Support (A)*, 17–11, 17–12
SCDRP$L_IRP • *Device Support (A)*, 17–27
SCDRP$L_MEDIA • *Device Support (A)*, 17–15
SCDRP$L_PAD_COUNT • *Device Support (A)*, 17–15
SCDRP$L_SCSI_FLAGS • *Device Support (A)*, 17–15, 17–16, 17–27; *Device Support (B)*, 2–78
SCDRP$L_SPTE_SVAPTE • *Device Support (A)*, 17–16
SCDRP$L_STS_PTR • *Device Support (A)*, 17–11, 17–18; *Device Support (B)*, 2–85, 2–86
SCDRP$L_SVAPTE • *Device Support (A)*, 17–15; *Device Support (B)*, 2–78
SCDRP$L_SVA_SPTE • *Device Support (B)*, 2–79
SCDRP$L_SVA_USER • *Device Support (A)*, 17–15, 17–16; *Device Support (B)*, 2–78, 2–79, 2–85
SCDRP$L_TRANS_CNT • *Device Support (A)*, 17–19; *Device Support (B)*, 2–86
SCDRP$V_BUFFER_MAPPED • *Device Support (A)*, 17–16, 17–27
SCDRP$V_S0BUF • *Device Support (A)*, 17–16, 17–27
SCDRP$W_BOFF • *Device Support (A)*, 17–15; *Device Support (B)*, 2–78
SCDRP$W_FUNC • *Device Support (A)*, 17–15; *Device Support (B)*, 2–85
SCDRP$W_MAPREG • *Device Support (A)*, 17–17; *Device Support (B)*, 2–79
SCDRP$W_NUMREG • *Device Support (A)*, 17–16; *Device Support (B)*, 2–79
SCDRP$W_PAD_BCNT • *Device Support (B)*, 2–85
SCDRP$W_STS • *Device Support (A)*, 17–15, 17–16; *Device Support (B)*, 2–78
SCDRP (SCSI class driver request packet) • *Device Support (A)*, 17–7; *Device Support (B)*, 1–46 to 1–54
  allocating • *Device Support (A)*, 17–27
  deallocating • *Device Support (A)*, 17–28
  defining fields of • *Device Support (A)*, 17–24
  initializing • *Device Support (A)*, 17–15 to 17–16, 17–27
$SCDRPDEF macro • *Device Support (A)*, 17–24
SCDT (SCSI connection descriptor table) • *Device Support (A)*, 17–7; *Device Support (B)*, 1–54 to 1–60
SCF option • *File Def Language*, FDL–24
SCH$GL_CURPCB • *Delta/XDelta*, DELTA–9
  replaced in VMS Version 5.0 • *Device Support (A)*, E–6
SCH$GL_PCBVEC • *Delta/XDelta*, DELTA–9; *Device Support (A)*, 13–13
SCH$POSTEF • *Device Support (B)*, 1–39
SCH$QAST • *Device Support (A)*, 3–4
SCH$RESCHED • *Device Support (A)*, 3–7
SCHED spin lock • *Device Support (A)*, 3–4, 3–8, 3–14; *Device Support (B)*, 3–19
Scheduler
  blocking activity of • *Device Support (A)*, 3–5
  global symbols • *System Dump Analyzer*, SDA–61
  synchronization of • *Device Support (A)*, 3–7
S command • *Delta/XDelta*, DELTA–34
Scope
  built-in symbol • *Debugger*, 7–4, 7–8, 7–17, 7–19, C–4, C–5, C–6, D–11
  canceling • *Debugger*, 5–12, CD–27; *Patch*, PAT–44
  current • *Debugger*, 5–11, CD–170

Scope (Cont.)
- default • *Debugger,* 5–8, CD–27, CD–171, CD–241
  - with DECwindows • *Debugger,* 1–28
- displaying • *Debugger,* 5–12, CD–241
- displaying current setting • *Patch,* PAT–88
- for instruction display • *Debugger,* 7–10, CD–170
  - with DECwindows • *Debugger,* 1–8, 1–22
- for source display • *Debugger,* 7–6, CD–170
  - with DECwindows • *Debugger,* 1–8, 1–21
- for symbol search • *Debugger,* 5–8, 5–11, CD–27, CD–170, CD–241
  - with DECwindows • *Debugger,* 1–8, 1–28
- PC • *Debugger,* 5–8
- relation to call stack • *Debugger,* 5–10, 5–11, 7–6, 7–10, CD–170
  - with DECwindows • *Debugger,* 1–8, 1–21, 1–22, 1–28
- SEARCH command • *Debugger,* 6–6, CD–120
- search list • *Debugger,* 5–8, 5–11, CD–27, CD–170, CD–241
  - with DECwindows • *Debugger,* 1–8, 1–28
- SET SCOPE command • *Debugger,* 5–11, 7–6, 7–10, CD–170
- setting • *Debugger,* 5–11, CD–170; *Patch,* PAT–84
  - with DECwindows • *Debugger,* 1–28
- specifying with path name • *Debugger,* 5–9
- TYPE command • *Debugger,* 6–4, CD–275
- vector register • *Debugger,* 11–1

/SCOPE-/NOSCOPE qualifier
- with DELETE command • *Patch,* PAT–53
- with DEPOSIT command • *Patch,* PAT–56
- with EXAMINE command • *Patch,* PAT–63
- with INSERT command • *Patch,* PAT–69
- with REPLACE command • *Patch,* PAT–72
- with SET MODE command • *Patch,* PAT–77
- with VERIFY command • *Patch,* PAT–91

SCOPE-NOSCOPE mode • *Patch,* PAT–17

Scratch file • *Convert,* CONV–11

Screen
- enabling resizing of • *VAXTPU,* 7–372
- minimal update • *RTL Screen Management,* 2–17
- resizing • *VAXTPU,* 7–391, 7–501
- specifying size of • *VAXTPU,* 7–458
- updating
  - controlling support for • *VAXTPU,* 7–460

Screen display
- See Display, screen mode

SCREEN keyword
- using with widget-related built-in procedures • *VAXTPU,* 4–16

Screen layout
- built-in procedures
  - ADJUST_WINDOW • *VAXTPU,* 7–19
  - CREATE_WINDOW • *VAXTPU,* 7–77
  - MAP • *VAXTPU,* 7–259
  - REFRESH • *VAXTPU,* 7–310
  - SHIFT • *VAXTPU,* 7–503
  - UNMAP • *VAXTPU,* 7–536
  - UPDATE • *VAXTPU,* 7–538

Screen management • *Programming Resources,* 7–7; *RTL Screen Management,* 1–1
- See also Key table
- See also Pasteboard
- See also Video attribute
- See also Viewport
- See also Virtual display
- See also Virtual keyboard
- debugging DECwindows application • *Debugger,* 1–36
- debugging screen-oriented program • *Debugger,* 9–5
- deleting text • *Programming Resources,* 7–21
- double-width characters • *Programming Resources,* 7–19, 7–20
- drawing lines • *Programming Resources,* 7–20
- inserting characters • *Programming Resources,* 7–18
- menus
  - creating • *Programming Resources,* 7–22
  - reading • *Programming Resources,* 7–23
  - types of • *Programming Resources,* 7–22
- reading data • *Programming Resources,* 7–23
- scrolling • *Programming Resources,* 7–20
- setting background color • *Programming Resources,* 7–9
- setting screen dimensions • *Programming Resources,* 7–9
- using system routines • *Programming Resources,* 1–23
- video attributes • *Programming Resources,* 7–20
- viewport • *Programming Resources,* 7–17

Screen management resources • *Modular Procedures,* 2–17

Screen manager • *VAXTPU,* 2–28, 6–1 to 6–12
- automatic update • *VAXTPU,* 6–7
- line changes • *VAXTPU,* 6–6
- partial update • *VAXTPU,* 6–8
- specific window update • *VAXTPU,* 6–8
- suppressing updates • *VAXTPU,* 6–6
- update all windows • *VAXTPU,* 6–9
- update order • *VAXTPU,* 6–7

Screen manager (Cont.)
  updates • *VAXTPU,* 6–6
  update with ADJUST_WINDOW • *VAXTPU,* 7–22
  update with CURSOR_HORIZONTAL • *VAXTPU,* 7–94
  update with CURSOR_VERTICAL • *VAXTPU,* 7–97
Screen mode • *Debugger,* 7–1, CD–154
  multiprocess program • *Debugger,* 10–16
  summary reference information • *Debugger,* C–1
Screen object
  in VAXTPU • *VAXTPU,* 4–14
Screen-oriented program
  debugging • *Debugger,* 9–5
    with DECwindows • *Debugger,* 1–36
Screen size
  displaying • *Debugger,* 7–24, CD–256
  %PAGE, %WIDTH symbols • *Debugger,* C–6
  setting • *Debugger,* 7–23, CD–185
Screen update
  See Screen manager
/SCREEN_LAYOUT qualifier • *Debugger,* CD–101
SCREEN_UPDATE keyword • *VAXTPU,* 7–460
"Screen_update" string constant parameter to GET_ INFO • *VAXTPU,* 7–201
Script
  EDIT/FDL • *File Def Language,* FDL–63
  list of • *File Applications,* 4–4
  optimize • *File Applications,* 10–1
  touch-up • *File Applications,* 10–28
/SCRIPT=OPTIMIZE qualifier • *File Applications,* 10–29
/SCRIPT qualifier • *File Applications,* 10–28; *File Def Language,* FDL–42, FDL–57
Scroll
  backward • *Programming Resources,* 7–19
  down • *Programming Resources,* 7–19
  forward • *Programming Resources,* 7–19
  output • *Programming Resources,* 7–19
  up • *Programming Resources,* 7–19
Scroll bar
  disabling • *VAXTPU,* 7–462
  enabling • *VAXTPU,* 7–462
Scroll bar slider
  adjusting automatically • *VAXTPU,* 7–224
Scroll bar widget
  example of fetching • *VAXTPU,* B–19 to B–22
SCROLL built-in procedure • *VAXTPU,* 6–10, 7–324 to 7–326
SCROLL command • *Debugger,* 7–11, CD–118
Scrolling
  effect of on cursor position • *VAXTPU,* 7–324

Scrolling (Cont.)
  effect of on editing point • *VAXTPU,* 7–324
  with records deleted • *VAXTPU,* 6–5
  with records inserted • *VAXTPU,* 6–5
SCROLLING keyword • *VAXTPU,* 7–467
Scroll mode • *Debugger,* CD–154
  jump • *RTL Screen Management,* SMG–347
  smooth • *RTL Screen Management,* SMG–347
/SCROLL qualifier • *Debugger,* 7–21, CD–124
"Scroll" string constant parameter to GET_INFO • *VAXTPU,* 7–201, 7–224
"Scroll_amount" string constant parameter to GET_ INFO • *VAXTPU,* 7–224
"Scroll_bottom" string constant parameter to GET_ INFO • *VAXTPU,* 7–224
"Scroll_top" string constant parameter to GET_ INFO • *VAXTPU,* 7–225
SCS (system communications services) • *Device Support (B),* 1–33
  base address • *System Dump Analyzer,* SDA–14
  displaying SDA information • *System Dump Analyzer,* SDA–82, SDA–83, SDA–87, SDA–123, SDA–148
  global symbols • *System Dump Analyzer,* SDA–60
SCSDEF.STB • *System Dump Analyzer,* SDA–60
SCSI (Small Computer System Interface)
  definition • *Device Support (A),* 17–1
  hardware considerations • *Device Support (A),* 1–18
SCSI bus
  releasing in AEN operation • *Device Support (B),* 2–81
  resetting • *Device Support (B),* 2–82
  sensing phase of • *Device Support (B),* 2–87
  setting phase of • *Device Support (B),* 2–90
  VAX systems concepts • *Device Support (A),* 17–1
SCSI bus analyzer • *Device Support (A),* 17–32
SCSI class driver • *I/O User's I,* 11–2
  See also Class driver
  See also Disk class driver
  See also Generic SCSI class driver
  See also Tape class driver
  See also Template class driver
  See also Third-party SCSI class driver
SCSI class driver request packet
  See SCDRP
SCSI class/port architecture • *I/O User's I,* 11–2; *Device Support (A),* 17–2 to 17–5
  summary of I/O request servicing • *Device Support (A),* 17–22 to 17–24

SCSI command
- controlling the number of retries • *Device Support (A),* 17–13
- determining timeout setting for • *Device Support (B),* 2–76
- disabling retry • *I/O User's I,* 11–8; *Device Support (A),* 17–12; *Device Support (B),* 2–75, 2–88
- enabling retry • *I/O User's I,* 11–13; *Device Support (B),* 2–75
- examining status of • *Device Support (A),* 17–17 to 17–19, 17–27
- padding, when required • *I/O User's I,* 11–14
- preparing to issue • *Device Support (A),* 17–10 to 17–13
- sending to SCSI device • *Device Support (A),* 17–11; *Device Support (B),* 2–84 to 2–86
- setting disconnect timeout for • *I/O User's I,* 11–8, 11–14; *Device Support (A),* 17–11, 17–12; *Device Support (B),* 2–76, 2–89
- setting DMA timeout for • *I/O User's I,* 11–8, 11–14; *Device Support (A),* 17–11, 17–12; *Device Support (B),* 2–76, 2–89
- setting phase change timeout for • *I/O User's I,* 11–8, 11–14; *Device Support (A),* 17–11, 17–12; *Device Support (B),* 2–76, 2–89
- size of • *Device Support (A),* 17–11
- terminating • *Device Support (A),* 17–28; *Device Support (B),* 2–68

SCSI command byte
- buffering • *Device Support (A),* 17–11, 17–27; *Device Support (B),* 2–69

SCSI command descriptor block
- creating • *Device Support (A),* 17–11
- initializing pointer to • *Device Support (A),* 17–11

SCSI connection descriptor table
- See SCDT

SCSI controller
- NCR 5380 • *Device Support (A),* 1–18
- SII • *Device Support (A),* 1–19

SCSI device
- connecting to • *Device Support (A),* 17–9

SCSI device ID • *Device Support (A),* 17–2

SCSI device UCB • *Device Support (A),* 17–8
- extending • *Device Support (A),* 17–24

SCSI disconnect feature
- enabling • *I/O User's I,* 11–7

SCSI disk
- class driver • *I/O User's I,* 3–22
- error recovery • *I/O User's I,* 3–17, 3–22

SCSI ID • *Device Support (A),* 17–2

SCSI port descriptor table
- See SPDT

SCSI port driver • *I/O User's I,* 11–2
- See Port driver

SCSI port ID • *Device Support (A),* 17–1

SCSI port interface
- See SPI

SCSI port UCB • *Device Support (A),* 17–8

SCSI status byte
- examining • *Device Support (A),* 17–18
- initializing • *Device Support (A),* 17–11
- servicing CHECK CONDITION status • *Device Support (A),* 17–18

SCSI_NOAUTO system parameter • *I/O User's I,* 11–10; *Device Support (A),* 17–31

SCSLOA symbol • *System Dump Analyzer,* SDA–14

/SCS qualifier • *System Dump Analyzer,* SDA–82

SCU/XMI bus
- I/O address space • *Device Support (A),* 16–5

SCU/XMI bus architecture • *Device Support (A),* 1–16

SDA$INIT logical name • *System Dump Analyzer,* SDA–8

SDA (System Dump Analyzer) • *Programming Resources,* 1–21 to 1–22; *Device Support (A),* 13–22
- analyzing dump file • *Programming Resources,* 1–21
- command format • *System Dump Analyzer,* SDA–10 to SDA–14, SDA–32
- commands • *System Dump Analyzer,* SDA–1 to SDA–2, SDA–39 to SDA–165
- context • *System Dump Analyzer,* SDA–9, SDA–9 to SDA–10
- current process • *Device Support (A),* E–19
- exiting • *System Dump Analyzer,* SDA–33, SDA–55
- expression • *System Dump Analyzer,* SDA–11 to SDA–14
- initialization file • *System Dump Analyzer,* SDA–8
- logging a session • *System Dump Analyzer,* SDA–71
- multiple screen displays • *System Dump Analyzer,* SDA–55
- obtaining help • *System Dump Analyzer,* SDA–58
- recording output • *System Dump Analyzer,* SDA–32, SDA–72
- SET CPU command • *Device Support (A),* E–19
- SHOW CPU command • *Device Support (A),* E–19
- SHOW CRASH command • *Device Support (A),* E–19
- SHOW SPINLOCKS command • *Device Support (A),* E–20
- specifying an alternate system symbol table • *System Dump Analyzer,* SDA–37

SDA (System Dump Analyzer) (Cont.)
- usage summary • *System Dump Analyzer*, SDA–32
- using to debug device driver • *Device Support (A)*, 13–29

SDA current CPU • *System Dump Analyzer*, SDA–10, SDA–68, SDA–74, SDA–89, SDA–93, SDA–126, SDA–157

SDA current process • *System Dump Analyzer*, SDA–9, SDA–10, SDA–68, SDA–73, SDA–93, SDA–126, SDA–157; *Device Support (A)*, E–19

SDA symbol table • *System Dump Analyzer*, SDA–13
- building • *System Dump Analyzer*, SDA–7
- expanding • *System Dump Analyzer*, SDA–8

Search
- anchored • *VAXTPU*, 7–24
- anchoring a pattern • *VAXTPU*, 2–19
- for pattern • *VAXTPU*, 2–11
- synonyms • *RMS*, 7–12
- unanchoring pattern elements • *VAXTPU*, 2–19 to 2–20

SEARCH built-in procedure • *VAXTPU*, 7–327 to 7–331

SEARCH command • *Debugger*, 6–6, CD–120; *System Dump Analyzer*, SDA–66
- displaying default qualifiers for • *Debugger*, 6–7, CD–243
- setting default qualifiers for • *Debugger*, 6–7, CD–174

Search list • *System Services Intro*, 6–2
- See also File specification
- and multiple file locations • *File Applications*, 5–7, 5–8
- as alternative to using wildcard characters • *RMS*, 4–10
- definition • *File Applications*, 5–7
- example • *File Applications*, 5–15
- scope • *Debugger*, 5–8, 5–11, CD–170, CD–241
  - with DECwindows • *Debugger*, 1–8, 1–28
- source file • *Debugger*, 6–2, CD–28, CD–176, CD–246
- translation • *File Applications*, 6–7 to 6–8
- using with Remove service • *RMS*, RMS–82

$SEARCH macro
- for processing wildcard characters • *RMS*, 4–10

Search operations • *System Services Intro*, 3–15

Search service • *File Applications*, 5–8 to 5–12; *RMS*, RMS–91, RMS–92
- condition values • *RMS*, RMS–94
- control block input fields • *RMS*, RMS–92
- control block output fields • *RMS*, RMS–93
- example of completion code handling • *RMS*, 4–12

Search service (Cont.)
- program example • *RMS*, 4–9
- requirement for Parse service • *RMS*, 4–9
- using with wildcard characters and search lists • *RMS*, RMS–92

Search string translation
- requirements for parsing • *RMS*, 4–9

SEARCH_QUIETLY built-in procedure • *VAXTPU*, 7–332 to 7–336

$SECDEF macro • *Device Support (A)*, 19–6

Secondary attribute • *File Applications*, 4–9; *File Def Language*, FDL–2

Secondary bootstrap program (SYSBOOT) • *Device Support (A)*, 13–21

Secondary completion status value field • *File Applications*, 5–12

Secondary controller data channel • *Device Support (A)*, 15–14, 15–15; *Device Support (B)*, 2–57
- obtaining ownership of • *Device Support (B)*, 2–63, 3–100 to 3–101
- releasing • *Device Support (B)*, 3–91

Secondary controller data channel wait queue • *Device Support (B)*, 3–91, 3–101

Secondary device characteristics field
- See FAB$L_SDC field

Secondary exception vector • *Programming Resources*, 9–13

Secondary index
- See Alternate index

Secondary index data record
- See SIDR

Secondary service
- effect on next-record position • *File Applications*, 8–16

Section • *System Services Intro*, 11–8
- characteristic • *System Services Intro*, 11–10
- creating • *System Services Intro*, 11–8; *System Services*, SYS–105
- defining extent • *System Services Intro*, 11–10
- deleting • *Programming Resources*, 8–9; *System Services Intro*, 11–18
- deleting global • *System Services*, SYS–140
- global • *Programming Resources*, 5–15
- global paging file • *System Services Intro*, 11–15
- image • *System Services Intro*, 11–19
- mapping • *Programming Resources*, 8–4; *System Services Intro*, 11–13; *System Services*, SYS–105
- page frame • *System Services Intro*, 11–19
- paging • *System Services Intro*, 11–16
- private • *Programming Resources*, 8–4
- releasing • *System Services Intro*, 11–18

Section (Cont.)
- unmapping • *System Services Intro*, 11–18
- updating • *Programming Resources*, 8–9
- using to share data • *System Services Intro*, 11–18
- writing back • *System Services Intro*, 11–19
- writing modifications to disk • *System Services*, SYS–532, SYS–536

Section file • *VAXTPU*, 5–16
- created with EVE editor$BUILD • *VAXTPU*, G–10 to G–11
- creating • *VAXTPU*, 4–23
- debugging • *VAXTPU*, 4–34
- default • *VAXTPU*, 4–21
- definition • *VAXTPU*, 1–10
- extending • *VAXTPU*, 4–24
- processing • *VAXTPU*, 4–24, 4–25
- recommended conventions • *VAXTPU*, 4–28
- updating • *System Services*, SYS–532, SYS–536

Section name
- made available to debugger • *MACRO*, 6–23

/SECTION qualifier • *VAXTPU*, 4–25, 5–16

"Section" string constant parameter to GET_INFO • *VAXTPU*, 7–178

"Section_file" string constant parameter to GET_INFO • *VAXTPU*, 7–178, 7–207

section_id data type • *Routines Intro*, A–12t

section_name data type • *Routines Intro*, A–12t

Sector • *File Applications*, 1–5

Sector translation • *I/O User's I*, 3–18

Security • *Programming Resources*, 1–23
- image • *Debugger*, 5–6

SECURITY.EXE
- global symbols • *System Dump Analyzer*, SDA–61

Security considerations • *VAXTPU*, 1–12, 7–59, 7–234, 7–235, 7–406

Security services • *System Services Intro*, 1–1

Seek operation • *I/O User's I*, 3–16; *Device Support (A)*, 8–6
- overlapping with data transfer • *Device Support (A)*, 8–2

Seek time • *File Applications*, 1–5

Segmented key • *File Def Language*, FDL–30; *RMS*, 13–13
- restriction against overlapping • *RMS*, 13–13

SEGn secondary • *File Def Language*, FDL–40

SEGn_LENGTH attribute • *File Def Language*, FDL–30

SEGn_POSITION attribute • *File Def Language*, FDL–30

SELECT built-in procedure • *VAXTPU*, 7–337 to 7–339

SELECT command • *Debugger*, 7–20, CD–123

Selected map register
- See MBA$L_SMR

Selection • *VAXTPU*, 4–16
- dynamic • *VAXTPU*, 4–17
- found range • *VAXTPU*, 4–18
- static • *VAXTPU*, 4–17
- using MODIFY_RANGE built-in to alter • *VAXTPU*, 7–273

/SELECTIVE_SEARCH positional qualifier • *Librarian*, LIB–40; *Linker*, LINK–27

Select range
- in EVE editor • *VAXTPU*, 4–16

SELECT_RANGE built-in procedure • *VAXTPU*, 7–340 to 7–341

Self-relative queue • *MACRO*, 9–85
- validating • *System Dump Analyzer*, SDA–164

Self-test status • *Device Support (A)*, 16–25

SELF_INSERT keyword • *VAXTPU*, 7–470

"Self_insert" string constant parameter to GET_INFO • *VAXTPU*, 7–204

/SELF_RELATIVE qualifier • *System Dump Analyzer*, SDA–164

Semaphore • *Programming Resources*, 4–17; *RTL Parallel Processing*, 4–9
- See also Synchronization
- adjusting maximum value • *RTL Parallel Processing*, 4–13
- binary • *Programming Resources*, 4–17; *RTL Parallel Processing*, 4–10
- counting • *Programming Resources*, 4–17; *RTL Parallel Processing*, 4–10
- creating • *RTL Parallel Processing*, 4–11
- decrementing • *RTL Parallel Processing*, 4–12
- deleting • *RTL Parallel Processing*, 4–12
- incrementing • *RTL Parallel Processing*, 4–13
- reading • *RTL Parallel Processing*, 4–13
- setting maximum value • *RTL Parallel Processing*, 4–14

Semaphore synchronization
- advantages and disadvantages • *RTL Parallel Processing*, 5–8
- PPL$ routines for • *RTL Parallel Processing*, 4–11 to 4–14

Semicolon (;)
- as statement separator • *VAXTPU*, 1–8, 3–4, 3–15, 3–16, 3–17, 4–3
- command separator • *Debugger*, CD–4

SEND built-in procedure • *VAXTPU*, 7–342 to 7–343

SEND_CLIENT_MESSAGE built-in procedure • *VAXTPU*, 7–344 to 7–345

SEND_EOF built-in procedure • *VAXTPU*, 7–346

Sense device characteristics function • *Device Support (A)*, 7–9
Sense device mode function • *Device Support (A)*, 7–9
Sense tape mode function • *I/O User's I*, 6–22
Separate window
  debugger (on workstation using VWS) • *Debugger*, 9–5, CD–154
Separator
  in symbolic name • *RMS*, 2–3
  semicolon used as • *VAXTPU*, 1–8, 3–4, 3–15, 3–16, 3–17, 4–3
SEQUENCE keyword
  description • *National Char Set*, NCS–13
Sequential access • *File Applications*, 8–6
  mode • *File Applications*, 1–2
  to indexed files • *File Applications*, 2–4, 8–10
  to relative files • *File Applications*, 2–4, 8–9
  to sequential files • *File Applications*, 2–3
  use with sequential files • *File Applications*, 8–7
  with multibuffer count • *File Applications*, 3–26
SEQUENTIAL attribute • *File Def Language*, FDL–22
Sequential file • *File Applications*, 2–14; *File Def Language*, FDL–25
  advantages and disadvantages of using • *File Applications*, 2–15
  allocating • *File Applications*, A–1
  buffering • *File Applications*, 7–18 to 7–19
  creating • *Programming Resources*, 8–10
  designing • *File Applications*, 3–9 to 3–12
  examining • *File Applications*, 10–12, 10–13
  maximum record size • *File Applications*, 3–10
  merging • *Programming Resources*, 8–13, 8–14
  optimizing performance • *File Applications*, 3–9 to 3–12
  organization • *File Applications*, 1–2
  read-ahead and write-behind • *File Applications*, 3–9
  record access • *File Applications*, 8–7 to 8–8, 8–12 to 8–13
  sorting • *Programming Resources*, 8–13, 8–14
  structure • *Analyze/RMS_File*, ARMS–1
  tuning • *File Applications*, 3–9 to 3–12
  updating • *Programming Resources*, 8–11
Sequential only option
  See FAB$V_SQO option
/SEQUENTIAL qualifier • *File Applications*, 7–19
SEQUENTIAL_ONLY attribute • *File Def Language*, FDL–24
Serial line multiplexer • *I/O User's I*, 8–1
Service
  allowable program execution modes • *RMS*, 2–7
Service (Cont.)
  block I/O • *RMS*, 3–5
  calling example • *RMS*, 3–11
  invoking at run time • *RMS*, 3–1
  naming conventions • *RMS*, 3–3
  passing argument list to • *RMS*, 3–10
  restrictions to calling • *RMS*, 2–7
Service macro
  description • *RMS*, 3–1
  for creating and processing files • *RMS*, 4–1
  format • *RMS*, 3–10, 3–11
  format rules • *RMS*, 3–11
  types • *RMS*, 3–12
Service routine
  AST • *System Services Intro*, 5–4
SET (ACTIVE_AREA) built-in procedure • *VAXTPU*, 7–350
SET (AUTO_REPEAT) built-in procedure • *VAXTPU*, 7–353 to 7–354
SET (BELL) built-in procedure • *VAXTPU*, 7–355 to 7–356
SET (CLIENT_MESSAGE) built-in procedure • *VAXTPU*, 7–357 to 7–358
SET (COLUMN_MOVE_VERTICAL) built-in procedure • *VAXTPU*, 7–359 to 7–360
SET (CROSS_WINDOW_BOUNDS) built-in procedure • *VAXTPU*, 7–361
SET (DEBUG) built-in procedure • *VAXTPU*, 7–362 to 7–365
SET (DEFAULT_DIRECTORY) built-in procedure • *VAXTPU*, 7–366
SET (DETACHED_ACTION) built-in procedure • *VAXTPU*, 7–367 to 7–369
SET (DISPLAY_VALUE) built-in procedure • *VAXTPU*, 7–370
SET (DRM_HIERARCHY) built-in procedure • *VAXTPU*, 7–371
SET (ENABLE_RESIZE) built-in procedure • *VAXTPU*, 7–372
SET (EOB_TEXT) built-in procedure • *VAXTPU*, 7–374
SET (ERASE_UNMODIFIABLE) built-in procedure • *VAXTPU*, 7–375 to 7–377
SET (FACILITY_NAME) built-in procedure • *VAXTPU*, 7–378
SET (FORWARD) built-in procedure • *VAXTPU*, 7–379
SET (GLOBAL_SELECT) built-in procedure • *VAXTPU*, 7–380
SET (GLOBAL_SELECT_GRAB) built-in procedure • *VAXTPU*, 7–382
SET (GLOBAL_SELECT_READ) built-in procedure • *VAXTPU*, 7–385

SET (GLOBAL_SELECT_TIME) built-in procedure • *VAXTPU*, 7–387
SET (GLOBAL_SELECT_UNGRAB) built-in procedure • *VAXTPU*, 7–389
SET (HEIGHT) built-in procedure • *VAXTPU*, 7–391
SET (ICONIFY_PIXMAP) built-in procedure • *VAXTPU*, 7–395 to 7–396
SET (ICON_NAME) built-in procedure • *VAXTPU*, 7–392
SET (ICON_PIXMAP) built-in procedure • *VAXTPU*, 7–393 to 7–394
SET (INFORMATIONAL) built-in procedure • *VAXTPU*, 7–397
SET (INPUT_FOCUS) built-in procedure • *VAXTPU*, 7–398
SET (INPUT_FOCUS_GRAB) built-in procedure • *VAXTPU*, 7–400
SET (INPUT_FOCUS_UNGRAB) built-in procedure • *VAXTPU*, 7–402
SET (INSERT) built-in procedure • *VAXTPU*, 7–404
SET (JOURNALING) built-in procedure • *VAXTPU*, 7–405 to 7–407
SET (KEYSTROKE_RECOVERY) built-in procedure • *VAXTPU*, 7–408 to 7–409
SET (KEY_MAP_LIST) built-in procedure • *VAXTPU*, 7–410 to 7–411
SET (LEFT_MARGIN) built-in procedure • *VAXTPU*, 7–412 to 7–413
SET (LEFT_MARGIN_ACTION) built-in procedure • *VAXTPU*, 7–414 to 7–415
SET (LINE_NUMBER) built-in procedure • *VAXTPU*, 7–416 to 7–417
SET (MAPPED_WHEN_MANAGED) built-in procedure • *VAXTPU*, 7–418
SET (MARGINS) built-in procedure • *VAXTPU*, 7–419 to 7–420
SET (MAX_LINES) built-in procedure • *VAXTPU*, 7–421
SET (MENU_POSITION) built-in procedure • *VAXTPU*, 7–422 to 7–423
SET (MESSAGE_ACTION_LEVEL) built-in procedure • *VAXTPU*, 7–424 to 7–425
SET (MESSAGE_ACTION_TYPE) built-in procedure • *VAXTPU*, 7–426
SET (MESSAGE_FLAGS) built-in procedure • *VAXTPU*, 7–427 to 7–428
SET (MODIFIABLE) built-in procedure • *VAXTPU*, 7–429 to 7–430
SET (MODIFIED) built-in procedure • *VAXTPU*, 7–431
SET (MOUSE) built-in procedure • *VAXTPU*, 7–432 to 7–433
SET (NO_WRITE) built-in procedure • *VAXTPU*, 7–434
SET (OUTPUT) built-in procedure • *VAXTPU*, 7–203
SET (OUTPUT_FILE) built-in procedure • *VAXTPU*, 7–435
SET (OVERSTRIKE) built-in procedure • *VAXTPU*, 7–436
SET (PAD) built-in procedure • *VAXTPU*, 7–437 to 7–438
SET (PAD_OVERSTRUCK_TABS) built-in procedure • *VAXTPU*, 7–439 to 7–440
SET (PERMANENT) built-in procedure • *VAXTPU*, 7–441
SET (POST_KEY_PROCEDURE) built-in procedure • *VAXTPU*, 7–442 to 7–443
SET (PRE_KEY_PROCEDURE) built-in procedure • *VAXTPU*, 7–444 to 7–445
SET (PROMPT_AREA) built-in procedure • *VAXTPU*, 7–446 to 7–447
SET (RECORD_ATTRIBUTE) built-in procedure • *VAXTPU*, 7–448 to 7–450
SET (RESIZE_ACTION) built-in procedure • *VAXTPU*, 7–451
SET (REVERSE) built-in procedure • *VAXTPU*, 7–453
SET (RIGHT_MARGIN) built-in procedure • *VAXTPU*, 7–454 to 7–455
SET (RIGHT_MARGIN_ACTION) built-in procedure • *VAXTPU*, 7–456 to 7–457
SET (SCREEN_LIMITS) built-in procedure • *VAXTPU*, 7–458
SET (SCREEN_UPDATE) built-in procedure • *VAXTPU*, 7–460 to 7–461
SET (SCROLLING) built-in procedure • *VAXTPU*, 7–467 to 7–469
SET (SCROLL_BAR) built-in procedure • *VAXTPU*, 7–462
example of use • *VAXTPU*, B–22 to B–25
SET (SCROLL_BAR_AUTO_THUMB) built-in procedure • *VAXTPU*, 7–465
example of use • *VAXTPU*, B–22 to B–25
SET (SELF_INSERT) built-in procedure • *VAXTPU*, 7–470 to 7–471
SET (SHIFT_KEY) built-in procedure • *VAXTPU*, 7–472 to 7–473
SET (SPECIAL_ERROR_SYMBOL) built-in procedure • *VAXTPU*, 7–474 to 7–475
SET (STATUS_LINE) built-in procedure • *VAXTPU*, 7–476 to 7–478
SET (SUCCESS) built-in procedure • *VAXTPU*, 7–479
SET (SYSTEM) built-in procedure • *VAXTPU*, 7–480
SET (TAB_STOPS) built-in procedure • *VAXTPU*, 7–481 to 7–482
SET (TEXT) built-in procedure • *VAXTPU*, 7–483 to 7–485

SET (TIMER) built-in procedure • *VAXTPU,* 7–486 to 7–487
SET (TRACEBACK) built-in procedure • *VAXTPU,* 7–488 to 7–489
SET (UNDEFINED_KEY) built-in procedure • *VAXTPU,* 7–490 to 7–491
SET (VIDEO) built-in procedure • *VAXTPU,* 7–492 to 7–493
SET (WIDGET) built-in procedure • *VAXTPU,* 7–494
  example of use • *VAXTPU,* B–22 to B–25, B–25 to B–27
  using to specify resource values • *VAXTPU,* 4–12
SET (WIDGET_CALLBACK) built-in procedure • *VAXTPU,* 7–499
  example of use • *VAXTPU,* B–22 to B–25
  using to specify callback routine • *VAXTPU,* 4–9
SET (WIDGET_CALL_DATA) built-in procedure • *VAXTPU,* 7–496 to 7–498
SET (WIDTH) built-in procedure • *VAXTPU,* 7–501 to 7–502
SET ABORT_KEY command • *Debugger,* 2–7, CD–127
Set All Processes Writable command • *Delta/XDelta,* DELTA–43
SET ATSIGN command • *Debugger,* 8–2, CD–129
Set attention AST
  See Attention AST
SET BREAK command • *Debugger,* 3–9, 6–7, 9–10, 11–3, CD–130
SET built-in procedure • *VAXTPU,* 7–347 to 7–349
  WIDGET • *VAXTPU,* 4–10
SET CARD_READER command • *I/O User's I,* 2–2
Set characteristic
  card reader • *I/O User's I,* 2–7
  line printer • *I/O User's I,* 5–9
  magnetic tape • *I/O User's I,* 6–23
  terminal • *I/O User's I,* 8–38
SET command • *File Def Language,* FDL–66
SET COMMAND command
  See also Command Definition Utility
  delete mode • *Command Def,* CDU–15, CDU–39
  input for • *Command Def,* CDU–44
  object mode • *Command Def,* CDU–16, CDU–41
  output from • *Command Def,* CDU–42
  processing modes • *Command Def,* CDU–14
  qualifiers for • *Command Def,* CDU–38 to CDU–44
  replace mode • *Command Def,* CDU–15, CDU–43
SET CPU command • *System Dump Analyzer,* SDA–10, SDA–68
  analyzing a running system • *System Dump Analyzer,* SDA–9
SET DEFAULT command • *File Applications,* 6–14, 6–15
  /TRANSLATION_ATTRIBUTES qualifier • *File Applications,* 6–15
SET DEFINE command • *Debugger,* 8–6, CD–138
Set device characteristics function • *Device Support (A),* 7–9; *Device Support (B),* 1–76
Set device mode function • *Device Support (A),* 7–9; *Device Support (B),* 1–76
Set Display Mode command • *Delta/XDelta,* DELTA–16
SET ECO command • *Patch,* PAT–75
  affect of UPDATE command • *Patch,* PAT–89
  applying patches • *Patch,* PAT–2
SET EDITOR command • *Debugger,* CD–139
SET EVENT_FACILITY command • *Debugger,* CD–141
SET FILE command
  /ACL qualifier • *File Applications,* 4–22
  /EXTENSION qualifier • *File Applications,* 3–5
  for changing global buffer count value • *RMS,* 5–19
  /GLOBAL_BUFFERS qualifier • *File Applications,* 3–9, 7–22
SET HOST facility • *I/O User's I,* 8–11
SET IMAGE command • *Debugger,* 5–15, CD–142
  effect on symbol definitions • *Debugger,* CD–48
SETIPL macro • *Device Support (A),* 3–9, 3–10, E–4; *Device Support (B),* 2–65
  example • *Device Support (B),* 2–66
  replacing with spin lock synchronization macro • *Device Support (A),* E–13
SET KEY command • *Debugger,* 8–10, CD–144
SET LANGUAGE command • *Debugger,* 4–10, CD–145
SET LOG command • *Debugger,* 8–5, CD–147; *System Dump Analyzer,* SDA–71
  compared with SET OUTPUT command • *System Dump Analyzer,* SDA–71
SET MARGINS command • *Debugger,* 6–9, CD–148
SET MAX_SOURCE_FILES command • *Debugger,* 6–3, CD–151
SET MESSAGE command • *Message,* MSG–5
Set mode
  card reader • *I/O User's I,* 2–7
  line printer • *I/O User's I,* 5–9
  magnetic tape • *I/O User's I,* 6–23
  mailbox • *I/O User's I,* 7–9
  terminal • *I/O User's I,* 8–38
SET MODE command • *Debugger,* CD–152; *Patch,* PAT–76
Set mode function • *Device Support (B),* 1–76

SET MODE [NO]DYNAMIC command • *Debugger*, 5–7, 5–14, CD–152

SET MODE [NO]G_FLOAT command • *Debugger*, CD–152

SET MODE [NO]INTERRUPT command • *Debugger*, 10–6, CD–152

SET MODE [NO]KEYPAD command • *Debugger*, 8–8, CD–153, B–1

SET MODE [NO]LINE command • *Debugger*, CD–153

SET MODE [NO]OPERANDS command • *Debugger*, 4–20, CD–153

SET MODE [NO]SCREEN command • *Debugger*, 7–1, CD–154

SET MODE [NO]SCROLL command • *Debugger*, CD–154

SET MODE [NO]SEPARATE command • *Debugger*, 9–5, CD–154

SET MODE [NO]SYMBOLIC command • *Debugger*, 4–14, CD–154

SET MODULE command • *Debugger*, 5–7, CD–156; *Patch*, PAT–78

SET NOLOG command • *System Dump Analyzer*, SDA–71

SET OUTPUT command • *Debugger*, CD–159; *System Dump Analyzer*, SDA–72

- compared with SET LOG command • *System Dump Analyzer*, SDA–71

SET OUTPUT [NO]LOG command • *Debugger*, 8–5 CD–159

SET OUTPUT [NO]SCREEN_LOG command • *Debugger*, 8–6, CD–159

SET OUTPUT [NO]TERMINAL command • *Debugger*, CD–159

SET OUTPUT [NO]VERIFY command • *Debugger*, 8–2, CD–159

SET PATCH_AREA command • *Patch*, PAT–79

- creating and accessing patch area • *Patch*, PAT–19
- with /INITIALIZE qualifier • *Patch*, PAT–80

SET PROCESS command • *Debugger*, 10–7, 10–8, CD–161; *System Dump Analyzer*, SDA–9, SDA–73; *Device Support (A)*, E–19

SET PROMPT command • *Debugger*, CD–165

SET PROTECTION command • *File Applications*, 4–21

SET RADIX command • *Debugger*, 4–11, 9–8, CD–168

SET RMS command • *System Dump Analyzer*, SDA–76

SET RMS_DEFAULT command • *RMS*, 7–6

- /BUFFER_COUNT qualifier • *File Applications*, 3–8, 3–11, 3–13, 7–19, 7–20

SET RMS_DEFAULT command (Cont.)

- /EXTEND_QUANTITY qualifier • *File Applications*, 3–5, 9–8
- /INDEXED qualifier • *File Applications*, 7–20
- /RELATIVE/BUFFER_COUNT qualifier • *File Applications*, 3–14
- /RELATIVE qualifier • *File Applications*, 7–19
- /SEQUENTIAL qualifier • *File Applications*, 7–19
- to limit default extension quantity • *RMS*, 5–6

SET SCOPE command • *Debugger*, 5–11, 6–4, 7–6, 7–10, CD–170; *Patch*, PAT–84

SET SEARCH command • *Debugger*, 6–7, CD–174

SET SOURCE command • *Debugger*, 6–2, CD–176

SET STEP command • *Debugger*, 3–7, 6–7, 11–3, CD–179

SETSWM • *Programming Resources*, 10–4

Set system failure exception mode

- See SYS$SETSFM

SET TASK command • *Debugger*, CD–182

SET TERMINAL command • *Debugger*, 7–23, CD–185; *I/O User's I*, 8–4, 8–19, 8–25

SET TRACE command • *Debugger*, 3–10, 6–7, 9–10, 11–3, CD–187

Set translation mode • *I/O User's I*, 2–2

SET TYPE command • *Debugger*, 4–25, CD–195

SET TYPE/OVERRIDE command • *Debugger*, 4–26, CD–195

SET VECTOR_MODE command • *Debugger*, 11–20, CD–198

SET VERIFY command • *Linker*, 3–4

SET WATCH command • *Debugger*, 3–17, 6–7, 11–3, CD–200

SET WINDOW command • *Debugger*, 7–15, CD–207

/SET_STATE qualifier • *Debugger*, 8–10, CD–51; *System Dump Analyzer*, SDA–45

/SEVERE qualifier

- in message definition • *Message*, MSG–23

Severity code • *Routines Intro*, 2–9, 2–10

- handling of • *Routines Intro*, 2–10
- in completion status code field • *RMS*, 2–6
- interpreting • *Routines Intro*, 2–10
- meanings • *Routines Intro*, 2–10
- symbols • *Routines Intro*, 2–10

.SEVERITY directive • *Programming Resources*, 9–8

Severity directive (.SEVERITY)

- in message source file • *Message*, MSG–26

Severity level • *Message*, MSG–1

S field in symbolic offset

- for specifying field length • *RMS*, 2–3

SFSB (shared file synchronization block) • *System Dump Analyzer*, SDA–77

- Shadow set
  - displaying SDA information • *System Dump Analyzer,* SDA–99
- Shadow set virtual unit driver • *I/O User's I,* 10–1
  - functions • *I/O User's I,* 10–4
  - hardware configurations • *I/O User's I,* 10–2
  - system configuration • *I/O User's I,* 10–2
- Shareable device • *Device Support (B),* 1–75
- Shareable image • *Programming Resources,* 5–3; *Modular Procedures,* A–6; *Linker,* 6–2; *Patch,* PAT–3, PAT–19; *RTL Intro,* 1–19
  - See also Module
  - activating • *RTL Library,* LIB–160
  - adding • *Programming Resources,* 5–8
  - as separate cluster • *Linker,* 6–7
  - based • *Linker,* 1–11, 4–9, 6–7
  - benefit of • *Linker,* 4–1
  - CANCEL IMAGE command • *Debugger,* 5–14, CD–22
  - code references to, in map • *Linker,* 5–8
  - coding for position independence • *Linker,* 4–5
  - contents of • *Programming Resources,* 5–3; *Linker,* 1–4, 2–2
  - creating • *Programming Resources,* 5–6; *Modular Procedures,* 5–4; *Linker,* 1–11, 4–10
  - debugging • *Debugger,* 5–12
    - with DECwindows • *Debugger,* 1–30
  - default directory of • *Linker,* 1–11, 4–12
  - default file type • *Programming Resources,* 5–9
  - default location • *Programming Resources,* 5–9
  - deleting • *Programming Resources,* 5–8
  - files
    - used as linker input • *Linker,* 1–4
  - for COMMON area • *Linker,* 4–22
  - ID
    - major • *Programming Resources,* 5–5
    - minor • *Programming Resources,* 5–5
    - specifying major • *Programming Resources,* 5–7
    - specifying minor • *Programming Resources,* 5–7
  - identification of • *Linker,* LINK–28
  - input to linker • *Linker,* 1–4, 2–2, 6–3
  - in resource allocation • *Linker,* 4–13
  - installation of • *Linker,* 4–1, 4–11
  - library • *Programming Resources,* 5–8; *Linker,* 1–11, 4–11
  - linking • *Programming Resources,* 5–7, 5–8
  - linking of multiple • *Linker,* 4–18
  - linking several • *Linker,* 4–22
  - listing • *Programming Resources,* 5–8
  - location of by image activator • *Linker,* 4–12
- Shareable image (Cont.)
  - match control for • *Linker,* 1–8, 3–7
  - memory allocation for • *Linker,* 6–7
  - output of linker • *Linker,* 1–5, 2–5
  - position independent • *Linker,* 1–10, 4–4, 6–7
  - private copy of • *Linker,* 4–12
  - privileged • *Linker,* 1–11, 4–11
  - processing of • *Linker,* 6–14
  - program sections in • *Linker,* 1–10, 4–3
  - protection of • *Linker,* 1–8, 3–11
  - replacing • *Programming Resources,* 5–8
  - resolving references to • *Linker,* 6–7
  - restriction to use as input file • *Linker,* 1–1
  - rules for upward compatibility • *Linker,* 1–11, 4–9
  - SET BREAK/INTO command • *Debugger,* 3–14, CD–133
  - SET IMAGE command • *Debugger,* 5–15, CD–142
  - SET STEP INTO command • *Debugger,* 3–9, CD–180
  - SET TRACE/INTO command • *Debugger,* 3–14, CD–190
  - shareability • *Linker,* 4–3
    - guidelines for • *Linker,* 1–10, 4–4
  - shared image • *Programming Resources,* 5–10
  - SHOW IMAGE command • *Debugger,* 5–14, CD–222
  - specification of • *Linker,* 1–11, 4–11
  - specifying alternate locations • *Programming Resources,* 5–9
  - STEP/INTO command • *Debugger,* CD–266
  - symbol table of • *Linker,* 6–2
  - transfer vector • *Programming Resources,* 5–3, 5–6; *Linker,* 1–10, 4–5
  - universal symbol • *Programming Resources,* 5–5; *Linker,* 1–11, 4–10
  - updating • *Modular Procedures,* 6–6; *Linker,* 3–8, 3–9
  - use for • *Linker,* 1–5, 2–5
  - use of for COMMON area • *Linker,* 4–18
  - use of GSMATCH • *Linker,* 3–8, 3–9, 4–10
  - writing code for • *Linker,* 4–3
- Shareable image library • *Programming Resources,* 1–18; *Librarian,* LIB–1, LIB–3
  - See also Shareable image
  - as user default library • *Linker,* LINK–21
  - content of • *Linker,* 1–5, 2–3
  - creating • *Modular Procedures,* 5–10
  - input to linker • *Linker,* 1–5, 2–3
  - processing of • *Linker,* 6–13, 6–14
  - shareable image in • *Librarian,* LIB–3
  - system default • *Linker,* LINK–18
  - updating • *Modular Procedures,* 6–7

/SHAREABLE positional qualifier • *Linker*, LINK–28
/SHAREABLE qualifier • *Debugger*, 5–12; *Linker*, 1–5, 2–5, LINK–15
  LIBRARY command • *Programming Resources*, 5–8
Shared access • *File Applications*, 3–3
  requirement to specify • *RMS*, 4–1
Shared files • *Programming Resources*, 5–19
  See also File sharing
  end-of-file positioning • *RMS*, RMS–7
Shared file synchronization block
  See SFSB
Shared image
  creating • *Programming Resources*, 5–10
Shared memory • *RTL Parallel Processing*, 3–1 to 3–3
  creating • *RTL Parallel Processing*, 3–1
  definition of • *RTL Parallel Processing*, 1–2
  deleting • *RTL Parallel Processing*, 3–3
  flushing to disk • *RTL Parallel Processing*, 3–3
  possible error when creating • *RTL Parallel Processing*, 3–2
/SHARED qualifier
  in .FACILITY directive • *Message*, MSG–18
/SHARE qualifier • *Debugger*, 3–14, CD–133, CD–190, CD–231, CD–266; *Librarian*, LIB–41; *Convert*, CONV–21
SHARING attribute • *File Def Language*, FDL–2, FDL–36
Sharing data
  VMS RMS shared files • *Programming Resources*, 5–19
SHARING primary attribute
  secondary attributes • *File Applications*, 7–4, 7–7, 7–22
SHDRIVER.EXE • *I/O User's I*, 10–1
SHIFT built-in procedure • *VAXTPU*, 7–503 to 7–504
Shifting operator (@) • *System Dump Analyzer*, SDA–13
Shift instruction
  vector • *MACRO*, 10–67
SHIFT key
  restriction on defining in EVE • *VAXTPU*, 7–472
Shift operator • *MACRO*, 3–16
"Shift_amount" string constant parameter to GET_INFO • *VAXTPU*, 7–225
SHIFT_KEY keyword • *VAXTPU*, 7–472
"Shift_key" string constant parameter to GET_INFO • *VAXTPU*, 7–204, 7–207
Short literal mode
  usage restricted in vector floating-point instructions • *MACRO*, 10–16
Should Be Zero
  See SBZ field
SHOW (KEYWORDS) built-in procedure • *VAXTPU*, 2–5
SHOW ABORT_KEY command • *Debugger*, CD–209
SHOW AST command • *Debugger*, 9–17, CD–210
SHOW ATSIGN command • *Debugger*, 8–2, CD–211
SHOW BREAK command • *Debugger*, 3–10, CD–212
SHOW built-in procedure • *VAXTPU*, 7–505 to 7–507
SHOW CALLS command • *Debugger*, 2–14, 3–3, 9–11, 9–17, CD–214
SHOW CALL_FRAME command • *System Dump Analyzer*, SDA–65, SDA–79
SHOW CLUSTER command • *System Dump Analyzer*, SDA–82
SHOW CLUSTER/SCS command • *System Dump Analyzer*, SDA–123
SHOW CONNECTIONS command • *System Dump Analyzer*, SDA–87
SHOW CPU command • *System Dump Analyzer*, SDA–10, SDA–68, SDA–89
  analyzing a running system • *System Dump Analyzer*, SDA–9
SHOW CRASH command • *System Dump Analyzer*, SDA–10, SDA–15, SDA–16, SDA–68, SDA–93
  analyzing a running system • *System Dump Analyzer*, SDA–9
SHOW DEFAULTS BUFFER command • *VAXTPU*, 4–32
SHOW DEFINE command • *Debugger*, 8–6, CD–216
SHOW DEVICE command • *System Dump Analyzer*, SDA–15, SDA–24, SDA–98; *Device Support (B)*, 1–80
SHOW directive • *MACRO*, 6–89
SHOW DISPLAY command • *Debugger*, 7–12, CD–217
SHOW EDITOR command • *Debugger*, CD–219
SHOW entry point • *Modular Procedures*, 4–8
SHOW EVENT_FACILITY command • *Debugger*, 3–15, CD–220
SHOW EXECUTIVE command • *System Dump Analyzer*, SDA–15, SDA–104
SHOW EXIT_HANDLERS command • *Debugger*, 9–17, CD–221
SHOW HEADER command • *System Dump Analyzer*, SDA–106
SHOW IMAGE command • *Debugger*, 5–14, CD–222
Showing version number • *VAXTPU*, 4–2
SHOW KEY command • *Debugger*, 8–9, CD–223
SHOW LANGUAGE command • *Debugger*, 4–10, CD–226
SHOW LOCK command • *System Dump Analyzer*, SDA–108

SHOW LOG command • *Debugger*, 8–6, CD–227
SHOW MARGINS command • *Debugger*, 6–9, CD–228
SHOW MAX_SOURCE_FILES command • *Debugger*, 6–3, CD–229
SHOW MEMORY command • *System Dump Analyzer*, SDA–3
SHOW MODE command • *Debugger*, CD–230; *Patch*, PAT–85
SHOW MODULE command • *Debugger*, 5–7, CD–231; *Patch*, PAT–86
SHOW OUTPUT command • *Debugger*, 8–2, 8–6, CD–234
SHOW PAGE_TABLE command • *System Dump Analyzer*, SDA–23, SDA–111
SHOW PATCH_AREA command • *Patch*, PAT–87
SHOW PFN_DATA command • *System Dump Analyzer*, SDA–115
SHOW POOL command • *System Dump Analyzer*, SDA–118
SHOW PORTS command • *System Dump Analyzer*, SDA–123
SHOW PROCESS/ALL command • *System Dump Analyzer*, SDA–128
SHOW PROCESS command • *Debugger*, 10–3, 11–2, CD–235; *System Dump Analyzer*, SDA–74, SDA–126
SHOW PROCESS/LOCKS command • *System Dump Analyzer*, SDA–108
SHOW PROCESS/RMS command • *System Dump Analyzer*, SDA–147
  selecting display options • *System Dump Analyzer*, SDA–76
SHOW RADIX command • *Debugger*, 4–11, CD–240
SHOW RESOURCE command • *System Dump Analyzer*, SDA–108, SDA–143
SHOW RMS command • *System Dump Analyzer*, SDA–147
SHOW RMS_DEFAULT command • *File Applications*, 3–8, 3–14; *Convert*, CONV–19; *File Def Language*, FDL–30
  current default extension size • *File Applications*, 9–8
  current process-default buffer count • *File Applications*, 7–19 to 7–20
SHOW RSPID command • *System Dump Analyzer*, SDA–148
SHOW SCOPE command • *Debugger*, 5–12, CD–241; *Patch*, PAT–88
SHOW SEARCH command • *Debugger*, 6–7, CD–243
SHOW SELECT command • *Debugger*, 7–22, CD–244
SHOW SOURCE command • *Debugger*, 6–2, CD–246
SHOW SPINLOCKS command • *System Dump Analyzer*, SDA–151; *Device Support (A)*, E–17
SHOW STACK command • *Debugger*, 9–13, CD–248; *System Dump Analyzer*, SDA–21, SDA–157
SHOW STEP command • *Debugger*, 3–8, CD–249
SHOW SUMMARY command • *System Dump Analyzer*, SDA–126, SDA–159
SHOW SYMBOL command • *Debugger*, 5–9, CD–250; *System Dump Analyzer*, SDA–161
SHOW SYMBOL/DEFINED command • *Debugger*, 8–6
SHOW TASK command • *Debugger*, CD–253
SHOW TERMINAL command • *Debugger*, 7–24, CD–256
SHOW TRACE command • *Debugger*, 3–10, CD–257
SHOW TYPE command • *Debugger*, 4–26, CD–259
SHOW VECTOR_MODE command • *Debugger*, 11–20, CD–260
SHOW WATCH command • *Debugger*, 3–17, CD–261
SHOW WINDOW command • *Debugger*, 7–15, CD–262
SHOW_BUFFER identifier • *VAXTPU*, 7–506
SHOW_BUFFER variable • *VAXTPU*, 4–29
SHR$_HALTED • *I/O User's II*, 4–32
SHR$_NOCMDMEM • *I/O User's II*, 4–28, 4–31, 4–32, 4–33
SHR$_QEMPTY • *I/O User's II*, 4–32
SHR field
  See FAB$B_SHR field
Shutdown
  operator-requested • *System Dump Analyzer*, SDA–5
SIDR (secondary index data record) • *File Applications*, 3–15, 3–19, 10–22; *Analyze/RMS_File*, ARMS–7; *File Def Language*, FDL–5
  for storing sorted pointers • *Convert*, CONV–12
SID register
  displaying • *System Dump Analyzer*, SDA–90
Signal
  enabling an event • *RTL Parallel Processing*, 4–7
Signal argument vector • *RTL Library*, 4–7, 4–9, 4–20
Signal array • *Programming Resources*, 9–14; *System Dump Analyzer*, SDA–18
Signal array argument • *System Services Intro*, 10–10
Signaler's registers • *Routines Intro*, 2–53
Signaling • *Programming Resources*, 9–5
  changing to return status • *Programming Resources*, 9–6
Signaling a condition • *Routines Intro*, 2–47

Signaling and condition handling • *Modular Procedures*, 2–22
Signaling error conditions • *Modular Procedures*, 2–23
Signaling errors
  example in a VAX MACRO program • *File Applications*, 5–12
Signal primitive operation • *RTL Parallel Processing*, 4–10
Signed byte storage directive (.SIGNED BYTE) • *MACRO*, 6–91
Signed word storage directive (.SIGNED_WORD) • *MACRO*, 6–92
.SIGNED_BYTE directive • *MACRO*, 6–91
.SIGNED_WORD directive • *MACRO*, 6–92
Sign-Extended longword field • *RTL Library*, LIB–142
Significance indicator • *MACRO*, 9–185
Sign representation
  preference for key type coding • *RMS*, 13–7
SII controller • *Device Support (A)*, 1–19
SII integral adapter • *I/O User's I*, 3–4
/SILENT qualifier • *Debugger*, 3–14, CD–133, CD–190, CD–201, CD–266
Simple breakpoint • *Delta/XDelta*, DELTA–28
Simple key • *RMS*, 13–13
Simplified callable interface
  See VAXTPU routines
/SINCE qualifier • *Librarian*, LIB–42; *National Char Set*, NCS–41
Sine
  hyperbolic • *RTL Math*, MTH–100, MTH–133
  in degrees • *RTL Math*, MTH–99, MTH–127, MTH–131
  in radians • *RTL Math*, MTH–98, MTH–122, MTH–124
  of complex number • *RTL Math*, MTH–53, MTH–54
Single instruction access • *Modular Procedures*, 3–22
SIRR (software interrupt request register) • *Device Support (A)*, 3–9
SISR register
  displaying • *System Dump Analyzer*, SDA–90
Site-specific startup
  procedure
    See SYS$MANAGER:SYSTARTUP.COM
Size
  allocating pages for PPL$ data structures • *RTL Parallel Processing*, PPL–11
  NCS library, specifying • *National Char Set*, NCS–24, NCS–25
SIZE attribute • *File Def Language*, FDL–35
/SIZE qualifier • *Debugger*, CD–73
SIZE secondary attribute • *File Applications*, 4–29
Skip file function • *I/O User's I*, 6–20
Skip sectoring • *I/O User's I*, 3–17
SKPC (Skip Character) instruction • *MACRO*, 9–139
Slash (/)
  division operator • *Debugger*, D–7
Slave formatter • *I/O User's I*, 6–8
SLEEP built-in procedure • *VAXTPU*, 7–508 to 7–509
Slider • *VAXTPU*, 7–224
  example of fetching • *VAXTPU*, B–19 to B–22
SLR register
  displaying • *System Dump Analyzer*, SDA–90
Small Computer System Interface
  See SCSI
Small request packet
  See SRP
SMB$CHECK_FOR_MESSAGE routine • *Utility Routines*, SMB–15
SMB$INITIALIZE routine • *Utility Routines*, SMB–16
SMB$READ_MESSAGE routine • *Utility Routines*, SMB–18
SMB$READ_MESSAGE_ITEM routine • *Utility Routines*, SMB–21
SMB$SEND_TO_JOBCTL routine • *Utility Routines*, SMB–31
SMB routines
  See also Job Controller
  See also Symbiont
  introduction • *Utility Routines*, SMB–1
SMG$
  debugging screen-oriented program • *Debugger*, 9–5
SMG$ADD_KEY_DEF • *Programming Resources*, 7–28; *RTL Screen Management*, 3–2, SMG–3
SMG$BEGIN_DISPLAY_UPDATE • *RTL Screen Management*, 2–18, SMG–7
SMG$BEGIN_PASTEBOARD_UPDATE • *RTL Screen Management*, 2–18, SMG–8
SMG$CANCEL_INPUT • *RTL Screen Management*, 1–7, 3–1, SMG–9
SMG$CHANGE_PBD_CHARACTERISTICS • *RTL Screen Management*, 1–5, SMG–10
SMG$CHANGE_RENDITION • *RTL Screen Management*, 2–9, SMG–13
SMG$CHANGE_VIEWPORT • *RTL Screen Management*, 2–13, SMG–16
SMG$CHANGE_VIRTUAL_DISPLAY • *Programming Resources*, 7–15; *RTL Screen Management*, 2–9, SMG–21
SMG$CHECK_FOR_OCCLUSION • *Programming Resources*, 7–12; *RTL Screen Management*, 2–5, SMG–24
SMG$CONTROL_MODE • *RTL Screen Management*, 2–16, SMG–28

SMG$COPY_VIRTUAL_DISPLAY • *RTL Screen Management*, SMG–31

SMG$CREATE_KEY_TABLE • *Programming Resources*, 7–28; *RTL Screen Management*, 3–2, SMG–36

SMG$CREATE_MENU • *RTL Screen Management*, 2–14, SMG–37

SMG$CREATE_PASTEBOARD • *Programming Resources*, 7–8; *RTL Screen Management*, 1–4, 6–2, SMG–41

SMG$CREATE_SUBPROCESS • *Programming Resources*, 7–16; *RTL Screen Management*, SMG–45

SMG$CREATE_VIEWPORT • *RTL Screen Management*, 2–13, SMG–58

SMG$CREATE_VIRTUAL_DISPLAY • *Programming Resources*, 7–8; *RTL Screen Management*, 1–6, SMG–49

SMG$CREATE_VIRTUAL_KEYBOARD • *Programming Resources*, 7–24; *RTL Screen Management*, 1–7, 3–1, SMG–54

SMG$CURSOR_COLUMN • *RTL Screen Management*, 2–6, SMG–62

SMG$CURSOR_ROW • *RTL Screen Management*, 2–6, SMG–63

SMG$DEFINE_KEY • *RTL Screen Management*, 3–2, SMG–64

SMG$DELETE_CHARS • *Programming Resources*, 7–22; *RTL Screen Management*, 2–7, SMG–67

SMG$DELETE_KEY_DEF • *RTL Screen Management*, 3–2, SMG–71

SMG$DELETE_LINE • *Programming Resources*, 7–22; *RTL Screen Management*, 2–7, SMG–73

SMG$DELETE_MENU • *RTL Screen Management*, 2–14, SMG–77

SMG$DELETE_PASTEBOARD • *Programming Resources*, 7–9; *RTL Screen Management*, 1–4, SMG–78

SMG$DELETE_SUBPROCESS • *Programming Resources*, 7–16; *RTL Screen Management*, SMG–80

SMG$DELETE_VIEWPORT • *RTL Screen Management*, 2–13, SMG–81

SMG$DELETE_VIRTUAL_DISPLAY • *Programming Resources*, 7–14; *RTL Screen Management*, 1–6, 2–4, 6–1, SMG–82

SMG$DELETE_VIRTUAL_KEYBOARD • *RTL Screen Management*, 3–1, SMG–83

SMG$DEL_TERM_TABLE • *RTL Screen Management*, 5–2, SMG–66

SMG$DISABLE_BROADCAST_TRAPPING • *RTL Screen Management*, SMG–84

SMG$DISABLE_UNSOLICITED_INPUT • *RTL Screen Management*, SMG–94

SMG$DRAW_CHAR • *RTL Screen Management*, 2–11, SMG–96

SMG$DRAW_LINE • *Programming Resources*, 7–20; *RTL Screen Management*, 2–11, SMG–100

SMG$DRAW_RECTANGLE • *Programming Resources*, 7–20; *RTL Screen Management*, 2–11, SMG–105

SMG$ENABLE_UNSOLICITED_INPUT • *RTL Screen Management*, 4–2, SMG–110

SMG$END_DISPLAY_UPDATE • *RTL Screen Management*, 2–18, SMG–113

SMG$END_PASTEBOARD_UPDATE • *RTL Screen Management*, 2–18, SMG–114

SMG$ERASE_CHARS • *Programming Resources*, 7–21; *RTL Screen Management*, 2–8, SMG–116

SMG$ERASE_COLUMN • *Programming Resources*, 7–22; *RTL Screen Management*, 2–8, SMG–120

SMG$ERASE_DISPLAY • *Programming Resources*, 7–21; *RTL Screen Management*, 2–8, SMG–122

SMG$ERASE_LINE • *Programming Resources*, 7–21; *RTL Screen Management*, 2–8, SMG–126

SMG$ERASE_PASTEBOARD • *Programming Resources*, 7–9; *RTL Screen Management*, 1–5, SMG–130

SMG$EXECUTE_COMMAND • *Programming Resources*, 7–16; *RTL Screen Management*, SMG–133

SMG$FIND_CURSOR_DISPLAY • *RTL Screen Management*, SMG–136

SMG$FLUSH_BUFFER • *RTL Screen Management*, 2–17, SMG–138

SMG$GET_BROADCAST_MESSAGE • *RTL Screen Management*, 4–1, SMG–139

SMG$GET_CHAR_AT_PHYSICAL_CURSOR • *RTL Screen Management*, SMG–141

SMG$GET_DISPLAY_ATTR • *RTL Screen Management*, SMG–143

SMG$GET_KEYBOARD_ATTRIBUTES • *RTL Screen Management*, 3–1, SMG–149

SMG$GET_KEY_DEF • *RTL Screen Management*, SMG–146

SMG$GET_NUMERIC_DATA • *RTL Screen Management*, 5–2, SMG–152

SMG$GET_PASTEBOARD_ATTRIBUTES • *RTL Screen Management*, 1–5, SMG–154

SMG$GET_PASTING_INFO • *RTL Screen Management*, SMG–158

SMG$GET_TERM_DATA • *RTL Screen Management*, 5–2, SMG–160

SMG$GET_VIEWPORT_CHAR • *RTL Screen Management*, 2–14, SMG–162

SMG$HOME_CURSOR • *Programming Resources*, 7–17; *RTL Screen Management*, 2–7, SMG–166

SMG$INIT_TERM_TABLE • *RTL Screen Management*, 5–2, SMG–168

SMG$INIT_TERM_TABLE_BY_TYPE • *RTL Screen Management*, 5–2, SMG–170

SMG$INSERT_CHARS • *Programming Resources*, 7–18; *RTL Screen Management*, 2–8, SMG–172

SMG$INSERT_LINE • *Programming Resources*, 7–20; *RTL Screen Management*, 2–8, SMG–177

SMG$INVALIDATE_DISPLAY • *RTL Screen Management*, SMG–183

SMG$KEYCODE_TO_NAME • *RTL Screen Management*, 3–4, SMG–184

SMG$LABEL_BORDER • *Programming Resources*, 7–10; *RTL Screen Management*, SMG–186

SMG$LIST_KEY_DEFS • *RTL Screen Management*, SMG–192

SMG$LIST_PASTING_ORDER • *Programming Resources*, 7–14; *RTL Screen Management*, 2–5, SMG–195

SMG$LOAD_KEY_DEFS • *RTL Screen Management*, 3–2, SMG–197

SMG$LOAD_VIRTUAL_DISPLAY • *RTL Screen Management*, 2–15, SMG–199

SMG$MOVE_TEXT • *RTL Screen Management*, 2–4, SMG–201

SMG$MOVE_VIRTUAL_DISPLAY • *RTL Screen Management*, 2–3, SMG–204

SMG$NAME_TO_KEYCODE • *RTL Screen Management*, 3–4, SMG–207

SMG$PASTE_VIRTUAL_DISPLAY • *Programming Resources*, 7–8; *RTL Screen Management*, 2–1, SMG–209

SMG$POP_VIRTUAL_DISPLAY • *Programming Resources*, 7–32; *RTL Screen Management*, 2–4, 6–2, SMG–212

SMG$PRINT_PASTEBOARD • *RTL Screen Management*, SMG–214

SMG$PUT_CHARS • *RTL Screen Management*, 2–8, SMG–216

SMG$PUT_CHARS_HIGHWIDE • *Programming Resources*, 7–19; *RTL Screen Management*, 2–8, SMG–221

SMG$PUT_CHARS_MULTI • *RTL Screen Management*, 2–8, SMG–224

SMG$PUT_CHARS_WIDE • *RTL Screen Management*, 2–8, SMG–227

SMG$PUT_HELP_TEXT • *RTL Screen Management*, SMG–230

SMG$PUT_LINE • *Programming Resources*, 7–19; *RTL Screen Management*, 2–9, SMG–233

SMG$PUT_LINE_HIGHWIDE • *RTL Screen Management*, 2–9, SMG–240

SMG$PUT_LINE_MULTI • *RTL Screen Management*, 2–9, SMG–244

SMG$PUT_LINE_WIDE • *Programming Resources*, 7–20; *RTL Screen Management*, 2–9, SMG–249

SMG$PUT_PASTEBOARD • *RTL Screen Management*, SMG–254

SMG$PUT_STATUS_LINE • *RTL Screen Management*, SMG–256

SMG$PUT_WITH_SCROLL • *Programming Resources*, 7–19

SMG$READ_COMPOSED_LINE • *Programming Resources*, 7–28; *RTL Screen Management*, 1–7, 3–2, SMG–258

SMG$READ_FROM_DISPLAY • *Programming Resources*, 7–23; *RTL Screen Management*, 2–12, SMG–263

SMG$READ_KEYSTROKE • *RTL Screen Management*, 3–1, SMG–267

SMG$READ_STRING • *Programming Resources*, 7–24; *RTL Screen Management*, 1–7, 3–1, SMG–275

SMG$READ_VERIFY • *RTL Screen Management*, 3–1, SMG–285

SMG$REMOVE_LINE • *RTL Screen Management*, 2–11, SMG–292

SMG$REPAINT_LINE • *RTL Screen Management*, SMG–294

SMG$REPAINT_SCREEN • *RTL Screen Management*, SMG–296

SMG$REPASTE_VIRTUAL_DISPLAY • *RTL Screen Management*, 2–3, SMG–299

SMG$REPLACE_INPUT_LINE • *RTL Screen Management*, SMG–304

SMG$RESTORE_PHYSICAL_SCREEN • *Programming Resources*, 7–31; *RTL Screen Management*, 6–3, SMG–307

SMG$RETURN_CURSOR_POS • *Programming Resources*, 7–18; *RTL Screen Management*, 2–6, SMG–309

SMG$RETURN_INPUT_LINE • *RTL Screen Management*, SMG–311

SMG$RING_BELL • *RTL Screen Management*, SMG–315

SMG$SAVE_PHYSICAL_SCREEN • *Programming Resources*, 7–31; *RTL Screen Management*, 6–3, SMG–316

SMG$SAVE_VIRTUAL_DISPLAY • *RTL Screen Management*, 2–15, SMG–318

SMG$SCROLL_DISPLAY_AREA • *Programming Resources*, 7–20; *RTL Screen Management*, SMG–320

SMG$SCROLL_VIEWPORT • *RTL Screen Management*, 2–13, SMG–323

SMG$SELECT_FROM_MENU • *RTL Screen Management*, 2–15, SMG–328

SMG$SET_BROADCAST_TRAPPING • *RTL Screen Management*, 4–1, SMG–343

SMG$SET_CURSOR_ABS • *Programming Resources*, 7–17; *RTL Screen Management*, 2–7, SMG–345

SMG$SET_CURSOR_MODE • *RTL Screen Management*, SMG–347

SMG$SET_CURSOR_REL • *Programming Resources*, 7–17; *RTL Screen Management*, 2–7, SMG–349

SMG$SET_DEFAULT_STATE • *RTL Screen Management*, SMG–351

SMG$SET_DISPLAY_SCROLLING_REGION • *RTL Screen Management*, SMG–353

SMG$SET_DISPLAY_SCROLL_REGION • *Programming Resources*, 7–20

SMG$SET_KEYPAD_MODE • *RTL Screen Management*, 3–2, SMG–355

SMG$SET_OUT_OF_BAND_ASTS • *RTL Screen Management*, 4–2, SMG–357

SMG$SET_PHYSICAL_CURSOR • *Programming Resources*, 7–18; *RTL Screen Management*, SMG–361

SMG$SET_TERM_CHARACTERISTICS • *RTL Screen Management*, SMG–363

SMG$SNAPSHOT • *RTL Screen Management*, SMG–367

SMG$UNPASTE_VIRTUAL_DISPLAY • *Programming Resources*, 7–14; *RTL Screen Management*, 2–1, 6–1, SMG–369

SMP$ACQNOIPL • *Device Support (A)*, 13–29, E–18; *Device Support (B)*, 2–17

SMP$ACQUIRE • *Device Support (A)*, 13–28, 13–29, E–18; *Device Support (B)*, 2–34, 2–47

SMP$ACQUIREL • *Device Support (A)*, 13–28, 13–29, E–18; *Device Support (B)*, 2–17

SMP$AR_IPLVEC • *Device Support (B)*, 2–33, 3–26, 3–30

SMP$AR_SPNLKVEC • *Device Support (A)*, 3–13; *Device Support (B)*, 1–66, 2–34, 2–47, 2–96

SMP$GL_FLAGS • *Device Support (A)*, 12–13, E–3

SMP$RELEASE • *Device Support (A)*, 13–28, 13–29, E–18; *Device Support (B)*, 2–35, 2–96

SMP$RELEASEL • *Device Support (A)*, 13–28, 13–29, E–18; *Device Support (B)*, 2–19

SMP$RESTORE • *Device Support (A)*, 13–28, 13–29, E–18; *Device Support (B)*, 2–35, 2–96

SMP$RESTOREL • *Device Support (A)*, 13–28, 13–29, E–18; *Device Support (B)*, 2–19

SMP$V_UNMOD_DRIVER • *Device Support (A)*, 12–13, E–3

SOBGEQ (Subtract One and Branch Greater Than or Equal) instruction • *MACRO*, 9–61

SOBGTR (Subtract One and Branch Greater Than) instruction • *MACRO*, 9–62

SOFTINT macro • *Device Support (A)*, 3–10; *Device Support (B)*, 2–67, 3–26, 3–30

Software error • *File Applications*, 10–1

Software life cycle • *Modular Procedures*, 1–1

Software Performance Report
  See SPR

Software timer interrupt service routine • *Device Support (A)*, 3–8, 10–4

Solicited interrupt
  See Device interrupt

SOR$$STAT routine • *Utility Routines*, SOR–50

SOR$BEGIN_MERGE • *Programming Resources*, 8–19

SOR$BEGIN_MERGE routine • *Utility Routines*, SOR–18

SOR$BEGIN_SORT • *Programming Resources*, 8–15

SOR$BEGIN_SORT routine • *Utility Routines*, SOR–25

SOR$DTYPE routine • *Utility Routines*, SOR–31

SOR$END_SORT • *Programming Resources*, 8–15

SOR$END_SORT routine • *Utility Routines*, SOR–34

SOR$PASS_FILES • *Programming Resources*, 8–15, 8–19

SOR$PASS_FILES routine • *Utility Routines*, SOR–36

SOR$RELEASE_REC • *Programming Resources*, 8–16

SOR$RELEASE_REC routine • *Utility Routines*, SOR–41

SOR$RETURN_REC • *Programming Resources*, 8–16

SOR$RETURN_REC routine • *Utility Routines*, SOR–43

SOR$SORT_MERGE • *Programming Resources*, 8–15

SOR$SORT_MERGE routine • *Utility Routines*, SOR–45

SOR$SPEC_FILE routine • *Utility Routines*, SOR–48

SOR routines
  examples • *Utility Routines*, SOR–4 to SOR–17
  interface
    file • *Utility Routines*, SOR–2
    record • *Utility Routines*, SOR–2
  introduction • *Utility Routines*, SOR–1
  list of • *Utility Routines*, SOR–1

SOR routines (Cont.)
  reentrancy
    using context argument • *Utility Routines*, SOR–4
Sort
  suggestions for improving performance • *Convert*, CONV–22
SORT
  See Sort/Merge Utility
SORT32
  open file limitation • *Convert*, CONV–22
SORT command • *Programming Resources*, 8–13
  file interface • *Programming Resources*, 8–15
  record interface • *Programming Resources*, 8–16
Sort/Merge routines
  See SOR routines
Sort/Merge Utility (SORT) • *Programming Resources*, 8–13
  file interface • *Programming Resources*, 8–14, 8–15, 8–19
  keys • *Programming Resources*, 8–14
  multiple sort operations • *Programming Resources*, 8–14
  record interface • *Programming Resources*, 8–14, 8–16, 8–21
Sort order
  establishing • *RMS*, 7–5
/SORT qualifier • *Convert*, CONV–22, CONV–27
SOURCE attribute • *File Def Language*, FDL–38
Source code
  See Source display
Source Code Analyzer
  See SCA
Source directory
  displaying • *Debugger*, 6–2, CD–246
  search list • *Debugger*, 6–2, CD–28, CD–176
Source display • *Debugger*, 2–10, 6–1, 7–1
  discrepancies in • *Debugger*, 7–4, 9–1
    with DECwindows • *Debugger*, 1–9
  display kind • *Debugger*, 7–18, C–1
  EXAMINE/SOURCE command • *Debugger*, 6–4, 7–6, 7–18, C–4
  for routine on call stack • *Debugger*, 7–6, CD–170
    with DECwindows • *Debugger*, 1–8, 1–9, 1–21
  line-oriented • *Debugger*, 6–3
  margins in • *Debugger*, 6–9, CD–228
  multiprocess program • *Debugger*, 10–16
  not available • *Debugger*, 2–11, 2–12, 6–1, 7–4, CD–176, C–4
    with DECwindows • *Debugger*, 1–9, 1–21
  optimized code • *Debugger*, 2–5, 5–2, 7–8, 9–1
    with DECwindows • *Debugger*, 1–9
Source display (Cont.)
  SEARCH command • *Debugger*, 6–6, CD–120
  SET BREAK command • *Debugger*, 6–7
  SET SCOPE/CURRENT command • *Debugger*, 7–6, CD–170
  SET STEP command • *Debugger*, 6–7, CD–179
  SET TRACE command • *Debugger*, 6–7
  SET WATCH command • *Debugger*, 6–7
  SRC, predefined • *Debugger*, 7–4, C–4
    with DECwindows • *Debugger*, 1–9
  STEP command • *Debugger*, 6–7
  TYPE command • *Debugger*, 6–3, CD–275
  with DECwindows • *Debugger*, 1–8, 1–9, 1–21
Source file
  See also Message source file
  See also Source display
  correct version of • *Debugger*, CD–176, CD–246
  defined • *Debugger*, 6–2; *VAXTPU*, 7–308
  EVE editor • *VAXTPU*, 1–11
  file specification • *Debugger*, 6–2
  location • *Debugger*, 6–2, CD–28, CD–176, CD–246
  maximum number • *Debugger*, 6–3, CD–151, CD–229
  not available • *Debugger*, 6–2, CD–176
Source file statements
  See Message source file statements
Source line • *File Def Language*, FDL–40
Source line correlation • *Debugger*, 6–1
/SOURCE qualifier • *Debugger*, 6–4, 6–8, 7–6, 7–21, CD–88, CD–124, CD–134, CD–190, CD–201, CD–267
Source statement
  See Statement
Source window
  See also Source display
  SRC, DECwindows • *Debugger*, 1–9, 1–21
%SOURCE_SCOPE • *Debugger*, 7–19, C–4
%SP • *Debugger*, 4–23, D–3
Space
  allocating for PPL$ • *RTL Parallel Processing*, PPL–11
Space service • *RMS*, RMS–95
  condition values • *RMS*, RMS–96
  control block input fields • *RMS*, RMS–96
  control block output fields • *RMS*, RMS–96
SPAN built-in procedure • *VAXTPU*, 7–510 to 7–511
SPANC (Span Characters) instruction • *MACRO*, 9–140
SPANL built-in procedure • *VAXTPU*, 7–512 to 7–514
SPAWN built-in procedure • *VAXTPU*, 7–515 to 7–517

SPAWN command • *Debugger,* 3–5, CD–263; *System Dump Analyzer,* SDA–162

Spawned subprocess
  See Subprocess

Spawning a subordinate • *RTL Parallel Processing,* 2–3

SPDT (SCSI port descriptor table) • *Device Support (A),* 17–7; *Device Support (B),* 1–60 to 1–66
  creation of • *Device Support (A),* 17–26

Special analysis sections • *Analyze/RMS_File,* ARMS–14

Special characters • *Librarian,* LIB–5

SPECIAL_GRAPHICS keyword
  with SET (STATUS_LINE) • *VAXTPU,* 7–476

"Special_graphics_status" string constant parameter to GET_INFO • *VAXTPU,* 7–225

Specification
  of file • *File Def Language,* FDL–19

Speed
  See Performance

SPI$ABORT_COMMAND macro • *Device Support (A),* 17–6, 17–28; *Device Support (B),* 2–68

SPI$ALLOCATE_COMMAND_BUFFER macro • *Device Support (A),* 17–6, 17–11, 17–27; *Device Support (B),* 2–69

SPI$CONNECT macro • *Device Support (A),* 17–6, 17–10, 17–26, 17–29; *Device Support (B),* 2–70 to 2–71

SPI$DEALLOCATE_COMMAND_BUFFER macro • *Device Support (A),* 17–6, 17–11, 17–28; *Device Support (B),* 2–72

SPI$DISCONNECT macro • *Device Support (A),* 17–6; *Device Support (B),* 2–73

SPI$FINISH_COMMAND macro • *Device Support (A),* 17–29; *Device Support (B),* 2–74

SPI$GET_CONNECTION_CHAR macro • *Device Support (A),* 17–6; *Device Support (B),* 2–75 to 2–76, 2–88

SPI$MAP_BUFFER macro • *Device Support (A),* 17–6, 17–16 to 17–17, 17–27; *Device Support (B),* 2–77 to 2–79

SPI$RECEIVE_BYTES macro • *Device Support (A),* 17–29; *Device Support (B),* 2–80

SPI$RELEASE_BUS macro • *Device Support (A),* 17–29; *Device Support (B),* 2–81

SPI$RESET macro • *Device Support (A),* 17–6

SPI$SEND_BYTES macro • *Device Support (A),* 17–29; *Device Support (B),* 2–83

SPI$SEND_COMMAND macro • *Device Support (A),* 17–6, 17–11, 17–17, 17–27; *Device Support (B),* 2–84 to 2–86

SPI$SENSE_PHASE macro • *Device Support (A),* 17–29; *Device Support (B),* 2–87

SPI$SET_CONNECTION_CHAR macro • *Device Support (A),* 17–6, 17–12, 17–13, 17–14, 17–27; *Device Support (B),* 2–88 to 2–89

SPI$SET_PHASE macro • *Device Support (A),* 17–29; *Device Support (B),* 2–90

SPI$UNMAP_BUFFER macro • *Device Support (A),* 17–6, 17–17; *Device Support (B),* 2–91

SPI (SCSI port interface) • *Device Support (A),* 17–5 to 17–6; *Device Support (B),* 2–68 to 2–90
  calling protocol for • *Device Support (A),* 17–6; *Device Support (B),* 2–68
  extensions to • *Device Support (A),* 17–29 to 17–30; *Device Support (B),* 2–73 to 2–90

Spin lock • *Programming Resources,* 4–16; *Device Support (A),* 1–7, 3–3, 3–12 to 3–17
  See also Device lock
  See also Fork lock
  See also Spin lock index
  See also Spin wait
  See also SPL
  See also Synchronization
  acquisition IPL • *Device Support (A),* 3–11, 3–15, E–17, E–20; *Device Support (B),* 1–67, 3–111
  acquisition PC list • *Device Support (A),* E–17; *Device Support (B),* 1–68
  address • *Device Support (A),* E–20
  creating • *RTL Parallel Processing,* 4–14
  definition of • *RTL Parallel Processing,* 4–14
  deleting • *RTL Parallel Processing,* 4–15
  displaying SDA information • *System Dump Analyzer,* SDA–150
  dynamic • *Device Support (A),* 3–13; *Device Support (B),* 1–68
  multiple acquisition of • *Device Support (A),* 3–15, E–20; *Device Support (B),* 2–96, 3–116
  name • *Device Support (A),* E–20
  obtaining • *Device Support (A),* 3–10; *Device Support (B),* 2–47 to 2–48, 3–111 to 3–112
  owned • *System Dump Analyzer,* SDA–90
  ownership • *Device Support (A),* 3–15, 13–30, E–20; *Device Support (B),* 1–67, 1–68
  rank • *Device Support (A),* 3–13 to 3–14, 3–15, 3–17, E–17, E–20; *Device Support (B),* 1–67
  reading • *RTL Parallel Processing,* 4–16
  releasing • *RTL Parallel Processing,* 4–15; *Device Support (A),* 3–10; *Device Support (B),* 2–96, 3–114
  restoring • *Device Support (B),* 2–96, 3–116
  seizing • *RTL Parallel Processing,* 4–15
  static • *Device Support (A),* 3–13; *Device Support (B),* 1–68

Spin lock (Cont.)
status • *Device Support (A),* E–20
system • *Device Support (A),* 3–13; *Device Support (B),* 1–68
Spin lock index • *Device Support (A),* 3–13, 3–13 to 3–14, E–20
Spin lock IPL vector
See SMP$AR_SPNLKVEC
Spin lock synchronization
advantages and disadvantages • *RTL Parallel Processing,* 5–8
PPL$ routines for • *RTL Parallel Processing,* 4–14 to 4–16
Spin lock synchronization macros • *Device Support (A),* E–4, E–13
See also DEVICELOCK
See also DEVICEUNLOCK
See also FORKLOCK
See also FORKUNLOCK
See also LOCK
See also UNLOCK
Spin wait • *Device Support (A),* 3–15; *Device Support (B),* 1–68, 3–110, 3–112, 3–113
SPL$B_IPL • *Device Support (A),* 3–9, E–18; *Device Support (B),* 1–77
SPL$B_RANK • *Device Support (A),* E–18
SPL$L_BUSY_WAITS • *Device Support (A),* E–17
SPL$L_OWN_PC_VEC • *Device Support (A),* E–17
SPL$Q_ACQ_COUNT • *Device Support (A),* E–17
SPL (spin lock data structure) • *Device Support (B),* 1–66 to 1–68
SPLACQERR bugcheck • *Device Support (A),* 13–28, 13–30, E–18; *Device Support (B),* 3–111
$SPLCODDEF macro • *Device Support (A),* E–8; *Device Support (B),* 2–23, 2–25
SPLIPLHIGH bugcheck • *Device Support (A),* 13–28, E–18; *Device Support (B),* 3–111, 3–113
SPLIPLLOW bugcheck • *Device Support (A),* 13–28, E–18; *Device Support (B),* 3–114, 3–115, 3–116, 3–117
SPLIT_LINE built-in procedure • *VAXTPU,* 7–518 to 7–519
SPL option • *File Def Language,* FDL–23
SPLRELERR bugcheck • *Device Support (A),* 13–29, 13–30, E–18; *Device Support (B),* 3–114, 3–115
SPLRSTERR bugcheck • *Device Support (A),* 13–29, 13–30, E–18; *Device Support (B),* 3–116, 3–117
Spooled device • *Device Support (B),* 1–74
Spool file option
See FAB$V_SPL option
Spool on close option • *File Applications,* 9–12
SPR (Software Performance Report) • *File Applications,* 10–2; *System Dump Analyzer,* SDA–2, SDA–28
SP symbol • *System Dump Analyzer,* SDA–14
SPTREQ parameter • *Device Support (B),* 3–16
SQO option • *File Def Language,* FDL–24
Square root • *RTL Math,* MTH–102, MTH–136
/SQUEEZE qualifier • *Librarian,* LIB–43
SRC
source display, screen mode • *Debugger,* 7–4, C–4
source window, DECwindows • *Debugger,* 1–9, 1–21
SRP (small request packet) • *System Dump Analyzer,* SDA–119
SRP lookaside list
displaying contents • *System Dump Analyzer,* SDA–119
/SRP qualifier • *System Dump Analyzer,* SDA–119
SS$_ABORT • *I/O User's II,* 2–15, 4–23, 6–33, A–1, A–3, A–4, A–5, A–6; *Device Support (A),* 10–6
SS$_ABORT return • *I/O User's I,* 8–45, 8–50, A–2, A–3, A–5, A–7, A–9
SS$_ACCONFLICT return • *I/O User's I,* A–1
SS$_ACCVIO • *I/O User's II,* A–6; *Device Support (B),* 3–32, 3–33, 3–35, 3–41, 3–43, 3–46, 3–50, 3–51, 3–55, 3–56, 3–59, 3–73
SS$_ACCVIO return • *I/O User's I,* 7–12, 8–51
SS$_ACPVAFUL return • *I/O User's I,* A–1
SS$_BADATTRIB return • *I/O User's I,* A–1
SS$_BADCHKSUM return • *I/O User's I,* A–1
SS$_BADESCAPE return • *I/O User's I,* 8–7, A–9
SS$_BADFILEHDR return • *I/O User's I,* A–1
SS$_BADFILENAME return • *I/O User's I,* A–1
SS$_BADFILEVER return • *I/O User's I,* A–1
SS$_BADIRECTORY return • *I/O User's I,* A–1
SS$_BADPARAM • *I/O User's II,* 3–11, 4–22, 4–26, 4–27, 4–31, 6–9, 6–23, 6–35, A–1, A–3, A–4, A–5, A–6; *Device Support (B),* 3–32, 3–35, 3–41, 3–43, 3–46, 3–55, 3–56, 3–59, 3–107
SS$_BADPARAM return • *I/O User's I,* 8–51, A–1, A–5, A–9
SS$_BADQFILE return • *I/O User's I,* A–1
SS$_BADQUEHDR • *I/O User's II,* 4–33, A–4
SS$_BADQUEUEHDR • *I/O User's II,* 4–28, 4–31, 4–32
SS$_BLOCKCNTERR return • *I/O User's I,* A–1
SS$_BUFFEROVF • *I/O User's II,* 2–20, 5–10, 5–11, 6–38, A–3, A–5, A–6
SS$_BUFFEROVF return • *I/O User's I,* 7–6, A–7
SS$_BUFNOTALIGN • *I/O User's II,* 4–23, A–4
SS$_BUFNOTALIGN return • *I/O User's I,* A–5

SS$_CANCEL • *I/O User's II*, 4–23, A–3, A–4, A–5; *Device Support (A)*, 11–7

SS$_CANCEL return • *I/O User's I*, A–3, A–5, A–7, A–9

SS$_COMMHARD • *I/O User's II*, A–6

SS$_CONTROLC return • *I/O User's I*, 8–46, A–9

SS$_CONTROLO return • *I/O User's I*, A–9

SS$_CONTROLY return • *I/O User's I*, A–9

SS$_CREATED return • *I/O User's I*, A–1

SS$_CTRLERR • *I/O User's II*, 3–8, 4–23, 4–33, 4–36, A–3, A–4, A–6

SS$_CTRLERR return • *I/O User's I*, A–3, A–5, A–7

SS$_DATACHECK • *I/O User's II*, A–6

SS$_DATACHECK return • *I/O User's I*, A–3, A–5, A–7

SS$_DATAOVERUN • *I/O User's II*, 1–6, 2–8, 5–5, 6–19, A–1, A–6

SS$_DATAOVERUN return • *I/O User's I*, 8–9, A–2, A–3, A–7, A–9

SS$_DEBUG condition • *Debugger*, D–1

SS$_DEVACTIVE • *I/O User's II*, 4–20, A–1, A–3, A–4, A–5, A–6

SS$_DEVACTIVE return • *I/O User's I*, 8–50, A–5

SS$_DEVALLOC • *I/O User's II*, A–6

SS$_DEVCMDERR return • *I/O User's I*, A–5

SS$_DEVICEFULL • *I/O User's II*, A–3, A–5

SS$_DEVICEFULL return • *I/O User's I*, A–1

SS$_DEVINACT • *I/O User's II*, A–3, A–5, A–6

SS$_DEVOFFLINE • *I/O User's II*, A–1, A–3, A–5, A–6

SS$_DEVOFFLINE return • *I/O User's I*, A–7

SS$_DEVREQERR • *I/O User's II*, 4–23, 4–36, A–4, A–6

SS$_DEVREQERR return • *I/O User's I*, A–5

SS$_DIRFULL return • *I/O User's I*, A–1

SS$_DIRNOTEMPTY return • *I/O User's I*, A–1

SS$_DISCONNECT • *I/O User's II*, A–6

SS$_DRVERR • *I/O User's II*, 3–8, A–3

SS$_DRVERR return • *I/O User's I*, A–3, A–7

SS$_DUPDSKQUOTA return • *I/O User's I*, A–1

SS$_DUPFILENAME return • *I/O User's I*, A–1

SS$_DUPUNIT • *I/O User's II*, A–6

SS$_ENDOFFILE • *I/O User's II*, 2–8, 5–5, 6–19, A–1, A–6

SS$_ENDOFFILE return • *I/O User's I*, 6–21, 7–6, 7–9, A–1, A–2, A–7

SS$_ENDOFTAPE return • *I/O User's I*, A–7

SS$_ENDOFVOLUME return • *I/O User's I*, 6–21, A–7

SS$_EXBYTLM return • *I/O User's I*, A–1

SS$_EXDISKQUOTA return • *I/O User's I*, A–1

SS$_EXQUOTA • *I/O User's II*, 4–23, A–3, A–4, A–6; *Device Support (A)*, E–6; *Device Support (B)*, 3–6, 3–20, 3–22

SS$_EXQUOTA return • *I/O User's I*, A–5

SS$_FCPREADERR return • *I/O User's I*, A–1

SS$_FCPREWNDERR return • *I/O User's I*, A–1

SS$_FCPSPACERR return • *I/O User's I*, A–1

SS$_FCPWRITERR return • *I/O User's I*, A–1

SS$_FILELOCKED return • *I/O User's I*, A–1

SS$_FILENUMCHK return • *I/O User's I*, A–1

SS$_FILEPURGED return • *I/O User's I*, A–1

SS$_FILESEQCHK return • *I/O User's I*, A–1

SS$_FILESTRUCT return • *I/O User's I*, A–1

SS$_FILNOTEXP return • *I/O User's I*, A–1

SS$_FORCEDERR return • *I/O User's I*, A–3

SS$_FORMAT return • *I/O User's I*, A–3, A–7

SS$_HANGUP return • *I/O User's I*, 8–13

SS$_HEADERFULL return • *I/O User's I*, A–1

SS$_IBCERROR return • *I/O User's I*, A–1

SS$_IDXFILEFULL return • *I/O User's I*, A–1

SS$_ILLCNTRFUNC return • *I/O User's I*, A–1

SS$_ILLIOFUNC • *Device Support (B)*, 3–51

SS$_ILLIOFUNC return • *I/O User's I*, 8–50, A–3, A–7

SS$_INCOMPAT return • *I/O User's I*, A–9

SS$_INSFBUFDP return • *I/O User's I*, A–5

SS$_INSFMAPREG • *I/O User's II*, A–6; *Device Support (B)*, 3–64

SS$_INSFMAPREQ return • *I/O User's I*, A–5

SS$_INSFMEM • *I/O User's II*, 4–23, 4–28, 4–31, A–4, A–6; *Device Support (B)*, 3–6, 3–12, 3–14, 3–15, 3–16, 3–52, 3–61

SS$_INSFMEM return • *I/O User's I*, 7–12, A–5

SS$_INSFSPTS • *Device Support (B)*, 3–16, 3–107

SS$_INSFWSL • *Device Support (B)*, 3–33, 3–35, 3–41, 3–46, 3–59

SS$_IVADDR return • *I/O User's I*, A–3

SS$_IVBUFLEN • *I/O User's II*, 4–23, 6–21, A–4, A–6

SS$_IVBUFLEN return • *I/O User's I*, A–3, A–5

SS$_IVCHAN • *Device Support (B)*, 3–103

SS$_IVMODE return • *I/O User's I*, A–5

SS$_MBFULL • *Device Support (B)*, 3–52, 3–61

SS$_MBFULL return • *I/O User's I*, 7–2, 7–7, 7–12

SS$_MBTOOSML • *Device Support (B)*, 3–52, 3–61

SS$_MBTOOSML return • *I/O User's I*, 7–12

SS$_MCNOTVALID • *I/O User's II*, 4–23, A–4

SS$_MCNOTVALID return • *I/O User's I*, A–5

SS$_MEDOFL • *I/O User's II*, A–6

SS$_MEDOFL return • *I/O User's I*, A–3, A–7

SS$_NODISKQUOTA return • *I/O User's I*, A–1

SS$_NOMOREFILES return • *I/O User's I*, A–1

SS$_NONEXDRV return • *I/O User's I*, A–3, A–7

SS$_NONSMPDRV • *Device Support (A),* E–4
SS$_NOPRIV • *I/O User's II,* A–3, A–6; *Device Support (B),* 3–52, 3–61, 3–103
SS$_NOPRIV return • *I/O User's I,* 7–12, 8–51, A–1
SS$_NOQFILE return • *I/O User's I,* A–1
SS$_NORMAL • *I/O User's II,* 4–23, A–1, A–3, A–4, A–5, A–6
SS$_NORMAL return • *I/O User's I,* 8–50, 8–51, A–2, A–3, A–7, A–9
SS$_NOSUCHFILE return • *I/O User's I,* A–1
SS$_NOTAPEOP return • *I/O User's I,* A–2
SS$_NOTLABELMT return • *I/O User's I,* A–2
SS$_NOTPRINTED return • *I/O User's I,* A–2
SS$_NOTVOLSET return • *I/O User's I,* A–2
SS$_OPINCOMPL • *I/O User's II,* 3–12, 6–33, A–3, A–6
SS$_OPINCOMPL return • *I/O User's I,* A–3, A–7
SS$_OVRDSKQUOTA return • *I/O User's I,* A–2
SS$_PARITY • *I/O User's II,* 4–20, 4–23, 4–36, A–3, A–4
SS$_PARITY return • *I/O User's I,* A–3, A–5, A–7, A–9
SS$_PARTESCAPE return • *I/O User's I,* 8–7, 8–30, A–9
SS$_POWERFAIL • *I/O User's II,* 4–3, 4–20, 4–23, A–4
SS$_POWERFAIL return • *I/O User's I,* A–5
SS$_QFACTIVE return • *I/O User's I,* A–2
SS$_QFNOTACT return • *I/O User's I,* A–2
SS$_RCT return • *I/O User's I,* A–3
SS$_RDDELDATA return • *I/O User's I,* A–3
SS$_SERIOUSEXCP return • *I/O User's I,* A–2, A–7
SS$_SSFAIL • *Device Support (B),* 3–64, 3–75, 3–85, 3–93
SS$_SUPERSEDE return • *I/O User's I,* A–2
SS$_TAPEPOSLOST return • *I/O User's I,* A–2
SS$_TIMEOUT • *I/O User's II,* 6–33, A–3, A–6
SS$_TIMEOUT return • *I/O User's I,* 8–27, 8–50, A–3, A–5, A–7, A–9
SS$_TOOMANYVER return • *I/O User's I,* A–2
SS$_TOOMUCHDATA • *I/O User's II,* A–6
SS$_UNSAFE return • *I/O User's I,* A–3, A–7
SS$_VOLINV return • *I/O User's I,* A–3, A–7
SS$_WASECC return • *I/O User's I,* A–3
SS$_WRITLCK return • *I/O User's I,* A–2, A–3, A–7
SS$_WRONGACP return • *I/O User's I,* A–2
SSP symbol • *System Dump Analyzer,* SDA–14
SSRVEXCEPT bugcheck • *System Dump Analyzer,* SDA–16
Stack
  See also Call frame
  See also Call stack
Stack (Cont.)
  See also Scope
  device driver use of • *Device Support (A),* 8–1
  displaying contents • *System Dump Analyzer,* SDA–157
  using for temporary storage • *Device Support (A),* 5–3
  variable • *Debugger,* 3–19, 4–1
    with DECwindows • *Debugger,* 1–25
Stack frame • *MACRO,* 9–64
  displaying in SDA • *System Dump Analyzer,* SDA–79
  following a chain • *System Dump Analyzer,* SDA–79
Stack limit
  changing size of • *System Services,* SYS–427
Stack pointer
  adjusting • *System Services,* SYS–8
Stack pointer symbol • *Delta/XDelta,* DELTA–9, DELTA–13
Stack usage • *Routines Intro,* 2–14, 2–45
Standard Disk Interconnect (SDI) • *I/O User's I,* 3–5
STARLET.OLB • *Programming Resources,* 5–1, 5–12
  See also SYS$LIBRARY:STARLET.OLB
Start I/O routine • *System Dump Analyzer,* SDA–99; *Device Support (A),* 1–3
  See also Alternate start I/O routine
  activating • *Device Support (B),* 3–28
  address • *Device Support (A),* 2–4, 6–4; *Device Support (B),* 1–30, 4–17
  checking for zero length buffer • *Device Support (B),* 3–32, 3–41, 3–55
  context • *Device Support (A),* 4–15, 8–1 to 8–2; *Device Support (B),* 4–17
  entry point • *Device Support (B),* 4–17
  exit method • *Device Support (B),* 4–18
  for connect to interrupt facility • *Device Support (A),* 19–10, 19–15 to 19–16
  for MASSBUS device • *Device Support (A),* 15–13
  functions • *Device Support (A),* 4–15 to 4–16
  input • *Device Support (B),* 4–17
  of CONINTERR.EXE • *Device Support (A),* 19–13
  of third-party SCSI class driver • *Device Support (A),* 17–27 to 17–28
  reactivating • *Device Support (A),* 4–18
  register usage • *Device Support (A),* 8–1; *Device Support (B),* 4–17
  suspending • *Device Support (A),* 4–16
  synchronization requirements • *Device Support (A),* 3–6, 3–22, 8–5, E–9 to E–11; *Device Support (B),* 4–17

Start I/O routine (Cont.)
transferring control to • *Device Support (A),* 4–13 to 4–15, 8–1, 10–3; *Device Support (B),* 3–38, 3–70 to 3–71
writing • *Device Support (A),* 8–1 to 8–7
Starting key position • *File Def Language,* FDL–29
Starting logical block number field
See XAB$L_SBN field
Startup file • *VAXTPU,* 1–10 to 1–11, 4–21 to 4–33
command file • *VAXTPU,* 1–10
definition • *VAXTPU,* 1–10
initialization file • *VAXTPU,* 1–10
order of execution • *VAXTPU,* 4–22
section file • *VAXTPU,* 1–10
"Start_character" string constant parameter to GET_INFO • *VAXTPU,* 7–178
/START_POSITION qualifier • *Debugger,* CD–139; *VAXTPU,* 5–17
"Start_record" string constant parameter to GET_INFO • *VAXTPU,* 7–178
State • *RTL Screen Management,* 3–3
Statement • *MACRO,* 1–1
character set • *MACRO,* 3–1
comment • *MACRO,* 2–3
continuation of • *MACRO,* 2–1
for command definition file • *Command Def,* CDU–19 to CDU–37
format • *MACRO,* 2–1
label • *MACRO,* 2–2
operand • *MACRO,* 2–3
operator • *MACRO,* 2–3, C–7
separator for • *VAXTPU,* 4–3
special characters • *MACRO,* C–6
STAT entry point • *Modular Procedures,* 4–9
/STATE qualifier • *Debugger,* 8–9, CD–60, CD–144, CD–224, CD–254
/STATIC qualifier • *Debugger,* CD–201
Static selection • *VAXTPU,* 4–17
Static spin lock • *Device Support (A),* 3–13
Static variable • *Debugger,* 3–19, 4–1
Statistical report • *Analyze/RMS_File,* ARMS–10
Statistics
produced by CONVERT • *Convert,* CONV–24
produced by CONVERT/RECLAIM • *Convert,* CONV–24
/STATISTICS qualifier • *Debugger,* CD–254; *File Applications,* 10–6
description • *Analyze/RMS_File,* ARMS–19
example of • *Analyze/RMS_File,* ARMS–16
format • *Analyze/RMS_File,* ARMS–19
limitation • *Analyze/RMS_File,* ARMS–13, ARMS–14, ARMS–19, ARMS–20

/STATISTICS qualifier (Cont.)
overview • *Analyze/RMS_File,* ARMS–19
using with /OUTPUT qualifier • *Analyze/RMS_File,* ARMS–16
with CONVERT/RECLAIM • *Convert,* CONV–5, CONV–24
with wildcard characters • *Analyze/RMS_File,* ARMS–10
Statistics report • *File Applications,* 10–6, 10–11
Status
See Port
See SCSI command
See SCSI status byte
Status line
default information • *VAXTPU,* 7–77
fields added with EVE$BUILD • *VAXTPU,* G–7 to G–8
video attributes • *VAXTPU,* 7–476
Status register
See CSR
See MBA$L_SR
STATUS_LINE keyword • *VAXTPU,* 7–476
"Status_line" string constant parameter to GET_INFO • *VAXTPU,* 7–225
"Status_video" string constant parameter to GET_INFO • *VAXTPU,* 7–225
Step button
with DECwindows • *Debugger,* 1–9
STEP command • *Debugger,* 3–7, 6–7, CD–265
displaying default qualifiers for • *Debugger,* CD–249
multiprocess program • *Debugger,* 10–6
setting default qualifiers for • *Debugger,* CD–179
vectorized program • *Debugger,* 11–3
with DECwindows • *Debugger,* 1–24
Step Instruction command • *Delta/XDelta,* DELTA–34
Step Instruction Over Subroutine command • *Delta/XDelta,* DELTA–35
Sticky default
defined • *File Applications,* 6–9
Stop button
with DECwindows • *Debugger,* 1–9, 1–21
STOP command • *Debugger,* 3–5
STOPPED processor state • *Device Support (B),* 1–16
STOPPING processor state • *Device Support (B),* 1–16
Storage • *Modular Procedures,* 2–12
heap • *Modular Procedures,* 2–12
initializing • *Modular Procedures,* 3–14
stack • *Modular Procedures,* 2–12
static • *Modular Procedures,* 2–13, A–5

Storage (Cont.)
summary • *Modular Procedures*, 2–15
STR$ADD • *RTL String Manipulation*, STR–3
STR$ANALYZE_SDESC • *RTL String Manipulation*, 2–4, STR–7
STR$APPEND • *RTL String Manipulation*, 2–9, STR–9
STR$CASE_BLIND_COMPARE • *RTL String Manipulation*, STR–11
STR$COMPARE • *RTL String Manipulation*, STR–13
STR$COMPARE_EQL • *RTL String Manipulation*, STR–15
STR$COMPARE_MULTI • *RTL String Manipulation*, STR–17
STR$CONCAT • *RTL String Manipulation*, 2–9, STR–20
STR$COPY_DX • *RTL String Manipulation*, 2–7, 2–8, STR–23
STR$COPY_R • *RTL String Manipulation*, STR–25
STR$DIVIDE • *RTL String Manipulation*, STR–28
STR$DUPL_CHAR • *RTL String Manipulation*, STR–32
STR$ELEMENT • *RTL String Manipulation*, STR–34
STR$FIND_FIRST_IN_SET • *RTL String Manipulation*, STR–36
STR$FIND_FIRST_NOT_IN_SET • *RTL String Manipulation*, STR–38
STR$FIND_FIRST_SUBSTRING • *RTL String Manipulation*, STR–41
STR$FREE1_DX • *RTL String Manipulation*, STR–45
STR$GET1_DX • *RTL String Manipulation*, STR–46
STR$GET1_DX routine • *File Applications*, 5–10
STR$LEFT • *RTL String Manipulation*, 2–9, STR–48
STR$LEN_EXTR • *RTL String Manipulation*, STR–51
STR$MATCH_WILD • *RTL String Manipulation*, STR–55
STR$MUL • *RTL String Manipulation*, STR–58
STR$POSITION • *RTL String Manipulation*, STR–62
STR$POS_EXTR • *RTL String Manipulation*, 2–9, STR–65
STR$PREFIX • *RTL String Manipulation*, 2–9, STR–68
STR$RECIP • *RTL String Manipulation*, STR–70
STR$REPLACE • *RTL String Manipulation*, STR–74
STR$RIGHT • *RTL String Manipulation*, 2–9, STR–77
STR$ROUND • *RTL String Manipulation*, STR–80
STR$TRANSLATE • *RTL String Manipulation*, STR–84
STR$TRIM • *RTL String Manipulation*, STR–87
STR$UPCASE • *RTL String Manipulation*, STR–89
STR built-in procedure • *VAXTPU*, 7–520 to 7–522
STREAM carriage control • *Convert*, CONV–2
STREAM format • *File Def Language*, FDL–35
Streamlined synchronization image • *Device Support (A)*, 13–28
loading • *Device Support (A)*, E–2
Stream record format • *File Applications*, 2–12
Stream record format option
See FAB$C_STM option
Stream record format with carriage return option
See FAB$C_STMCR option
Stream record format with line feed option
See FAB$C_STMLF option
STREAM_CR format • *File Def Language*, FDL–35
STREAM_LF format • *File Def Language*, FDL–35
Stride
vector • *MACRO*, 10–49
String • *Command Def*, CDU–4
See also Descriptor
See also String manipulation routine
allocating • *RTL General Purpose*, OTS–96
appending source string to end of destination string • *RTL String Manipulation*, STR–9
comparing for equality, no padding • *RTL String Manipulation*, STR–15
comparing two • *RTL String Manipulation*, STR–13
comparing without regard to case • *RTL String Manipulation*, STR–11
concatenating • *VAXTPU*, 3–4
concatenation • *RTL String Manipulation*, STR–20
converting contents of buffer to using STR • *VAXTPU*, 7–520
converting contents of range to using STR • *VAXTPU*, 7–520
converting to uppercase • *RTL String Manipulation*, STR–89
copying by descriptor • *RTL General Purpose*, OTS–90; *RTL Library*, LIB–336; *RTL String Manipulation*, STR–23
copying by reference • *RTL General Purpose*, OTS–92; *RTL Library*, LIB–338; *RTL String Manipulation*, STR–25
depositing ASCII • *Delta/XDelta*, DELTA–37
dividing two decimal strings • *RTL String Manipulation*, STR–28
dynamic length • *RTL String Manipulation*, 2–2, 2–3, 2–11, 2–12
evaluation rules • *RTL String Manipulation*, 2–1
finding substring • *RTL String Manipulation*, STR–62
fixed length • *RTL String Manipulation*, 2–1
formatting output • *System Services*, SYS–165
freeing • *RTL General Purpose*, OTS–95
how denoted • *National Char Set*, NCS–7

- String (Cont.)
  - inserting source string at front of destination • *RTL String Manipulation*, STR–68
  - limit on numeric representation • *National Char Set*, NCS–7
  - maximum length of • *RTL String Manipulation*, 2–2
  - null string • *RTL String Manipulation*, 2–11
  - output length argument • *RTL String Manipulation*, 2–8
  - ranges used in collating sequence • *National Char Set*, NCS–18
  - reciprocal of decimal string • *RTL String Manipulation*, STR–70
  - removing trailing blanks and tabs • *RTL String Manipulation*, STR–87
  - rounding or truncating a decimal string • *RTL String Manipulation*, STR–80
  - searching for file specification in • *System Services*, SYS–179
  - semantics of • *RTL String Manipulation*, 2–1, 2–4
  - skipping characters in • *RTL Library*, LIB–379
  - to insert with FAO • *VAXTPU*, 7–138
  - to insert with MESSAGE • *VAXTPU*, 7–268
  - to insert with MESSAGE_TEXT • *VAXTPU*, 7–271
  - translating matched characters • *RTL String Manipulation*, STR–84
- String argument • *MACRO*, 4–3
- String arithmetic
  - addition of decimal strings • *RTL String Manipulation*, STR–3
  - division of decimal strings • *RTL String Manipulation*, STR–28
  - multiplication • *RTL String Manipulation*, STR–58
- String constants • *VAXTPU*, 3–5
- String data type • *Routines Intro*, 2–17
  - character • *MACRO*, 8–7
  - leading separate numeric • *MACRO*, 8–11
  - packed decimal • *MACRO*, 8–13
  - trailing numeric • *MACRO*, 8–8
- STRING data type • *VAXTPU*, 2–23 to 2–24
- String descriptor • *RTL Library*, LIB–10; *RTL String Manipulation*, STR–7
- String instructions • *MACRO*, 9–126, 9–144
- String manipulation routine • *RTL String Manipulation*, 2–1
  - descriptor classes and string semantics • *RTL String Manipulation*, 2–4
  - how to select • *RTL String Manipulation*, 2–8
  - list of severe errors • *RTL String Manipulation*, 2–10
  - reading input string arguments • *RTL String Manipulation*, 2–6
- String manipulation routine (Cont.)
  - writing output string arguments • *RTL String Manipulation*, 2–6
- String operator
  - in macro • *MACRO*, 4–8
- /STRING qualifier • *Debugger*, 6–7, CD–121
- String type • *Debugger*, 4–17, 4–28
- String value • *File Def Language*, FDL–2, FDL–32
- String with bounds descriptor • *Routines Intro*, 2–41
- Strong definition • *Linker*, 2–9, 2–10
- Strong reference • *Linker*, 2–9
- Structure
  - error • *Analyze/RMS_File*, ARMS–13
  - examining • *Analyze/RMS_File*, ARMS–15
  - of file • *Analyze/RMS_File*, ARMS–1, ARMS–10
  - of indexed file • *Analyze/RMS_File*, ARMS–6; *File Def Language*, FDL–29
  - of relative file • *Analyze/RMS_File*, ARMS–2
  - of sequential file • *Analyze/RMS_File*, ARMS–1
  - pointer • *Analyze/RMS_File*, ARMS–21
- STS (status) field
  - See also Completion status field
  - See also FAB$L_STV field
  - contents • *RMS*, 2–6
- STUFF_SELECTION client message • *VAXTPU*, 7–344
- STV (status value) field
  - See also Completion status value field
  - contents • *RMS*, 2–6
- SUBB2 (Subtract Byte 2 Operand) instruction • *MACRO*, 9–30
- SUBB3 (Subtract Byte 3 Operand) instruction • *MACRO*, 9–30
- Subclass
  - finding out if a widget is a member of • *VAXTPU*, 7–214
- Subconditional assembly block directive • *MACRO*, 6–43
  - .IF_FALSE • *MACRO*, 6–43
  - .IF_TRUE • *MACRO*, 6–43
  - .IF_TRUE_FALSE • *MACRO*, 6–43
- Subconditional assembly block directive (.IF_x) • *MACRO*, 6–43
- Subcontroller • *Device Support (B)*, 1–33
- SUBD2 (Subtract D_floating 2 Operand) instruction • *MACRO*, 9–123
- SUBD3 (Subtract D_floating 3 Operand) instruction • *MACRO*, 9–123
- SUBF2 (Subtract F_floating 2 Operand) instruction • *MACRO*, 9–123
- SUBF3 (Subtract F_floating 3 Operand) instruction • *MACRO*, 9–123

SUBG2 (Subtract G_floating 2 Operand) instruction • *MACRO*, 9–123
SUBG3 (Subtract G_floating 3 Operand) instruction • *MACRO*, 9–123
SUBH2 (Subtract H_floating 2 Operand) instruction • *MACRO*, 9–123
SUBH3 (Subtract H_floating 3 Operand) instruction • *MACRO*, 9–123
Subkeys • *Librarian*, LIB–5, LIB–6
SUBL2 (Subtract Long 2 Operand) instruction • *MACRO*, 9–30
SUBL3 (Subtract Long 3 Operand) instruction • *MACRO*, 9–30
Sublock • *System Services Intro*, 12–11
Submit command file option
  See FAB$V_SCF option
SUBMIT_ON_CLOSE attribute • *File Def Language*, FDL–24
Subordinate
  creation of • *RTL Parallel Processing*, 2–3
  definition of • *RTL Parallel Processing*, 1–2
  deletion of • *RTL Parallel Processing*, 2–3
  notification of abnormal termination • *RTL Parallel Processing*, 2–3
  retrieving information about • *RTL Parallel Processing*, 2–4
SUBP4 (Subtract Packed 4 Operand) instruction • *MACRO*, 9–167
SUBP6 (Subtract Packed 6 Operand) instruction • *MACRO*, 9–167
Subprocess • *System Services Intro*, 8–2; *System Services*, SYS–99; *RTL Screen Management*, 4–2; *System Dump Analyzer*, SDA–162
  at DCL level • *VAXTPU*, 7–67
  built-in procedures
    ATTACH • *VAXTPU*, 7–35
    CREATE_PROCESS • *VAXTPU*, 7–67
    RECOVER_BUFFER • *VAXTPU*, 7–307
    SEND • *VAXTPU*, 7–342
    SEND_EOF • *VAXTPU*, 7–346
  built-in procedures for defining
    SPAWN • *VAXTPU*, 7–515
  connecting to using LIB$ATTACH • *RTL Library*, 2–9
  creating • *RTL Screen Management*, 4–2
    with LIB$SPAWN • *Programming Resources*, 2–2; *RTL Library*, 2–9
    with PPL$CREATE_PROCESS • *Programming Resources*, 2–4
    with PPL$ routines • *Programming Resources*, 4–16
Subprocess
  creating (Cont.)
    with SMG$ routines • *Programming Resources*, 7–16
    with SYS$CREPRC • *Programming Resources*, 2–3
  definition of • *RTL Parallel Processing*, 1–2
  deleting • *RTL Screen Management*, 4–2; *VAXTPU*, 7–67
  deleting with PPL$ routines • *Programming Resources*, 4–16
  disk and directory default • *System Services Intro*, 8–5
  executing commands • *RTL Screen Management*, 4–3
  image • *System Services Intro*, 8–3
  input, output, and error device • *System Services Intro*, 8–3
  priority
    setting • *Programming Resources*, 2–12
  program debugging • *Programming Resources*, 2–5
  restrictions • *VAXTPU*, 2–20
  running VAXTPU from • *VAXTPU*, A–5
  within VAXTPU • *VAXTPU*, 7–67
Subroutine
  definition of • *Routines Intro*, 2–3
SUBSTR built-in procedure • *VAXTPU*, 7–523 to 7–525
Substring • *RTL String Manipulation*, 2–10
  replacing • *RTL String Manipulation*, STR–74
.SUBTITLE directive • *MACRO*, 6–94
Subtitle listing control directive
  (.SUBTITLE) • *MACRO*, 6–94
Subtraction
  quadword times • *RTL Library*, LIB–397
  two's complement • *RTL Library*, LIB–400
Subtraction operator (–) • *System Dump Analyzer*, SDA–12
SUBW2 (Subtract Word 2 Operand) instruction • *MACRO*, 9–30
SUBW3 (Subtract Word 3 Operand) instruction • *MACRO*, 9–30
Success completion routine • *RMS*, 2–5
SUCCESS keyword • *VAXTPU*, 7–479
Successor
  See Logical successor
/SUCCESS qualifier
  in message definition • *Message*, MSG–22
"Success" string constant parameter to GET_INFO • *VAXTPU*, 7–207

/SUFFIX qualifier • *Debugger*, 10–17, CD–20, CD–73, CD–98, CD–101, CD–110, CD–116, CD–118, CD–125, CD–165, CD–217
Summary extended address block
  See XABSUM block
Summary of OPCODES
  alphabetic order • *MACRO*, D–1
  numeric order • *MACRO*, D–12
/SUMMARY qualifier • *System Dump Analyzer*, SDA–119
  compared with /CHECK qualifier • *Analyze/RMS_File*, ARMS–20
  description • *Analyze/RMS_File*, ARMS–20
  example of • *Analyze/RMS_File*, ARMS–20
  format • *Analyze/RMS_File*, ARMS–20
  limitation • *Analyze/RMS_File*, ARMS–13, ARMS–14, ARMS–20
  overview • *Analyze/RMS_File*, ARMS–20
  using with /OUTPUT qualifier • *Analyze/RMS_File*, ARMS–16
  with wildcard characters • *Analyze/RMS_File*, ARMS–10
Summary report • *Analyze/RMS_File*, ARMS–10
Summary XAB
  for key information • *RMS*, 13–1
Sum of absolute values
  of a vector • *RTL Math*, MTH–152
SUMSLP
  See SUMSLP Utility
SUMSLP Utility (SUMSLP) • *Programming Resources*, 1–20 to 1–21
  command file, running SUMSLP from a • *SUMSLP*, SUM–12
  directing output from • *SUMSLP*, SUM–14
  examples • *SUMSLP*, SUM–21
  how SUMSLP processes files • *SUMSLP*, SUM–7
  input source file • *SUMSLP*, SUM–1
  invoking • *SUMSLP*, SUM–2, SUM–14
  output file • *SUMSLP*, SUM–3
  qualifiers • *SUMSLP*, SUM–15 to SUM–20
  SUMSLP editing commands, specifying • *SUMSLP*, SUM–3
  SUMSLP files • *SUMSLP*, SUM–1
  SUMSLP messages • *SUMSLP*, SUM–13
  update file • *SUMSLP*, SUM–1
SUPERSEDE attribute • *File Def Language*, FDL–24
Supersede existing file option
  See FAB$V_SUP option
Supersede option • *File Applications*, 4–27, 5–9
SUPERSEDE secondary attribute • *File Applications*, 4–27
Supervisor-mode (PSL$C_SUPER) constant
  for FAB$V_CHAN_MODE • *RMS*, 5–5
/SUPERVISOR qualifier • *System Dump Analyzer*, SDA–157
Supervisor stack
  displaying contents • *System Dump Analyzer*, SDA–157
Supervisor stack pointer • *System Dump Analyzer*, SDA–14
SUP option • *File Def Language*, FDL–24
Supported terminals • *VAXTPU*, 1–8
Surface_Plot graph • *File Applications*, 4–12, A–2
Suspension • *System Services Intro*, 8–11, 8–14
  compared with hibernation • *System Services Intro*, 8–11
SVPCTX (Save Process Context) instruction • *MACRO*, 9–194
Swap mode
  changing • *Programming Resources*, 10–4
Swapper
  global symbols • *System Dump Analyzer*, SDA–61
Swapping
  vector • *RTL Math*, MTH–187
Swapping I/O function • *Device Support (B)*, 1–40
SWI$GL_FQFL
  replaced by CPU$Q_SWIQFL • *Device Support (A)*, E–14
Switch • *File Def Language*, FDL–2
Sychronization
  exception • *Routines Intro*, 2–13
  memory • *Routines Intro*, 2–13
Symbiont
  See also Queue
  allocating memory • *Utility Routines*, SMB–4
  carriage control
    processing of • *Utility Routines*, PSM–12
  connecting to a device • *Utility Routines*, SMB–4
  device • *Utility Routines*, PSM–2
  environments • *Utility Routines*, SMB–5
  function • *Utility Routines*, PSM–4, SMB–2
  input • *Utility Routines*, PSM–2, SMB–1
    INPSMB.EXE file • *Utility Routines*, SMB–1
  input routines
    demand • *Utility Routines*, PSM–6
  internal logic • *Utility Routines*, PSM–5
    main format routine • *Utility Routines*, PSM–13
    main input routine • *Utility Routines*, PSM–11
    main output routine • *Utility Routines*, PSM–15
  invoking VMS print symbiont • *Utility Routines*, PSM–23
  job controller
    communication with • *Utility Routines*, SMB–1

Symbiont (Cont.)
job controller request • *Utility Routines*, SMB–5
asynchronous • *Utility Routines*, SMB–6
processing • *Utility Routines*, SMB–10
reading • *Utility Routines*, SMB–10
responding • *Utility Routines*, SMB–13
synchronous • *Utility Routines*, SMB–6
modifying • *Utility Routines*, PSM–7, SMB–4
format routine • *Utility Routines*, PSM–13
guidelines • *Utility Routines*, PSM–8
initialization routine • *Utility Routines*, PSM–16
input routine • *Utility Routines*, PSM–10
integration of routines • *Utility Routines*, PSM–17
output routine • *Utility Routines*, PSM–14
restrictions • *Utility Routines*, PSM–8
multistream • *Utility Routines*, SMB–9
multithreaded • *Utility Routines*, PSM–3
output • *Utility Routines*, PSM–2, SMB–1
PRTSMB.EXE file • *Utility Routines*, SMB–1
print symbiont
internal logic • *Utility Routines*, PSM–5
modifying • *Utility Routines*, PSM–1
processing it performs • *Utility Routines*, PSM–1
user-written • *Utility Routines*, PSM–1
processing it performs • *Utility Routines*, PSM–5
process-permanent file • *Utility Routines*, SMB–4
server • *Utility Routines*, PSM–2, SMB–1
single stream • *Utility Routines*, SMB–9
stream
active • *Utility Routines*, PSM–3
multiple streams • *Utility Routines*, PSM–3
single stream • *Utility Routines*, PSM–3
SYSGEN MAXBUF parameter • *Utility Routines*, PSM–7
type • *Utility Routines*, SMB–1
user-written • *Utility Routines*, SMB–1, SMB–3
guidelines • *Utility Routines*, SMB–4
user-written routines
interfaces • *Utility Routines*, PSM–7
VMS printer • *Utility Routines*, SMB–1
Symbiont/Job Controller Interface routine
See SMB routines
Symbiont/Job Controller Interface routines
See SMB routines
Symbiont thread • *Utility Routines*, PSM–3
symbol • *Delta/XDelta*, DELTA–9
Symbol • *Command Def*, CDU–4; *Patch*, PAT–7 to PAT–14; *System Dump Analyzer*, SDA–13 to SDA–14, SDA–23; *MACRO*, 3–4; *VAXTPU*, 3–3 to 3–4

Symbol (Cont.)
See also DST
See also GST
See also RST
See also Scope
ambiguity, resolving • *Debugger*, 5–7
with DECwindows • *Debugger*, 1–28
built-in • *Debugger*, C–6, D–2
commands that affect • *Patch*, PAT–14
compiler generated type • *Debugger*, 4–4
creating • *Patch*, PAT–11, PAT–50, PAT–51
cross-reference listing • *Modular Procedures*, 3–8
cross-referencing • *MACRO*, 6–16, 6–66
defining • *Programming Resources*, 5–11; *Debugger*, 8–6, CD–48
defining for SDA • *System Dump Analyzer*, SDA–43
definition • *Modular Procedures*, A–6; *Linker*, 2–7
determining value of • *Patch*, PAT–60; *MACRO*, 3–6
displaying • *Debugger*, 5–9, 8–6, CD–48, CD–250; *System Dump Analyzer*, SDA–14
with DECwindows • *Debugger*, 1–25
entering into symbol table • *Patch*, PAT–78
evaluating • *System Dump Analyzer*, SDA–161
external • *MACRO*, 6–34, 6–101
global • *Programming Resources*, 5–11; *Debugger*, 5–4, 5–10; *Linker*, 2–8; *Patch*, PAT–7; *MACRO*, 3–6, 6–34, 6–37, 6–96, 6–101
image setting • *Debugger*, 5–14
information about, in map • *Linker*, 5–7
in message source file • *Message*, MSG–7
in operand field • *MACRO*, 3–6
in operator field • *MACRO*, 3–6
in place of numbers • *Modular Procedures*, 3–8, A–6
listing • *System Dump Analyzer*, SDA–161
list of • *Delta/XDelta*, DELTA–9
loading into the SDA symbol table • *System Dump Analyzer*, SDA–59
local • *Programming Resources*, 5–11; *Debugger*, 5–4; *Linker*, 2–8; *Patch*, PAT–8; *MACRO*, 3–6
macro name • *MACRO*, 3–6
made available to debugger • *MACRO*, 6–22
module name • *Patch*, PAT–8
module setting • *Debugger*, 5–6
with DECwindows • *Debugger*, 1–27
name • *System Dump Analyzer*, SDA–13, SDA–43
not in symbol table • *Debugger*, 5–6, 5–15
with DECwindows • *Debugger*, 1–27

Symbol (Cont.)
- not unique • *Debugger,* 5–9
  - with DECwindows • *Debugger,* 1–28
- passing • *Patch,* PAT–7
- patch area • *Patch,* PAT–18, PAT–38
- PATCH symbol table • *Patch,* PAT–7
- pathname • *Patch,* PAT–12
- permanent • *MACRO,* 3–5, 3–6
- program section name • *Patch,* PAT–8
- referring to • *Programming Resources,* 5–10
- register name • *MACRO,* 3–5, 3–6
- relation to address expression • *Debugger,* 4–4
  - with DECwindows • *Debugger,* 1–23
- relation to path name • *Debugger,* 5–9
  - with DECwindows • *Debugger,* 1–10
- removing from symbol table • *Patch,* PAT–41
- representing executive modules • *System Dump Analyzer,* SDA–104
- routine name • *Patch,* PAT–8
- search based on call stack • *Debugger,* 5–11, 5–12, CD–170
  - with DECwindows • *Debugger,* 1–8, 1–28
- search conventions • *Debugger,* 5–8, CD–171
  - with DECwindows • *Debugger,* 1–8, 1–28
- SET SCOPE command • *Debugger,* 5–11, CD–170
- shareable image • *Debugger,* 5–14
  - with DECwindows • *Debugger,* 1–30
- show symbol
  - with DECwindows • *Debugger,* 1–25
- SHOW SYMBOL command • *Debugger,* 5–9
- storage • *Programming Resources,* 5–10
- suppressing • *MACRO,* 6–23
- symbolic instruction label • *Patch,* PAT–9
- symbolic mode • *Debugger,* 4–14, CD–154
- traceback information • *Debugger,* 5–3
- transferral to VAX Symbolic Debugger • *MACRO,* 6–18
- translating address value into • *Patch,* PAT–13
- translating into address values • *Patch,* PAT–13, PAT–17
- types of • *Linker,* 2–8
- undefined • *MACRO,* 6–22
- universal • *Programming Resources,* 5–5; *Debugger,* 5–5, 5–12; *Linker,* 2–8; *Patch,* PAT–8, PAT–9
- unresolved • *Programming Resources,* 5–12
- user-defined • *System Dump Analyzer,* SDA–43; *MACRO,* 3–5, 3–6
- vector register • *Debugger,* 11–1

Symbol attribute directive
- (.WEAK) • *MACRO,* 6–101

Symbol definition
- $FABDEF
  - for defining symbols to USEROPEN routine • *File Applications,* 5–10
- $NAMDEF
  - for defining symbols to USEROPEN routine • *File Applications,* 5–10
- $RABDEF
  - for defining symbols to USEROPEN routine • *File Applications,* 5–10

Symbol definition for shareable image • *MACRO,* 6–96

Symbol definition macro
- description • *RMS,* 3–1
- using • *RMS,* 3–7

Symbol for shareable image directive (.TRANSFER) • *MACRO,* 6–96

Symbolic address
- use in locating start of control block • *RMS,* 3–7

Symbolic bit offset
- use in specifying options • *RMS,* 2–3

Symbolic debugger • *Programming Resources,* 1–14 to 1–15
- See also Debugger

Symbolic definition macro • *System Services Intro,* 2–8

Symbolic instruction label
- function of • *Patch,* PAT–9
- side effects when using patch • *Patch,* PAT–9

Symbolic mode • *Debugger,* 4–14, CD–154

Symbolic names
- assigning to starting address • *Patch,* PAT–18, PAT–38
- creating • *Patch,* PAT–50
- for argument lists • *System Services Intro,* 2–8

Symbolic naming exception
- control block • *RMS,* 2–3

Symbolic offset
- control block • *RMS,* 2–4
- format • *RMS,* 2–2
- use in locating control block fields • *RMS,* 2–2

/SYMBOLIC qualifier • *Debugger,* 4–14, CD–88

Symbolize
- address • *Debugger,* 3–13, 4–14, CD–271
  - with DECwindows • *Debugger,* 1–26
- register • *Debugger,* 4–14, CD–271
  - with DECwindows • *Debugger,* 1–26
- vector register • *Debugger,* 11–1

SYMBOLIZE command • *Debugger,* 3–13, 4–14, CD–271

Symbol list
- defining • *Device Support (B),* 2–29 to 2–30

Symbol record
  See Symbol
Symbol reference • *Linker,* 2–7
Symbol resolution • *Linker,* 1–6, 2–3, 2–7, 2–10, 4–8, 6–14, LINK–19, LINK–27, LINK–31
/SYMBOLS-/NOSYMBOLS qualifier
  with DELETE command • *Patch,* PAT–53
  with DEPOSIT command • *Patch,* PAT–56
  with EXAMINE command • *Patch,* PAT–63
  with INSERT command • *Patch,* PAT–68
  with REPLACE command • *Patch,* PAT–72
  with SET MODE command • *Patch,* PAT–77
  with VERIFY command • *Patch,* PAT–91
Symbol search mode • *Patch,* PAT–17
  See also Entry and display modes
SYMBOLS-NOSYMBOLS mode • *Patch,* PAT–16
/SYMBOLS qualifier • *Message,* MSG–13
  for EVALUATE • *System Dump Analyzer,* SDA–48
Symbol table • *Patch,* PAT–7, PAT–12
  See also DST
  See also GST
  See also RST
  See also SDA symbol table
  See also System symbol table
  of a library • *Linker,* 6–13
  of a shareable image • *Linker,* 1–5, 2–2
  specifying an alternate SDA • *System Dump Analyzer,* SDA–37
Symbol table file
  content of • *Linker,* 1–5, 2–3
  input to linker • *Linker,* 1–5, 2–3, 6–3
  output of linker • *Linker,* 2–6, LINK–16
  reading into SDA symbol table • *System Dump Analyzer,* SDA–59
  used as linker input • *Linker,* 1–5
/SYMBOL_TABLE qualifier • *Linker,* 2–6, LINK–16
SYNC (Scalar/Vector Instruction Synchronization) instruction • *MACRO,* 10–20, 10–37, 10–88
Synchronization • *Programming Resources,* 1–24; *RTL Parallel Processing,* 4–1; *MACRO,* 10–37
  barrier • *Programming Resources,* 4–17
  binary semaphore • *RTL Parallel Processing,* 4–10
  counting semaphore • *RTL Parallel Processing,* 4–10
  critical section • *RTL Parallel Processing,* 4–9
  deadlock • *RTL Parallel Processing,* 5–4
  debugging vectorized program • *Debugger,* 11–20, CD–198, CD–260, CD–273
  delivery of vector exception • *Debugger,* 11–19, 11–20, 11–23
  element • *RTL Parallel Processing,* 4–1
Synchronization (Cont.)
  passing control to another image • *Programming Resources,* 4–19
  semaphore • *RTL Parallel Processing,* 4–9
    operations on • *RTL Parallel Processing,* 4–10
  SET VECTOR_MODE command • *Debugger,* 11–20, CD–198
  SHOW VECTOR_MODE command • *Debugger,* 11–20, CD–260
  using asynchronous system traps • *Programming Resources,* 4–7
  using detached processes • *Programming Resources,* 4–8
  using events flags • *Programming Resources,* 4–1
  using process priority • *Programming Resources,* 4–19
  using semaphores with PPL$ routines • *Programming Resources,* 4–17
  using spin locks with PPL$ routines • *Programming Resources,* 4–16
  using subprocesses • *Programming Resources,* 4–8
Synchronization element
  comparing use of • *RTL Parallel Processing,* 5–7
  definition of • *RTL Parallel Processing,* 1–2
  retrieving information about • *RTL Parallel Processing,* 4–1
Synchronization image
  full-checking • *Device Support (A),* 13–28, E–2, E–17 to E–18
  streamlined • *Device Support (A),* 13–28, E–2
  uniprocessing • *Device Support (A),* 13–28, E–2
Synchronization techniques • *Device Support (A),* 1–7, 3–1 to 3–27
  See also Fork queue
  See also IPL
  See also Resource wait queue
  See also Spin lock
Synchronization with parallel processing routines
  See Parallel processing
SYNCHRONIZE VECTOR_MODE command • *Debugger,* 11–20, CD–273
Synchronous backplane interconnect
  See SBI
Synchronous communications device • *Device Support (B),* 1–76
Synchronous input/output • *Programming Resources,* 7–46
Synchronous memory management exception handling • *MACRO,* 10–30
Synchronous operation • *File Applications,* 8–17

Index

Synchronous SCSI data transfer mode
- determining REQ-ACK offset setting • *Device Support (B)*, 2–75
- determining transfer period setting • *Device Support (B)*, 2–75
- enabling • *I/O User's I*, 11–7, 11–13; *Device Support (A)*, 17–13; *Device Support (B)*, 2–88
- setting REQ-ACK offset • *Device Support (A)*, 17–13; *Device Support (B)*, 2–88
- setting transfer period • *Device Support (A)*, 17–13; *Device Support (B)*, 2–88

Synchronous status option
- See FAB$V_SYNCSTS option
- See RAB$V_SYNCSTS option

Synchronous system service • *Sysrem Services Intro*, 2–12

SYNONYM clause
- for DEFINE VERB statement • *Command Def*, CDU–35

Synonyms for commands • *VAXTPU*, G–5 to G–7

Syntax • *VAXTPU*, 4–3
- See also DEFINE SYNTAX statement
- changing • *Command Def*, CDU–5 to CDU–6

SYNTAX clause
- for DEFINE TYPE statement • *Command Def*, CDU–28
- for QUALIFIER clause • *Command Def*, CDU–25, CDU–34

Syntax-name verb clause • *Command Def*, CDU–5

Syntax rules for PATCH commands
- delimiting parameter values • *Patch*, PAT–23
- entering ASCII data strings • *Patch*, PAT–20
- entering comments • *Patch*, PAT–23
- entering numeric data • *Patch*, PAT–22
- entering VAX MACRO instructions • *Patch*, PAT–21
- operators for addressing locations • *Patch*, PAT–24
- operators for arithmetic expressions • *Patch*, PAT–23
- VAX MACRO instructions with same opcodes • *Patch*, PAT–21

SYS$ADD_HOLDER • *System Services Intro*, 3–9; *System Services*, SYS–3

SYS$ADD_IDENT • *System Services Intro*, 3–9; *System Services*, SYS–5

SYS$ADJSTK • *System Services*, SYS–8

SYS$ADJWSL • *System Services Intro*, 11–6; *System Services*, SYS–10

SYS$ALLOC • *System Services*, SYS–12; *Device Support (B)*, 1–74, 1–77
- example • *System Services Intro*, 7–23

SYS$AR_JOBCTLMB • *Device Support (A)*, 9–7, E–7

SYS$AR_OPRMBX • *Device Support (A)*, 10–7, E–7

SYS$ASCEFC • *System Services*, SYS–15

SYS$ASCTIM • *Programming Resources*, 3–24; *System Services*, SYS–18
- example • *System Services Intro*, 9–2

SYS$ASCTOID • *System Services Intro*, 3–8; *System Services*, SYS–21

SYS$ASSIGN • *Programming Resources*, 7–45; *System Services*, SYS–23; *I/O User's II*, 2–9, 5–6, 6–2; *Device Support (A)*, 1–6, 2–3, 4–5, 19–9; *Device Support (B)*, 1–11, 1–77, 1–78
- example • *System Services Intro*, 7–14
- for template device • *Device Support (B)*, 4–6

SYS$ASSIGN routine • *I/O User's I*, 7–2, 8–17, 8–52

SYS$BINTIM • *Programming Resources*, 3–24; *System Services Intro*, 9–3; *System Services*, SYS–27; *RMS*, 3–10

SYS$BRKTHRU • *System Services*, SYS–30

SYS$BRKTHRUW • *System Services*, SYS–38

SYS$CANCEL • *System Services*, SYS–39; *Device Support (A)*, 1–4, 11–6, 11–8, 18–17, 19–19; *Device Support (B)*, 1–30, 4–4
- example • *System Services Intro*, 7–22

SYS$CANCEL routine • *I/O User's I*, 4–14

SYS$CANEXH • *System Services*, SYS–41

SYS$CANTIM • *System Services*, SYS–42
- example • *System Services Intro*, 9–6

SYS$CANWAK • *System Services Intro*, 9–7; *System Services*, SYS–44

SYS$CHANGE_ACL • *System Services Intro*, 3–18, 3–25; *System Services*, SYS–46

SYS$CHECK_ACCESS • *System Services Intro*, 3–31; *System Services*, SYS–51

SYS$CHFDEF macro • *System Services Intro*, 10–8

SYS$CHKPRO • *System Services Intro*, 3–30; *System Services*, SYS–56

SYS$CLOSE
- See Close service

SYS$CLREF • *System Services Intro*, 4–4; *System Services*, SYS–63

SYS$CMEXEC • *System Services*, SYS–64

SYS$CMKRNL • *System Services*, SYS–66

SYS$CONNECT
- See Connect service

SYS$CREATE • *Programming Resources*, 8–8
- See also Create service

SYS$CREATE_RDB • *System Services Intro*, 3–6; *System Services*, SYS–80

SYS$CRELNM • *System Services*, SYS–68

SYS$CRELNT • *System Services*, SYS–74

SYS$CREMBX • *Programming Resources*, 3–8; *System Services*, SYS–82

SYS$CREMBX routine • *I/O User's I*, 7–1
SYS$CREPRC • *System Services*, SYS–88
  example • *System Services Intro*, 8–3
SYS$CRETVA • *Programming Resources*, 10–3; *System Services*, SYS–102
  See also SYS$EXPREG
SYS$CRMPSC • *Programming Resources*, 8–4, 8–5; *System Services*, SYS–105; *Device Support (A)*, 19–5 to 19–6, 19–8
SYS$DACEFC • *System Services*, SYS–116
SYS$DALLOC • *System Services*, SYS–117; *Device Support (A)*, 11–8, 18–17; *Device Support (B)*, 1–30, 1–77, 4–4
SYS$DASSGN • *Programming Resources*, 8–9; *System Services*, SYS–119; *I/O User's II*, 6–2; *Device Support (A)*, 11–7, 11–8, 18–17; *Device Support (B)*, 1–30, 1–77, 4–4
  example • *System Services Intro*, 7–20
SYS$DASSGN routine • *I/O User's I*, 7–2
SYS$DCLAST • *System Services*, SYS–121
  example • *System Services Intro*, 5–5
SYS$DCLCMH • *System Services*, SYS–123
SYS$DCLEXH • *Programming Resources*, 9–27; *System Services*, SYS–125
  example • *System Services Intro*, 8–16
SYS$DELETE
  See Delete service
SYS$DELLNM • *System Services*, SYS–127
SYS$DELMBX • *System Services*, SYS–130
SYS$DELMBX routine • *I/O User's I*, 7–3
SYS$DELPRC • *System Services Intro*, 8–18; *System Services*, SYS–132
SYS$DELTVA • *Programming Resources*, 8–9; *System Services*, SYS–134
SYS$DEQ • *System Services*, SYS–136
  example • *System Services Intro*, 12–14
SYS$DGBLSC • *System Services*, SYS–140
SYS$DISCONNECT
  See Disconnect service
SYS$DISK
  applied to file specification • *File Applications*, 6–2
  as SDA output • *System Dump Analyzer*, SDA–72
  global read • *System Dump Analyzer*, SDA–60
SYS$DISMOU • *System Services Intro*, 7–26; *System Services*, SYS–143
SYS$DISMOUNT routine • *I/O User's I*, 1–32
SYS$DISPLAY
  See Display service
SYS$DLCEFC • *System Services*, SYS–146
SYS$ENQ • *System Services*, SYS–148
  example • *System Services Intro*, 12–7, 12–10
SYS$ENQW • *System Services*, SYS–158
SYS$ENTER
  See Enter service
SYS$ERAPAT • *System Services Intro*, 3–35; *System Services*, SYS–159
SYS$ERASE
  See Erase service
SYS$ERROR • *Programming Resources*, 9–24
SYS$ERROR warning message • *Convert*, CONV–3
SYS$EXIT • *System Services Intro*, 8–15; *System Services*, SYS–162
  causing call to for process • *System Services*, SYS–191
SYS$EXPREG • *Programming Resources*, 10–3; *System Services*, SYS–163
  example • *System Services Intro*, 11–2
SYS$EXTEND
  See Extend service
SYS$FAO • *Programming Resources*, 3–24; *System Services*, SYS–165
  directive
    format of • *System Services*, SYS–167
    list of • *System Services*, SYS–168
  example • *System Services Intro*, 7–28; *System Services*, SYS–171, SYS–172
SYS$FAOL
  example • *System Services*, SYS–174
SYS$FILESCAN • *System Services*, SYS–179; *File Applications*, 5–8
SYS$FIND
  See Find service
SYS$FIND_HELD • *System Services Intro*, 3–10, 3–15; *System Services*, SYS–184
SYS$FIND_HOLDER • *System Services Intro*, 3–10, 3–15; *System Services*, SYS–187
SYS$FINISH_RDB • *System Services*, SYS–190
SYS$FLUSH
  See Flush service
SYS$FORCEX • *System Services*, SYS–191
  See also SYS$DELPRC
  example • *System Services Intro*, 8–16
SYS$FORMAT_ACL • *System Services Intro*, 3–18, 3–25; *System Services*, SYS–193
SYS$FREE
  See Free service
SYS$GET
  See Get service
SYS$GETDVI • *Programming Resources*, 7–50; *System Services*, SYS–203
  asynchronous DDCMP driver • *I/O User's II*, 5–2
  DMC11/DMR11 device • *I/O User's II*, 1–3
  DMP11/DMF11 device • *I/O User's II*, 2–3

SYS$GETDVI (Cont.)
- DR11–W/DRV11–WA device • *I/O User's II*, 3–8
- DR32 device • *I/O User's II*, 4–3
- Ethernet/802 drivers • *I/O User's II*, 6–14
- SCSI generic class driver • *I/O User's I*, 11–14

SYS$GETDVI routine • *I/O User's I*, 6–11
- card reader • *I/O User's I*, 2–5
- disk • *I/O User's I*, 3–22
- line printer • *I/O User's I*, 5–3
- LPA11-K device • *I/O User's I*, 4–5
- mailbox • *I/O User's I*, 7–4
- terminal • *I/O User's I*, 8–20

SYS$GETDVIW • *System Services*, SYS–221

SYS$GETJPI • *System Services*, SYS–222
- example • *System Services*, SYS–237

SYS$GETJPIW • *System Services*, SYS–238

SYS$GETLKI • *System Services*, SYS–239

SYS$GETLKIW • *System Services*, SYS–252

SYS$GETMSG • *System Services*, SYS–253

SYS$GETQUI • *Programming Resources*, 3–22; *System Services*, SYS–257

SYS$GETQUIW • *System Services*, SYS–297

SYS$GETSYI • *Programming Resources*, 3–22; *System Services*, SYS–299

SYS$GETSYIW • *System Services*, SYS–313

SYS$GETTIM • *Programming Resources*, 3–24; *System Services Intro*, 9–2; *System Services*, SYS–314

SYS$GETUAI • *System Services*, SYS–315

SYS$GL_JOBCTLMB
- replaced by SYS$AR_JOBCTLMB • *Device Support (A)*, E–7

SYS$GL_OPRMBX
- replaced by SYS$AR_OPRMBX • *Device Support (A)*, E–7

SYS$GRANTID • *System Services*, SYS–326

SYS$HIBER • *System Services*, SYS–330
- example • *System Services Intro*, 8–13
- use of • *RTL Parallel Processing*, 5–5

SYS$IDTOASC • *System Services Intro*, 3–8, 3–15; *System Services*, SYS–332

SYS$INPUT • *Programming Resources*, 9–24; *Linker*, 3–4
- default value of • *Programming Resources*, 7–2
- redefining • *Programming Resources*, 7–3
- using with LIB$GET_INPUT • *Programming Resources*, 7–3
- using with LIB$PUT_OUTPUT • *Programming Resources*, 7–3

SYS$LCKPAG • *Programming Resources*, 10–4; *System Services*, SYS–335

SYS$LIBRARY • *Linker*, 6–14

SYS$LIBRARY:IMAGELIB.OLB • *Programming Resources*, 5–12; *Linker*, 1–5, 2–4, 4–11, 5–4, 6–7, 6–14, LINK–8
- searched by linker • *Linker*, LINK–17

SYS$LIBRARY:STARLET.MLB
- as source of macros • *RMS*, 1–1, 3–2

SYS$LIBRARY:STARLET.OLB • *Linker*, 1–5, 2–4, 6–14, LINK–8
- searched by linker • *Linker*, LINK–17

SYS$LKWSET • *Programming Resources*, 10–3; *System Services Intro*, 11–6; *System Services*, SYS–337

SYS$LOADABLE_IMAGES directory • *Device Support (A)*, E–8

SYS$MANAGER:SYSTARTUP.COM
- invoking SDA • *System Dump Analyzer*, SDA–5
- producing an SDA listing • *System Dump Analyzer*, SDA–5
- releasing page file blocks • *System Dump Analyzer*, SDA–3

SYS$MGBLSC • *Programming Resources*, 5–15; *System Services*, SYS–339

SYS$MOD_HOLDER • *System Services Intro*, 3–13; *System Services*, SYS–344

SYS$MOD_IDENT • *System Services Intro*, 3–12; *System Services*, SYS–347

SYS$MOUNT • *System Services Intro*, 7–24; *System Services*, SYS–350

SYS$MTACCESS • *System Services Intro*, 3–35; *System Services*, SYS–363

SYS$NUMTIM • *System Services Intro*, 9–7; *System Services*, SYS–366

SYS$NXTVOL
- See Next Volume service

SYS$OPEN • *Programming Resources*, 8–8
- See also Open service

SYS$OUTPUT • *Analyze/RMS_File*, ARMS–16
- default value of • *Programming Resources*, 7–2
- for check report • *File Applications*, 10–1
- redefining • *Programming Resources*, 7–3
- using with LIB$GET_INPUT • *Programming Resources*, 7–3
- using with LIB$PUT_OUTPUT • *Programming Resources*, 7–3
- with CONVERT • *Convert*, CONV–9

SYS$OUTPUT_HELP • *Programming Resources*, 8–36

SYS$PARSE
- See Parse service

SYS$PARSE_ACL • *System Services Intro*, 3–18, 3–25; *System Services*, SYS–368

SYS$PURGWS • *System Services*, SYS–370

SYS$PURGWS (Cont.)
See also SYS$ADJWSL
SYS$PUT
See Put service
SYS$PUTMSG • *Programming Resources*, 9–15, 9–22; *System Services*, SYS–371
SYS$QIO • *Programming Resources*, 7–45; *System Services*, SYS–379; *Device Support (A)*, 1–1, 2–2 to 2–4, 4–1 to 4–15; *Device Support (B)*, 1–37
device-dependent arguments of • *Device Support (B)*, 1–41
example • *System Services Intro*, 7–14
for additional processing • *RMS*, 5–18
for connect to interrupt facility • *Device Support (A)*, 19–9, 19–9 to 19–13
format for request to SCSI generic class driver • *I/O User's I*, 11–11
use in I/O operation • *RMS*, 2–7
SYS$QIOW • *Programming Resources*, 7–45; *System Services*, SYS–384; *Device Support (A)*, 2–7; *Device Support (B)*, 1–37
SYS$READ
See Read service
SYS$READEF • *System Services*, SYS–385
SYS$RELEASE
See Release service
SYS$REMOVE
See Remove service
SYS$REM_HOLDER • *System Services Intro*, 3–15; *System Services*, SYS–387
SYS$REM_IDENT • *System Services Intro*, 3–15; *System Services*, SYS–389
SYS$RENAME
See also Rename service
noting format difference • *RMS*, 3–11
SYS$RESUME • *System Services*, SYS–391
SYS$REVOKID • *System Services*, SYS–393
SYS$REWIND
See Rewind service
SYS$RMSRUNDWN • *System Services*, SYS–514
SYS$SCHDWK • *System Services*, SYS–397
canceling • *System Services Intro*, 9–7
converting time format for • *System Services*, SYS–27
example • *System Services Intro*, 9–6
request • *System Services Intro*, 9–6
SYS$SEARCH
See Search service
SYS$SETAST • *System Services*, SYS–400
SYS$SETDDIR • *System Services*, SYS–516; *File Applications*, 6–14
SYS$SETDFPROT • *System Services*, SYS–518
SYS$SETEF • *System Services Intro*, 4–4; *System Services*, SYS–401
SYS$SETEXV • *Programming Resources*, 9–13; *System Services*, SYS–402
example • *System Services Intro*, 10–6
SYS$SETIME • *System Services Intro*, 9–8; *System Services*, SYS–404
SYS$SETIMR • *System Services Intro*, 9–4; *System Services*, SYS–406
converting time format for • *System Services*, SYS–27
example with AST • *System Services Intro*, 5–1
SYS$SETPRA • *System Services*, SYS–409
SYS$SETPRI • *System Services*, SYS–411
SYS$SETPRN • *System Services*, SYS–413
SYS$SETPRT • *System Services*, SYS–414
SYS$SETPRV • *System Services*, SYS–417
SYS$SETRWM • *System Services Intro*, 7–3; *System Services*, SYS–421
SYS$SETSFM • *System Services*, SYS–423
example • *System Services Intro*, 2–14
use in signaling errors • *RMS*, 2–6
SYS$SETSSF • *System Services*, SYS–425
SYS$SETSTK • *System Services*, SYS–427
SYS$SETSWM • *System Services*, SYS–429
example • *System Services Intro*, 11–7
SYS$SETUAI • *System Services*, SYS–431
SYS$SHARE • *Programming Resources*, 5–9; *Linker*, 4–12, 4–17, 4–22
SYS$SNDERR • *System Services*, SYS–441
SYS$SNDJBC • *System Services*, SYS–441
SYS$SNDJBCW • *System Services*, SYS–493
SYS$SNDOPR • *System Services*, SYS–495
SYS$SPACE
See Space service
SYS$SUSPND • *System Services*, SYS–509
SYS$SYNCH • *System Services*, SYS–512; *Device Support (A)*, 2–7
SYS$SYSTEM:OPCCRASH.COM
involvement in writing crash dump • *System Dump Analyzer*, SDA–5
SYS$SYSTEM:PAGEFILE.SYS • *System Dump Analyzer*, SDA–5, SDA–28
See also System dump file
as dump file • *System Dump Analyzer*, SDA–3
releasing blocks containing a crash dump • *System Dump Analyzer*, SDA–36
SYS$SYSTEM:REQSYSDEF.STB • *System Dump Analyzer*, SDA–6, SDA–7

SYS$SYSTEM:SHUTDOWN.COM
  involvement in writing crash dump • *System Dump Analyzer*, SDA–5
SYS$SYSTEM:SYS.EXE • *Linker*, 2–6; *System Dump Analyzer*, SDA–59
  contents • *System Dump Analyzer*, SDA–60, SDA–104
SYS$SYSTEM:SYS.STB • *Linker*, LINK–27; *System Dump Analyzer*, SDA–6, SDA–7, SDA–9, SDA–15
SYS$SYSTEM:SYSDEF.STB • *System Dump Analyzer*, SDA–8
SYS$SYSTEM:SYSDUMP.DMP • *System Dump Analyzer*, SDA–28
  See also System dump file
  protection • *System Dump Analyzer*, SDA–5
  size of • *System Dump Analyzer*, SDA–3
SYS$TRNLNM • *System Services*, SYS–520
SYS$TRUNCATE
  See Truncate service
SYS$ULKPAG • *Programming Resources*, 10–4; *System Services*, SYS–526
SYS$ULWSET • *Programming Resources*, 10–4; *System Services*, SYS–528
SYS$UNWIND • *Programming Resources*, 9–18; *System Services*, SYS–530
  example • *System Services Intro*, 10–16
SYS$UPDATE
  See also Update service
SYS$UPDSEC • *Programming Resources*, 8–9; *System Services*, SYS–532
SYS$UPDSECW • *System Services*, SYS–536
SYS$WAIT
  See Wait service
SYS$WAITFR • *System Services*, SYS–537
SYS$WAKE • *System Services*, SYS–538
  See also SYS$HIBER
  example • *System Services Intro*, 8–13
  use of • *RTL Parallel Processing*, 5–5
SYS$WFLAND • *System Services*, SYS–540
SYS$WFLOR • *System Services*, SYS–542
SYS$WRITE
  See Write service
SYSAP (system application) • *System Dump Analyzer*, SDA–148
SYSDEVICE.EXE
  global symbols • *System Dump Analyzer*, SDA–61
SYSGEN
  See System Generation Utility
SYSGEN parameter
  global section • *RTL Parallel Processing*, 1–7
SYSGETSYI.EXE
  global symbols • *System Dump Analyzer*, SDA–61
/SYSLIB qualifier • *Linker*, LINK–17
SYSLICENSE.EXE
  global symbols • *System Dump Analyzer*, SDA–61
SYSLOA symbol • *System Dump Analyzer*, SDA–14
SYSMSG.EXE
  global symbols • *System Dump Analyzer*, SDA–61
SYSPRV privilege • *System Services Intro*, 7–6
  requirement for creating files with different UIC • *RMS*, 14–8
/SYSSHR qualifier • *Linker*, LINK–18
System
  analyzing a running • *System Dump Analyzer*, SDA–8 to SDA–9, SDA–32
  analyzing a running system • *System Dump Analyzer*, SDA–2
  default • *File Applications*, 4–14
  exception dispatcher • *System Services Intro*, 10–7
  getting information about
    asynchronously • *System Services*, SYS–299
    synchronously • *System Services*, SYS–313
  investigating performance problems • *System Dump Analyzer*, SDA–8
  library • *System Services Intro*, 2–1, 2–5
  mailbox • *System Services Intro*, 7–33
  message • *System Services Intro*, 2–17
  resources • *File Applications*, 1–15
System application
  See SYSAP
SYSTEM attribute • *File Def Language*, FDL–2, FDL–38
System block
  See SB
System buffer
  See Buffer
  See Nonpaged pool
System command table • *Command Def*, CDU–2
  adding commands to • *Command Def*, CDU–3
System configuration • *Device Support (A)*, 12–11
System console terminal • *I/O User's I*, 8–1
System context • *Device Support (A)*, 1–8
System control block
  See SCB
System Control Block (SCB) vector • *MACRO*, 10–28
System control unit (SCU) • *Device Support (A)*, 1–16
System default • *File Def Language*, FDL–30
System default library • *Linker*, 1–5, 2–4, LINK–18
  content of • *Linker*, 2–4
  linker's search of • *Linker*, LINK–17, LINK–31
  processing of • *Linker*, 6–14

System default library (Cont.)
searched by linker • *Linker*, LINK–17
symbols in • *Linker*, LINK–5
System directory table • *System Services Intro*, 6–3
System Dump Analyzer
See SDA
System dump file • *System Dump Analyzer*, SDA–2 to SDA–3
copying • *System Dump Analyzer*, SDA–4
header • *System Dump Analyzer*, SDA–5
mapping physical memory to • *System Dump Analyzer*, SDA–7
requirements for analysis • *System Dump Analyzer*, SDA–6
saving • *System Dump Analyzer*, SDA–4
size • *System Dump Analyzer*, SDA–3
System failure • *MACRO*, E–10
analyzing • *System Dump Analyzer*, SDA–15 to SDA–28
causing • *System Dump Analyzer*, SDA–28 to SDA–31
diagnosing from PC contents • *System Dump Analyzer*, SDA–15
example • *System Dump Analyzer*, SDA–21 to SDA–28
inducing with XDELTA • *Device Support (A)*, 13–21
summary • *System Dump Analyzer*, SDA–93
System Generation Utility (SYSGEN) • *Device Support (A)*, 12–2 to 12–23
AUTOCONFIGURE command • *Device Support (A)*, 11–4, 12–13 to 12–23; *Device Support (B)*, 1–2, 1–34, 1–68, 2–22, 4–21
configuring SCSI devices • *I/O User's I*, 11–9; *Device Support (A)*, 17–30
CONNECT command • *Device Support (A)*, 11–4, 12–2, 12–3 to 12–7, E–3; *Device Support (B)*, 1–7, 1–26, 1–36, 1–44, 1–68, 2–22, 4–8, 4–22
/ADAPTER qualifier • *Device Support (A)*, 12–5
/ADPUNIT qualifier • *Device Support (A)*, 12–6
/CSR qualifier • *Device Support (A)*, 12–5
/CSR_OFFSET qualifier • *Device Support (A)*, 12–6
/DRIVERNAME qualifier • *Device Support (A)*, 12–6
/MAXUNITS qualifier • *Device Support (A)*, 12–6
/NOADAPTER qualifier • *Device Support (A)*, 12–5
/NUMVEC qualifier • *Device Support (A)*, 12–6, 14–31, 14–32; *Device Support (B)*, 1–23
System Generation Utility (SYSGEN)
CONNECT command (Cont.)
/VECTOR qualifier • *Device Support (A)*, 12–6
/VECTOR_OFFSET qualifier • *Device Support (A)*, 12–6
device table • *Device Support (A)*, 12–15, 12–23
LOAD command • *Device Support (A)*, 11–4, 12–2 to 12–3, E–3
loading a VAXBI device driver using • *Device Support (A)*, 16–23
RELOAD command • *Device Support (A)*, 11–4, 12–7 to 12–8; *Device Support (B)*, 4–10
SHOW/ADAPTER command • *Device Support (A)*, 12–8
SHOW/BI command • *Device Support (A)*, 12–9
SHOW/BUS command • *Device Support (A)*, 12–10
SHOW/CONFIGURATION command • *Device Support (A)*, 12–11 to 12–12
SHOW/DEVICE command • *Device Support (A)*, 12–12
SHOW/XMI command • *Device Support (A)*, 12–11
System hang • *System Dump Analyzer*, SDA–28
System help
library • *Librarian*, LIB–8
System image • *Linker*, 6–2, LINK–19
content of • *Linker*, 6–2
contents • *System Dump Analyzer*, SDA–60, SDA–104
memory allocation for • *Linker*, 6–2
output of linker • *Linker*, 2–6
System information
See Timer, statistics
SYSTEM keyword • *VAXTPU*, 7–480
System logical name table • *System Services Intro*, 6–6
System management • *File Applications*, 3–8
creating a crash dump file • *System Dump Analyzer*, SDA–2
image activation • *File Applications*, 5–5
System manager • *File Def Language*, FDL–16
System map • *System Dump Analyzer*, SDA–15
System message routines
global symbols • *System Dump Analyzer*, SDA–61
System page
locking in memory • *Device Support (A)*, E–16
System page table (SPT)
displaying • *System Dump Analyzer*, SDA–23, SDA–111
in system dump file • *System Dump Analyzer*, SDA–2, SDA–7

- System page-table entry
  - allocating • *Device Support (A)*, 16–18, E–7; *Device Support (B)*, 3–107
  - allocating permanent • *Device Support (A)*, 6–2; *Device Support (B)*, 1–33, 1–79, 2–21, 3–79, 3–80
  - deallocating • *Device Support (B)*, 3–108
- System paging file
  - as dump file • *System Dump Analyzer*, SDA–3
  - releasing blocks containing a crash dump • *System Dump Analyzer*, SDA–36
- System parameters • *File Applications*, 1–16
- System PCB (process control block)
  - displaying • *System Dump Analyzer*, SDA–128
- System process • *System Dump Analyzer*, SDA–73
- SYSTEM protection code • *File Def Language*, FDL–23
- /SYSTEM qualifier • *Debugger*, 3–14, CD–134, CD–190, CD–267; *Linker*, 2–6, LINK–19; *System Dump Analyzer*, SDA–52, SDA–73, SDA–111, SDA–115, SDA–128
  - in .FACILITY directive • *Message*, MSG–18
- System region
  - examining • *System Dump Analyzer*, SDA–52
- System resources • *Modular Procedures*, 2–12
  - accessing • *Device Support (B)*, 2–47 to 2–48
- System routine documentation • *Routines Intro*, 1–1
  - arguments heading • *Routines Intro*, 1–7
    - access entry • *Routines Intro*, 1–9
    - mechanism entry • *Routines Intro*, 1–10
    - text entry • *Routines Intro*, 1–11
    - type entry • *Routines Intro*, 1–8
    - VMS Usage entry • *Routines Intro*, 1–7
  - condition values returned • *Routines Intro*, 1–12
    - returns • *Routines Intro*, 1–12, 1–14
    - returns in I/O status block • *Routines Intro*, 1–14
    - returns in mailbox • *Routines Intro*, 1–14
    - returns signaled • *Routines Intro*, 1–15
  - description of • *Routines Intro*, 1–1
  - format heading • *Routines Intro*, 1–2
    - explanatory text • *Routines Intro*, 1–4
    - JSB call format • *Routines Intro*, 1–4
    - procedure call format • *Routines Intro*, 1–3
  - main headings • *Routines Intro*, 1–1
  - returns heading • *Routines Intro*, 1–5
    - condition values • *Routines Intro*, 1–5
    - reigister data • *Routines Intro*, 1–6
  - routine name heading • *Routines Intro*, 1–1
  - routine overview heading • *Routines Intro*, 1–1
- System routines • *Programming Resources*, 1–22 to 1–24
- System routines (Cont.)
  - system services
    - asynchronous • *Programming Resources*, 4–12
    - synchronous • *Programming Resources*, 4–12
- System routine template • *Routines Intro*, 1–1
- Systems
  - communication between • *Programming Resources*, 3–26
- System service • *Programming Resources*, 1–29; *Modular Procedures*, 3–11, A–2
  - checking completion status of • *System Services*, SYS–512
  - executing
    - asynchronously • *System Services Intro*, 2–12
    - synchronously • *System Services Intro*, 2–12
  - failure exception condition • *System Services Intro*, 2–14
  - inhibiting user mode calls to • *System Services*, SYS–425
  - MACRO • *System Services Intro*, 2–1, 2–5
  - return status • *Programming Resources*, 9–3
  - setting failure exception mode • *System Services*, SYS–423
  - setting filter • *System Services*, SYS–425
  - what is available • *Modular Procedures*, 1–8
- System service access • *RTL Library*, 2–1, 2–2
- System service dispatcher
  - role in servicing I/O request • *Device Support (A)*, 4–1
- System service exception • *RMS*, 2–6
- System service exception generation
  - disabling • *RMS*, 2–6
- System space
  - base address • *System Dump Analyzer*, SDA–14
  - SET BREAK command • *Debugger*, CD–134
  - SET STEP command • *Debugger*, CD–180
  - SET TRACE command • *Debugger*, CD–190
  - STEP command • *Debugger*, CD–267
- System space operator (G) • *System Dump Analyzer*, SDA–12
- System space prefix symbol • *Delta/XDelta*, DELTA–9
- System spin lock • *Device Support (A)*, 3–13
- "System" string constant parameter to GET_INFO • *VAXTPU*, 7–175
- System symbol table • *Linker*, LINK–27; *System Dump Analyzer*, SDA–6, SDA–13
- System time • *Programming Resources*, 3–23; *Device Support (A)*, 3–8, 3–14, E–13; *Device Support (B)*, 3–69
  - reading • *Device Support (A)*, E–15; *Device Support (B)*, 2–52
  - setting • *System Services*, SYS–404

- System time quadword
  - examining • *System Dump Analyzer,* SDA–52
- System timer
  - canceling • *Programming Resources,* 4–12
  - setting • *Programming Resources,* 4–11
- system_access_id data type • *Routines Intro,* A–12t
- SYSTEM_PRIMITIVES.EXE
  - global symbols • *System Dump Analyzer,* SDA–61
- SYSTEM_SYNCHRONIZATION.EXE
  - global symbols • *System Dump Analyzer,* SDA–61

T

- Tab
  - Ctrl/I • *I/O User's I,* 8–6
  - terminal mechanical • *I/O User's I,* 8–21
  - terminal tab stops • *I/O User's I,* 8–35
- TAB key command • *Delta/XDelta,* DELTA–24
- Table
  - See also Command table
- /TABLE qualifier • *Command Def,* CDU–44
- Tab stops
  - in source statement • *MACRO,* 2–1
- TAB_STOPS keyword
  - used with SET • *VAXTPU,* 7–481
- "Tab_stops" string constant parameter to GET_ INFO • *VAXTPU,* 7–175
- Tangent • *RTL Math,* MTH–104, MTH–106, MTH–139, MTH–141
  - hyperbolic • *RTL Math,* MTH–108, MTH–143
- Tape
  - See Magnetic tape
- Tape class driver
  - disabling the loading of • *I/O User's I,* 11–10; *Device Support (A),* 17–31
- Tape driver • *Device Support (B),* 1–74, 4–13
  - using local tape UCB extension • *Device Support (B),* 1–69, 1–81 to 1–82
- Tape mark • *I/O User's I,* 6–17, 6–20
- Tape processing
  - run-time options • *File Applications,* 9–13 to 9–14
- Tape volume
  - mounting • *System Services Intro,* 7–24
- Target • *Device Support (A),* 17–2
  - enabling selection from • *Device Support (A),* 17–28 to 17–30; *Device Support (B),* 2–70, 2–73 to 2–90
- TARGET attribute • *File Def Language,* FDL–38
- Target mode
  - See Asynchronous event notification
- %TASK • *Debugger,* D–10
- Tasking
  - debugging • *Debugger,* CD–182, CD–253
    - with DECwindows • *Debugger,* 1–30
  - SET TASK command • *Debugger,* CD–182
  - SHOW TASK command • *Debugger,* CD–253
- /TASK qualifier • *Debugger,* CD–63, CD–88
- TB (Translation buffer)
  - vector • *MACRO,* 10–7, 10–8, 10–20, 10–32, 10–34, 10–41, 10–47
- TBIA (TB Invalidate All) instruction • *MACRO,* 10–47
- TBIS (TB Invalidate Single) instruction • *MACRO,* 10–47
- TEF option • *File Def Language,* FDL–25
- Template class driver • *Device Support (A),* 17–9
  - listing of • *Device Support (A),* B–1 to B–35
- Template device • *Device Support (A),* 11–12
- Template for a device driver • *Device Support (A),* A–1 to A–10
- Template UCB • *Device Support (B),* 1–78
- TEMPORARY attribute • *File Def Language,* FDL–24
- Temporary file • *Convert,* CONV–27; *File Def Language,* FDL–19, FDL–20
- Temporary file delete option
  - See FAB$V_TMD option
- Temporary file option
  - See FAB$V_TMP option
- Temporary mailbox • *I/O User's I,* 7–4
- Temporary option • *File Applications,* 4–27
  - delete option • *File Applications,* 4–27
- /TEMPORARY qualifier • *Debugger,* CD–134, CD–191, CD–201
- TEMPORARY secondary attribute • *File Applications,* 4–27
- Terminal • *Device Support (B),* 1–74, 1–76
  - See also Terminal class driver
  - See also Terminal controller
  - See also Terminal port driver
  - See also Terminal UCB extension
  - ANSI CRT terminal • *I/O User's I,* 8–22
  - autobaud detection • *I/O User's I,* 8–19, 8–22
  - baud rate • *I/O User's I,* 8–19, 8–22, 8–40
  - behavior • *VAXTPU,* C–1
  - bell (Ctrl/G) • *I/O User's I,* 8–9
  - broadcast message • *I/O User's I,* 8–18, 8–21, 8–23, 8–46
  - carriage control • *I/O User's I,* 8–36
  - characteristic
    - See Terminal characteristic
  - command line editing • *I/O User's I,* 8–3, 8–34
  - command recall (Ctrl/B) • *I/O User's I,* 8–3, 8–6

- Terminal (Cont.)
  - control and data signals • *I/O User's I,* 8–16
  - control characters • *I/O User's I,* 8–4, 8–4 to 8–6, 8–9, 8–27
    - numeric values • *I/O User's I,* B–1
  - control sequences • *I/O User's I,* 8–8
  - cursor movement • *I/O User's I,* 8–3, 8–5, 8–22
  - DEC_CRT2 • *VAXTPU,* C–3
  - delete character • *I/O User's I,* 8–3
  - delete line (Ctrl/U) • *I/O User's I,* 8–5, 8–27
  - detached • *Device Support (B),* 1–75
  - device characteristics • *I/O User's I,* 8–20
    - categories • *I/O User's I,* 8–25
    - changing • *I/O User's I,* 8–42
    - extended • *I/O User's I,* 8–22
  - dial-up
    - characteristic • *I/O User's I,* 8–22
    - lines • *I/O User's I,* 8–13, 8–23, 8–42
    - support • *I/O User's I,* 8–13
  - DIGITAL CRT terminal • *I/O User's I,* 8–23
  - discard output (Ctrl/O) • *I/O User's I,* 8–5, 8–27, 8–35
  - driver • *I/O User's I,* 8–1
  - duplex modes • *I/O User's I,* 8–10, 8–13
  - enable Ctrl/C AST • *I/O User's I,* 8–42
  - enable Ctrl/Y AST • *I/O User's I,* 8–42
  - escape sequences • *I/O User's I,* 8–7, 8–57
    - ANSI • *I/O User's I,* B–9
    - Digital-private • *I/O User's I,* B–9
    - overflow size (item code) • *I/O User's I,* 8–30
  - extended characteristics • *I/O User's I,* 8–22
  - fallback conversion • *I/O User's I,* 8–11, 8–24, 8–42
  - features • *I/O User's I,* 8–2
  - form feed • *I/O User's I,* 8–21, 8–35
  - frame size • *I/O User's I,* 8–41
  - function codes • *I/O User's I,* 8–26, A–8
  - function modifiers
    - See also Terminal, item codes
    - IO$M_BRDCST • *I/O User's I,* 8–46, 8–55
    - IO$M_BREAKTHRU • *I/O User's I,* 8–10, 8–35
    - IO$M_CANCTRLO • *I/O User's I,* 8–5, 8–35
    - IO$M_CTRLCAST • *I/O User's I,* 8–42
    - IO$M_CTRLYAST • *I/O User's I,* 8–5, 8–13, 8–42
    - IO$M_CVTLOW • *I/O User's I,* 8–27
    - IO$M_DSABLMBX • *I/O User's I,* 8–27
    - IO$M_ENABLMBX • *I/O User's I,* 8–35
    - IO$M_ESCAPE • *I/O User's I,* 8–7, 8–27
    - IO$M_EXTEND • *I/O User's I,* 8–27, 8–29
    - IO$M_HANGUP • *I/O User's I,* 8–42

- Terminal
  - function modifiers (Cont.)
    - IO$M_INCLUDE • *I/O User's I,* 8–19, 8–43, 8–46
    - IO$M_LOOP • *I/O User's I,* 8–45
    - IO$M_LT_CONNECT • *I/O User's I,* 8–49
    - IO$M_LT_DISCON • *I/O User's I,* 8–49
    - IO$M_LT_MAP_PORT • *I/O User's I,* 8–49
      - P1 parameters • *I/O User's I,* 8–50
    - IO$M_LT_RATING • *I/O User's I,* 8–49
    - IO$M_MAINT • *I/O User's I,* 8–44, 8–45
    - IO$M_NOECHO • *I/O User's I,* 8–9, 8–10, 8–24, 8–27
    - IO$M_NOFILTR • *I/O User's I,* 8–27
    - IO$M_NOFORMAT • *I/O User's I,* 8–11, 8–35, 8–45
    - IO$M_OUTBAND • *I/O User's I,* 8–46
    - IO$M_PURGE • *I/O User's I,* 8–27
    - IO$M_RD_MODEM • *I/O User's I,* 8–54
    - IO$M_REFRESH • *I/O User's I,* 8–36
    - IO$M_SET_MODEM • *I/O User's I,* 8–44
    - IO$M_TIMED • *I/O User's I,* 8–27
    - IO$M_TRMNOECHO • *I/O User's I,* 8–28
    - IO$M_TT_ABORT • *I/O User's I,* 8–19, 8–46
    - IO$M_TYPEAHDCNT • *I/O User's I,* 8–54
    - IO$M_UNLOOP • *I/O User's I,* 8–45
  - hang up • *I/O User's I,* 8–13, 8–17, 8–18, 8–23, 8–24, 8–42, 8–52
  - I/O function for • *Device Support (B),* 1–40
  - I/O functions
    - CTDRIVER • *I/O User's I,* 8–35
    - IO$_READLBLK • *I/O User's I,* 8–26
    - IO$_READPROMPT • *I/O User's I,* 8–26, 8–27
    - IO$_READVBLK • *I/O User's I,* 8–26
    - IO$_SENSECHAR • *I/O User's I,* 8–53
    - IO$_SENSEMODE • *I/O User's I,* 8–53
    - IO$_SETCHAR • *I/O User's I,* 8–38
    - IO$_SETMODE • *I/O User's I,* 8–38
    - IO$_TTY_PORT • *I/O User's I,* 8–49
    - IO$_WRITELBLK • *I/O User's I,* 8–34
    - IO$_WRITEPBLK • *I/O User's I,* 8–34
    - IO$_WRITEVBLK • *I/O User's I,* 8–34
  - I/O status block • *I/O User's I,* 8–56
  - initiate login • *I/O User's I,* 8–9
  - input processing • *I/O User's I,* 8–3
  - insert/overstrike (Ctrl/A) • *I/O User's I,* 8–3, 8–6
  - interrupt (Ctrl/Y) • *I/O User's I,* 8–5
  - item codes • *I/O User's I,* 8–30 to 8–33
  - itemlist read • *I/O User's I,* 8–29
    - example • *I/O User's I,* 8–70
    - item codes • *I/O User's I,* 8–30 to 8–33

- Terminal
  - itemlist read (Cont.)
    - item descriptor • *I/O User's I*, 8–30
  - LAT line • *I/O User's I*, 8–1
  - LAT port driver • *I/O User's I*, 8–48
    - application services creation • *I/O User's I*, 8–51
    - example • *I/O User's I*, 8–74
    - I/O functions • *I/O User's I*, 8–49
  - LAT rejection codes • *I/O User's I*, 8–58
  - line editing • *I/O User's I*, 8–3, 8–23
    - See also Terminal, item codes
  - line feed • *I/O User's I*, 8–35
  - line terminators • *I/O User's I*, 8–9
  - mailbox • *I/O User's I*, 8–17, 8–35
    - message format • *I/O User's I*, 8–18
    - message types • *I/O User's I*, 8–18
  - modem
    - characteristic • *I/O User's I*, 8–21
    - control signals • *I/O User's I*, 8–16
    - data signals • *I/O User's I*, 8–16
    - protocol • *I/O User's I*, 8–14
    - sense signals • *I/O User's I*, 8–54
    - signal control • *I/O User's I*, 8–13
  - modem signal control • *I/O User's I*, 8–13
  - no type-ahead • *I/O User's I*, 8–21
  - out-of-band
    - See also Out-of-band AST
    - characters • *I/O User's I*, 8–19
  - output
    - CTDRIVER • *I/O User's I*, 8–11
    - RTPAD • *I/O User's I*, 8–11
    - SET HOST • *I/O User's I*, 8–11
  - output formatting • *I/O User's I*, 8–11, 8–25
  - output processing • *I/O User's I*, 8–10
  - page length and width • *I/O User's I*, 8–40, 8–53
  - parity flag • *I/O User's I*, 8–41
  - pasthru mode • *I/O User's I*, 8–9, 8–11, 8–24, 8–27
  - process preservation • *I/O User's I*, 8–17
  - programming examples • *I/O User's I*, 8–59
  - protocol • *I/O User's I*, 8–14
  - read function • *I/O User's I*, 8–26
    - arguments • *I/O User's I*, 8–26
    - function modifiers • *I/O User's I*, 8–27
    - itemlist read • *I/O User's I*, 8–29
    - terminating • *I/O User's I*, 8–26
    - terminators • *I/O User's I*, 8–28
    - with timeout • *I/O User's I*, 8–26, 8–27
  - read verify • *I/O User's I*, 8–6, 8–33
    - example • *I/O User's I*, 8–70
  - receive speed • *I/O User's I*, 8–40
- Terminal (Cont.)
  - redirected • *Device Support (B)*, 1–75
  - redisplay data (Ctrl/R) • *I/O User's I*, 8–6, 8–27
  - ReGIS graphics • *I/O User's I*, 8–24
  - restart data (Ctrl/Q) • *I/O User's I*, 8–6
  - restoring width • *VAXTPU*, A–5
  - sense characteristics function • *I/O User's I*, 8–53
  - sense mode function • *I/O User's I*, 8–53
  - separate, for debugger input/output • *Debugger*, 9–5
  - serial line multiplexer • *I/O User's I*, 8–1
  - set characteristics function • *I/O User's I*, 8–38
    - arguments • *I/O User's I*, 8–39
  - set mode function • *I/O User's I*, 8–38
    - arguments • *I/O User's I*, 8–39
  - SET TERMINAL DCL command • *I/O User's I*, 8–4, 8–19, 8–25
  - setting • *VAXTPU*, C–1 to C–3
    - AUTO_REPEAT • *VAXTPU*, C–2
    - auxiliary keypad • *VAXTPU*, C–2
    - 132 columns • *VAXTPU*, C–2
    - control sequence introducer • *VAXTPU*, C–2
    - CSI • *VAXTPU*, C–2
    - cursor • *VAXTPU*, C–2
    - DEC_CRT • *VAXTPU*, C–2
    - edit mode • *VAXTPU*, C–2
    - eightbit characters • *VAXTPU*, C–2
    - scrolling • *VAXTPU*, C–3
    - video attributes • *VAXTPU*, C–3
    - wrap • *VAXTPU*, C–4
  - SIXEL graphics • *I/O User's I*, 8–24
  - special operating modes • *I/O User's I*, 8–10
  - status (Ctrl/T) • *I/O User's I*, 8–6
  - status returns • *I/O User's I*, A–9
  - stop data (Ctrl/S) • *I/O User's I*, 8–6
  - support • *VAXTPU*, C–1
  - supported devices • *I/O User's I*, 8–1
  - support for SET and SHOW TERMINAL commands • *RTL Screen Management*, 5–24
  - SYS$GETDVI returns • *I/O User's I*, 8–20
  - system password • *I/O User's I*, 8–24
  - tab
    - Ctrl/I • *I/O User's I*, 8–6
    - mechanical • *I/O User's I*, 8–21
    - stops • *I/O User's I*, 8–35
  - terminator mask • *I/O User's I*, 8–28, 8–29
  - time (Ctrl/T) • *I/O User's I*, 8–6
  - transmit speed • *I/O User's I*, 8–40
  - TTY_DIALTYPE SYSGEN parameter • *I/O User's I*, 8–13, 8–14, 8–16
  - type-ahead • *I/O User's I*, 8–8, 8–17, 8–21, 8–54
    - alternate buffer • *I/O User's I*, 8–22

- Terminal (Cont.)
  - unsolicited data • *I/O User's I*, 8–17
  - width
    - restoring • *VAXTPU*, A–5
  - write breakthrough function • *I/O User's I*, 8–36
  - write function • *I/O User's I*, 8–34
    - carriage control • *I/O User's I*, 8–36
    - function modifiers • *I/O User's I*, 8–35
  - XON/XOFF control • *I/O User's I*, 8–24
- Terminal characteristic
  - ANSI CRT • *I/O User's I*, 8–22
  - ASCII (8-bit) code • *I/O User's I*, 8–21
  - baud rate • *I/O User's I*, 8–22
  - block mode • *I/O User's I*, 8–23
  - dial-up line • *I/O User's I*, 8–23
  - dial-up terminal • *I/O User's I*, 8–22
  - DIGITAL CRT • *I/O User's I*, 8–23
  - DMA mode • *I/O User's I*, 8–23
  - edit • *I/O User's I*, 8–23
  - extended characteristics • *I/O User's I*, 8–22
  - local echo • *I/O User's I*, 8–24
  - modem • *I/O User's I*, 8–21
  - modify hang up • *I/O User's I*, 8–24
  - no echo • *I/O User's I*, 8–21
  - no type ahead • *I/O User's I*, 8–21
  - pasthru mode • *I/O User's I*, 8–24
  - ReGIS graphics • *I/O User's I*, 8–24
  - remote terminal • *I/O User's I*, 8–22
  - secure • *I/O User's I*, 8–24
  - set speed • *I/O User's I*, 8–24
  - SIXEL graphics • *I/O User's I*, 8–24
  - system password • *I/O User's I*, 8–24
  - XON/XOFF • *I/O User's I*, 8–24
- Terminal characteristics • *Programming Resources*, 7–51
- Terminal class driver • *Device Support (A)*, 18–1 to 18–23
  - binding to port driver • *Device Support (A)*, 18–9 to 18–10; *Device Support (B)*, 2–8
  - service routines • *Device Support (A)*, 18–19 to 18–23
  - structure • *Device Support (A)*, 18–7
- Terminal controller • *Device Support (B)*, 1–21
- Terminal device record-processing option • *RMS*, 7–18
- Terminal device width • *Programming Resources*, 7–6
- Terminal echo • *Programming Resources*, 7–40
  - disabling • *Programming Resources*, 7–41
- Terminal emulator • *VAXTPU*, 6–4
- Terminal extended address block
  - See XABTRM block
- Terminal I/O • *Modular Procedures*, 2–17
- Terminal I/O (Cont.)
  - example • *System Services Intro*, 7–20
- Terminal key
  - defining for SDA • *System Dump Analyzer*, SDA–43
- Terminal port driver • *Device Support (A)*, 18–1 to 18–23; *Device Support (B)*, 2–7
  - aborting output activity in • *Device Support (A)*, 18–16
  - binding to class driver • *Device Support (A)*, 18–9 to 18–10; *Device Support (B)*, 2–8
  - canceling I/O request in • *Device Support (A)*, 18–17
  - control flags • *Device Support (B)*, 1–89
  - detecting an error on terminal line in • *Device Support (A)*, 18–22
  - disconnecting a process from a terminal in • *Device Support (A)*, 18–19
  - forking in • *Device Support (A)*, 18–14, 18–20
  - implementing modem functions in • *Device Support (A)*, 18–15
  - initiate routines • *Device Support (A)*, 18–13 to 18–16
  - managing data set state transitions in • *Device Support (A)*, 18–20
  - obtaining characters for output in • *Device Support (A)*, 18–20
  - passing input characters to class driver from • *Device Support (A)*, 18–21
  - resuming stopped output in • *Device Support (A)*, 18–17
  - service routines • *Device Support (A)*, 18–16 to 18–18
  - starting output on an inactive line in • *Device Support (A)*, 18–16
  - startup routines • *Device Support (A)*, 18–12 to 18–13
  - stopping output in • *Device Support (A)*, 18–17
  - structure • *Device Support (A)*, 18–7
  - using input flow control character in • *Device Support (A)*, 18–17, 18–18
- Terminal read operation
  - RAB$L_ROP field options • *RMS*, 18–2
- Terminal screen size
  - See Screen size
- Terminal support • *VAXTPU*, 1–8
- Terminal timeout • *Programming Resources*, 7–41
- Terminal UCB extension • *Device Support (A)*, 18–2 to 18–3; *Device Support (B)*, 1–69, 1–84 to 1–91
  - initializing • *Device Support (A)*, 18–22
  - remote • *Device Support (B)*, 1–75

- /TERMINATE qualifier • *Debugger*, 8–8, CD–51; *System Dump Analyzer*, SDA–45
- Terminating
  - DELTA
    - See Exiting
  - DELTA/XDELTA commands • *Delta/XDelta*, DELTA–27
- Terminating access to PPL$ • *RTL Parallel Processing*, 2–2
- /TERMINATING qualifier • *Debugger*, 10–14, CD–18, CD–31, CD–134, CD–191
- Termination
  - debugging session • *Debugger*, 3–5, 10–9, CD–94, CD–112
    - with DECwindows • *Debugger*, 1–21
  - execution of handlers at • *Debugger*, 9–16
  - multiprocess program • *Debugger*, 10–9, 10–10, 10–14
- Termination mailbox • *System Services Intro*, 7–34, 8–18
- Termination message
  - format • *System Services*, SYS–96
- Termination of subordinate abnormally
  - notification of • *RTL Parallel Processing*, 2–3
- Terminator • *RTL Screen Management*, 3–3
  - See also Input/output
  - codes • *RTL Screen Management*, 3–4
  - echo • *Programming Resources*, 7–24
  - file • *Programming Resources*, 7–54
  - record • *Programming Resources*, 7–53
- Terminator character bit mask • *I/O User's I*, 8–28
- Terminator variations • *File Applications*, 3–10
- Term in MACRO statement • *MACRO*, 3–9
- TERMTABLE.EXE • *RTL Screen Management*, 5–1, 5–17
  - creating • *RTL Screen Management*, 5–22
- TERMTABLE.TXT • *RTL Screen Management*, 5–1, 5–17
- Test and set instructions • *Modular Procedures*, 3–23
- Testing new procedures • *Modular Procedures*, 4–1
  - black box • *Modular Procedures*, 4–2
  - integration • *Modular Procedures*, 4–1, 4–5
  - language independence • *Modular Procedures*, 4–1, 4–4
  - modularity • *Modular Procedures*, 4–1
  - reentrancy • *Modular Procedures*, 4–6
  - regression • *Modular Procedures*, 6–1
  - unit • *Modular Procedures*, 4–1
  - white box • *Modular Procedures*, 4–3
- Text
  - compression of • *Utility Routines*, DCX–1
- Text editor
  - creating command procedure with • *Patch*, PAT–5
  - to create FDL files • *File Def Language*, FDL–42
- Text entry
  - See Explanatory text
- TEXT keyword • *VAXTPU*, 7–483
- Text library • *Programming Resources*, 1–18; *Librarian*, LIB–1
  - character case in • *Librarian*, LIB–2
- Text manipulation
  - built-in procedures
    - APPEND_LINE • *VAXTPU*, 7–28
    - BEGINNING_OF • *VAXTPU*, 7–37
    - CHANGE_CASE • *VAXTPU*, 7–44
    - COPY_TEXT • *VAXTPU*, 7–53
    - CREATE_BUFFER • *VAXTPU*, 7–58
    - EDIT • *VAXTPU*, 7–111
    - END_OF • *VAXTPU*, 7–115
    - ERASE • *VAXTPU*, 7–117
    - ERASE_CHARACTER • *VAXTPU*, 7–119
    - ERASE_LINE • *VAXTPU*, 7–121
    - FILE_PARSE • *VAXTPU*, 7–140
    - FILE_SEARCH • *VAXTPU*, 7–143
    - FILL • *VAXTPU*, 7–146
    - MOVE_TEXT • *VAXTPU*, 7–280
    - READ_FILE • *VAXTPU*, 7–297
    - SEARCH • *VAXTPU*, 7–327
    - SEARCH_QUIETLY • *VAXTPU*, 7–332
    - SELECT • *VAXTPU*, 7–337
    - SELECT_RANGE • *VAXTPU*, 7–340
    - SPLIT_LINE • *VAXTPU*, 7–518
    - TRANSLATE • *VAXTPU*, 7–526
    - WRITE_FILE • *VAXTPU*, 7–543
- Text processing • *Programming Resources*, 1–3
  - EVE editor • *Programming Resources*, 1–5
- Text processing routines
  - See VAXTPU routines
- /TEXT qualifier • *Librarian*, LIB–44; *Message*, MSG–14
- "Text" string constant parameter to GET_INFO • *VAXTPU*, 7–225
- Textual operator • *MACRO*, 3–12
- T field in symbolic offset
  - for specifying varying field length • *RMS*, 2–3
- %THEN lexical keyword • *VAXTPU*, 3–36
- Third-party SCSI class driver
  - cancel-I/O routine of • *Device Support (A)*, 17–28
  - components • *Device Support (A)*, 17–24 to 17–28
  - data definitions • *Device Support (A)*, 17–24
  - debugging • *Device Support (A)*, 17–31 to 17–43
  - driver prologue table • *Device Support (A)*, 17–25

Third-party SCSI class driver (Cont.)
- error logging • *Device Support (A),* 17–20 to 17–22
- loading • *Device Support (A),* 17–30
- maintaining local context of • *Device Support (A),* 17–19 to 17–20
- receiving notification of asynchronous events on target • *Device Support (A),* 17–28 to 17–30; *Device Support (B),* 2–70, 2–73 to 2–90
- register dumping routine of • *Device Support (A),* 17–21, 17–28
- start-I/O routine of • *Device Support (A),* 17–27 to 17–28
- unit initialization routine of • *Device Support (A),* 17–26 to 17–27
- writing • *Device Support (A),* 17–1 to 17–43

Thrashing
- magnetic tape • *I/O User's I,* 6–10

Threads of execution • *Modular Procedures,* 3–19

Time • *Programming Resources,* 3–23
- See also Current
- absolute • *System Services Intro,* 9–2
- conversion • *System Services Intro,* 9–1
- converting ASCII to binary • *System Services Intro,* 9–3
- converting binary to ASCII string • *System Services,* SYS–18
- converting binary to numeric • *System Services,* SYS–366
- delta • *System Services Intro,* 9–2
- getting current system • *System Services Intro,* 9–2; *System Services,* SYS–314
- inserting with FAO • *VAXTPU,* 7–138
- inserting with MESSAGE • *VAXTPU,* 7–268
- inserting with MESSAGE_TEXT • *VAXTPU,* 7–271
- internal format • *Programming Resources,* 3–23
- numeric and ASCII • *System Services Intro,* 9–7
- obtaining
  - using SYS$ASCTIM • *Programming Resources,* 3–24
  - using SYS$BINTIM • *Programming Resources,* 3–24
  - using SYS$FAO • *Programming Resources,* 3–24
  - using SYS$GETTIM • *Programming Resources,* 3–24
- reading system • *Device Support (B),* 2–52
- setting system • *System Services Intro,* 9–8; *System Services,* SYS–404
- system format • *System Services Intro,* 9–2

TIMEDWAIT macro • *Device Support (B),* 2–92 to 2–93
- See also TIMEWAIT macro

TIMEDWAIT macro (Cont.)
- example • *Device Support (B),* 2–93

"Timed_message" string constant parameter to GET_INFO • *VAXTPU,* 7–207

Time manipulation • *Programming Resources,* 3–24
- converting • *Programming Resources,* 3–24
- formatting • *Programming Resources,* 3–24
- using LIB$ADDX • *Programming Resources,* 3–24
- using LIB$ADD_TIME • *Programming Resources,* 3–24
- using LIB$DAY • *Programming Resources,* 3–25
- using LIB$MULT_DELTA_TIME • *Programming Resources,* 3–24
- using LIB$SUBX • *Programming Resources,* 3–24
- using LIB$SUB_TIME • *Programming Resources,* 3–24

Timeout • *Device Support (B),* 1–78, 2–104
- caused by power failure recovery procedure • *Device Support (A),* 10–5
- detecting • *Device Support (B),* 1–79
- disabling • *Device Support (A),* 4–17, 10–1; *Device Support (B),* 2–43, 3–30
- due time • *Device Support (B),* 1–79
- expected • *Device Support (B),* 1–77, 3–105
- for SCSI device • *I/O User's I,* 11–8, 11–14; *Device Support (A),* 17–11, 17–12; *Device Support (B),* 2–89
- logging • *Device Support (A),* 10–6, 11–10

Timeout enable bit
- See UCB$V_TIM

Timeout field
- See RAB$B_TMO field

Timeout handling routine • *Device Support (A),* 1–4, 3–8, 9–4, 10–4 to 10–7, 11–8; *Device Support (B),* 2–104, 4–5
- aborting an I/O request in • *Device Support (A),* 10–6
- address • *Device Support (A),* 8–7, 10–1; *Device Support (B),* 4–19
- context • *Device Support (A),* 10–4; *Device Support (B),* 4–19
- entry point • *Device Support (B),* 4–19
- exit method • *Device Support (B),* 4–20
- functions • *Device Support (A),* 10–5; *Device Support (B),* 4–20
- input • *Device Support (B),* 4–20
- register usage • *Device Support (B),* 4–19
- retrying an I/O operation in • *Device Support (A),* 10–5 to 10–6
- synchronization requirements • *Device Support (A),* 3–22, E–12; *Device Support (B),* 4–19

Timeout interval • *Device Support (B),* 2–104
- specifying • *Device Support (A),* 10–4

Timeout option
See RAB$V_TMO option
TIMEOUT_ENABLE attribute • *File Def Language*, FDL–13
TIMEOUT_PERIOD attribute • *File Def Language*, FDL–13
TIMEOUT_PERIOD secondary attribute • *File Applications*, 7–12
/TIME qualifier • *System Dump Analyzer*, SDA–52
Timer
See Interval clock
See Software timer
deallocating • *Programming Resources*, 3–21
initializing • *Programming Resources*, 3–20
obtaining statistics • *Programming Resources*, 3–20, 3–21
setting • *System Services*, SYS–406
statistics
buffer input/output • *Programming Resources*, 3–20
CPU time • *Programming Resources*, 3–20
direct input/output • *Programming Resources*, 3–20
elapsed time • *Programming Resources*, 3–20
page fault • *Programming Resources*, 3–20
TIMER keyword • *VAXTPU*, 7–486
Timer queue • *Device Support (A)*, 3–14, E–13; *Device Support (B)*, 3–29, 3–48
Timer queue element
See TQE
Timer request • *System Services Intro*, 9–4
canceling • *System Services Intro*, 9–6; *System Services*, SYS–42
TIMER spin lock • *Device Support (A)*, 3–8, 3–13, E–13; *Device Support (B)*, 3–29, 3–48
TIMEWAIT macro • *Device Support (B)*, 2–94
See also TIMEDWAIT macro
example • *Device Support (B)*, 2–95
time_name data type • *Routines Intro*, A–12t
/TIME_SLICE qualifier • *Debugger*, CD–183, CD–254
TIMOUT processor state • *Device Support (B)*, 1–16
TITLE attribute • *File Def Language*, FDL–2, FDL–39
Title bar widget • *VAXTPU*, 4–16
.TITLE directive • *Programming Resources*, 9–9; *MACRO*, 6–95
Title directive (.TITLE)
in message source file • *Message*, MSG–7, MSG–28
Title listing control directive
(.TITLE) • *MACRO*, 6–95
/TMASK qualifier • *Debugger*, 11–13, CD–88
TMD option • *File Def Language*, FDL–24
TMO option • *File Def Language*, FDL–13
TMP option • *File Def Language*, FDL–20
Tools to aid in application development • *Modular Procedures*, 1–12
TOP command • *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–34
/TOP qualifier • *Debugger*, CD–119
Total buckets reclaimed • *Convert*, CONV–24
Total buckets scanned • *Convert*, CONV–24
Total exception records • *Convert*, CONV–24
Total key size field
See XAB$B_TKS field
Total records processed • *Convert*, CONV–24
Total valid records • *Convert*, CONV–24
TPT option • *File Def Language*, FDL–13
TPU
See VAXTPU
TPU$CLEANUP routine • *Utility Routines*, TPU–26
TPU$CLIPARSE routine • *Utility Routines*, TPU–29
TPU$CLOSE_TERMINAL routine • *Utility Routines*, TPU–30
TPU$COMMAND logical name • *VAXTPU*, 4–21, 5–6
TPU$CONTROL routine • *Utility Routines*, TPU–31
TPU$DEBUG logical name • *VAXTPU*, 5–8
TPU$EDIT routine • *Utility Routines*, TPU–32
TPU$EXECUTE_COMMAND routine • *Utility Routines*, TPU–34
TPU$EXECUTE_INIFILE routine • *Utility Routines*, TPU–35
TPU$FILEIO routine • *Utility Routines*, TPU–37
TPU$HANDLER routine • *Utility Routines*, TPU–41
TPU$INITIALIZE routine • *Utility Routines*, TPU–43
TPU$INIT_PROCEDURE procedure • *VAXTPU*, 4–22, 4–28
TPU$K_DISJOINT constant • *VAXTPU*, 7–198, 7–368
TPU$K_INVISIBLE constant • *VAXTPU*, 7–198, 7–368
TPU$K_OFF_LEFT constant • *VAXTPU*, 7–198, 7–368
TPU$K_OFF_RIGHT constant • *VAXTPU*, 7–198, 7–368
TPU$K_UNMAPPED constant • *VAXTPU*, 7–198, 7–368
TPU$LOCAL_INIT procedure • *VAXTPU*, 4–29
TPU$LOCAL_INIT_PROCEDURE procedure • *VAXTPU*, 4–23
TPU$MESSAGE routine • *Utility Routines*, TPU–48
TPU$PARSEINFO routine • *Utility Routines*, TPU–49
TPU$SECTION logical name • *VAXTPU*, 4–21, 4–27, 5–16

TPU$STACKOVER status
correcting • *VAXTPU*, 4–2
TPU$TPU routine • *Utility Routines*, TPU–50
TPU$WIDGET_INTEGER_CALLBACK callback routine • *VAXTPU*, 4–9, 4–10
TPU$WIDGET_STRING_CALLBACK callback routine • *VAXTPU*, 4–9, 4–10
TPU$X_MESSAGE_BUFFER variable • *VAXTPU*, 4–29
TPU$X_SHOW_BUFFER variable • *VAXTPU*, 4–29
TPU$X_SHOW_WINDOW variable • *VAXTPU*, 4–29
TPU$_UNKLEXICAL error message • *VAXTPU*, 3–38
TPU command • *VAXTPU*, 4–19
TPU debugger • *VAXTPU*, 4–33 to 4–37
ATTACH command • *VAXTPU*, 4–36
CANCEL BREAKPOINT command • *VAXTPU*, 4–36
DEBUGON procedure • *VAXTPU*, 4–35
DEPOSIT command • *VAXTPU*, 4–36
DISPLAY SOURCE command • *VAXTPU*, 4–36
EXAMINE command • *VAXTPU*, 4–36
GO command • *VAXTPU*, 4–34, 4–36
HELP command • *VAXTPU*, 4–36
invoking • *VAXTPU*, 4–33
QUIT command • *VAXTPU*, 4–36
SCROLL command • *VAXTPU*, 4–37
SET BREAKPOINT command • *VAXTPU*, 4–34, 4–37
SET WINDOW command • *VAXTPU*, 4–37
SHIFT command • *VAXTPU*, 4–37
SHOW BREAKPOINTS command • *VAXTPU*, 4–37
SPAWN command • *VAXTPU*, 4–37
STEP command • *VAXTPU*, 4–35, 4–37
TPU command • *VAXTPU*, 4–37
TQE$B_RQTYPE • *Device Support (B)*, 3–48
TQE$Q_TIME • *Device Support (B)*, 3–29
TQE (timer queue element)
calling a driver from • *Device Support (A)*, E–15
expiration time • *Device Support (A)*, 3–8; *Device Support (B)*, 3–29
inserting in timer queue • *Device Support (B)*, 3–29
removing in timer queue • *Device Support (B)*, 3–48
TQELM (timer queue entry limit) quota
effect of canceling timer request • *System Services*, SYS–43
Traceback • *MACRO*, 6–23
compiler option • *Debugger*, 5–3
link option • *Debugger*, 5–4
SHOW CALLS display • *Debugger*, 2–14
Traceback handler • *Programming Resources*, 9–5, 9–13
TRACEBACK keyword • *VAXTPU*, 7–488
/TRACEBACK qualifier • *Debugger*, 3–3, 5–4, 5–5; *Linker*, LINK–20
shareable image • *Debugger*, 5–13
"Traceback" string constant parameter to GET_INFO • *VAXTPU*, 7–207
Tracepoint
canceling • *Debugger*, 3–16, CD–30
defined • *Debugger*, 3–10
delayed triggering of • *Debugger*, 3–14, CD–188
displaying • *Debugger*, CD–257
DO clause • *Debugger*, 3–14
exception • *Debugger*, 9–10, CD–187
on activation (multiprocess program) • *Debugger*, 10–14
on termination (image exit) • *Debugger*, 10–14
on vector instruction • *Debugger*, 11–3
predefined • *Debugger*, 10–14
setting • *Debugger*, 3–10, CD–187
source display at • *Debugger*, 6–7
WHEN clause • *Debugger*, 3–14
with DECwindows • *Debugger*, 1–24
Trace trap enable (T) • *MACRO*, 8–15
Track • *File Applications*, 1–5
size • *File Applications*, 3–13
Trailing numeric string
data type • *MACRO*, 8–8
transaction_id data type • *Routines Intro*, A–12t
Transfer address • *Debugger*, 3–1, 9–7
.TRANSFER directive • *Linker*, 4–8; *MACRO*, 6–96
Transfer from disk volumes • *File Def Language*, FDL–23
Transfers, far-end DR device (DR32) • *I/O User's II*, 4–3
Transfer vector • *Programming Resources*, 5–3
See also Shareable image
advantage of • *Linker*, 4–6
changing • *Modular Procedures*, 6–6
coded for procedure call • *Linker*, 4–8
coded for subroutine call • *Linker*, 4–8
compiling • *Programming Resources*, 5–6
creating • *Programming Resources*, 5–6; *Modular Procedures*, 5–5; *Linker*, 4–7
deleting • *Programming Resources*, 5–4
example • *Linker*, 1–10
for upward compatibility • *Linker*, 1–11, 4–9
placement of • *Programming Resources*, 5–3
purpose of • *Linker*, 4–5
reasons for using • *Programming Resources*, 5–4
recommended length of • *Linker*, 4–7

Transfer vector (Cont.)
- updating • *Modular Procedures*, 6–3

TRANSLATE built-in procedure • *VAXTPU*, 7–526 to 7–529

Translation
- logical to physical • *I/O User's I*, 3–18
- of addresses to symbols • *Patch*, PAT–13
- of symbols to addresses • *Patch*, PAT–13

Translation buffer
- See TB
- invalidating • *Device Support (A)*, E–15; *Device Support (B)*, 2–41 to 2–42

Translation mode card
- 026 punch mode • *I/O User's I*, 2–2
- 029 punch mode • *I/O User's I*, 2–2

/TRANSLATION_ATTRIBUTES qualifier • *File Applications*, 5–7, 6–15

Trap
- arithmetic • *MACRO*, E–1
- arithmetic type code • *MACRO*, E–1
- change mode • *MACRO*, E–8
- decimal
  - string overflow • *MACRO*, E–3
- decimal overflow • *MACRO*, 8–16
- divide by zero • *MACRO*, 8–16
- floating
  - divide-by-zero • *MACRO*, E–2
  - overflow • *MACRO*, E–2
  - underflow • *MACRO*, E–3
- integer
  - divide-by-zero • *MACRO*, E–2
  - overflow • *MACRO*, E–2
- integer overflow • *MACRO*, 8–15
- subscript-range • *MACRO*, E–3
- trace • *MACRO*, 8–15

Tree structure • *File Applications*, 10–11
- of indexed file • *File Applications*, 10–19
- of relative file • *File Applications*, 10–16
- of sequential file • *File Applications*, 10–12

TRM$M_TM_ESCAPE • *Programming Resources*, 7–25

TRM$M_TM_NOECHO • *Programming Resources*, 7–25

TRM$M_TM_TRMNOECHO • *Programming Resources*, 7–24

TRUE logical value • *File Def Language*, FDL–2

Truncate at end-of-file option
- See FAB$V_TEF option

TRUNCATE attribute • *File Def Language*, FDL–3

Truncate-on-put option
- See also RAB$V_TPT option
- access requirement • *File Applications*, 7–7

Truncate option
- See FAB$V_TRN option

/TRUNCATE qualifier • *Convert*, CONV–3, CONV–26

TRUNCATE secondary attribute • *File Applications*, 7–3

Truncatė service • *File Applications*, 8–5; *RMS*, RMS–97
- condition values • *RMS*, RMS–98
  - See also Completion status code
- control block input fields • *RMS*, RMS–98
- control block output fields • *RMS*, RMS–98
- effect on next-record position • *File Applications*, 8–16
- use restriction • *RMS*, RMS–97

Truncate subfunction • *I/O User's I*, 1–13

TRUNCATE_ON_CLOSE attribute • *File Def Language*, FDL–25

TRUNCATE_ON_PUT attribute • *File Def Language*, FDL–13

Truncation of floating-point value • *RTL Math*, 1–6

Truncation of records • *Convert*, CONV–3

TSTB (Test Byte) instruction • *MACRO*, 9–31

TSTD (Test D_floating) instruction • *MACRO*, 9–125

TSTF (Test F_floating) instruction • *MACRO*, 9–125

TSTG (Test G_floating) instruction • *MACRO*, 9–125

TSTH (Test H_floating) instruction • *MACRO*, 9–125

TSTL (Test Long) instruction • *MACRO*, 9–31

TSTW (Test Word) instruction • *MACRO*, 9–31

TTDRIVER.EXE • *Device Support (A)*, 18–1

TTY$V_PC_NOTIME • *Device Support (A)*, 18–16

TTY$V_PC_PORTFDT • *Device Support (A)*, 18–14

TTY$V_TP_ABORT • *Device Support (A)*, 18–18

$TTYDEFS macro • *Device Support (A)*, 18–2

$TTYMACS macro • *Device Support (A)*, 18–12; *Device Support (B)*, 2–7, 2–8, 2–98, 2–99, 2–100

$TTYMDMDEF macro • *Device Support (A)*, 18–20

$TTYMODEMDEF macro • *Device Support (A)*, 18–13

$TTYUCBDEF macro • *Device Support (B)*, 1–69

TT_CANCEL_CONTROL_O attribute • *File Def Language*, FDL–14

TT_PROMPT attribute • *File Def Language*, FDL–14

TT_PURGE_TYPE_AHEAD attribute • *File Def Language*, FDL–14

TT_READ_NOECHO attribute • *File Def Language*, FDL–14

TT_READ_NOFILTER attribute • *File Def Language*, FDL–14

TT_UPCASE_INPUT attribute • *File Def Language*, FDL–14

TU58 console bootstrap procedures • *Delta/XDelta,* DELTA–6
TU58 magnetic tape
  See Disk
Tuning • *File Applications,* 3–3, 10–26
  indexed files • *File Applications,* 3–15
  relative files • *File Applications,* 3–12
  sequential files • *File Applications,* 3–9, 3–10
Type
  See also Built-in value type
  address expression • *Debugger,* 4–4, 4–25
  array • *Debugger,* 4–17
  ASCII string • *Debugger,* 4–17, 4–28
  compiler generated • *Debugger,* 4–4, 4–15
  conversion, numeric • *Debugger,* 4–7
  current • *Debugger,* 4–25, CD–195, CD–259
  displaying • *Debugger,* CD–259
  integer • *Debugger,* 4–15, 4–27
  override • *Debugger,* 4–26, CD–195
  pointer • *Debugger,* 4–19
  real • *Debugger,* 4–15
  record • *Debugger,* 4–19
  scalar • *Debugger,* 4–15
  SET TYPE command • *Debugger,* 4–25, CD–195
  symbolic address expression • *Debugger,* 4–4
  VAX instruction • *Debugger,* 4–20
  vector register • *Debugger,* 11–7
Type-ahead
  See Terminal, type-ahead
Type-ahead buffer • *Programming Resources,* 7–39
TYPE attribute • *File Def Language,* FDL–28, FDL–29, FDL–30
TYPE clause
  definition of value types • *Command Def,* CDU–6
  for VALUE clause • *Command Def,* CDU–24, CDU–26, CDU–33, CDU–34
  with VALUE clause • *Command Def,* CDU–29
Type code field in allocation XAB
  See XAB$B_COD field
Type code field in date and time XAB
  See XAB$B_COD field
Type code field in file header characteristics XAB
  See XAB$B_COD field
Type code field in item list XAB
  See XAB$B_COD field
Type code field in key XAB
  See XAB$B_COD field
Type code field in protection XAB
  See XAB$B_COD field
Type code field in revision date and time XAB
  See XAB$B_COD field
Type code field in summary XAB
  See XAB$B_COD field
Type code field in terminal XAB
  See XAB$B_COD field
TYPE command • *Debugger,* 6–3, 7–6, CD–275
Type entry • *Routines Intro,* 1–8
"Type" GET_INFO request_string • *VAXTPU,* 7–165
TYPE keyword
  with FILE_PARSE • *VAXTPU,* 7–141
  with FILE_SEARCH • *VAXTPU,* 7–144
Type override • *Debugger,* 4–26, CD–33, CD–196, CD–259
/TYPE qualifier • *Debugger,* 4–28, CD–63, CD–89, CD–251; *System Dump Analyzer,* SDA–56, SDA–119
Types of libraries • *Librarian,* LIB–1

U

UAF (user authorization file)
  getting information about • *System Services,* SYS–315
  modifying • *System Services,* SYS–431
UBA (UNIBUS adapter) • *Device Support (A),* 1–11
  See also UNIBUS adapter
UBI (UNIBUS interface) • *Device Support (A),* 1–11
  See also UNIBUS adapter
UBMAPEXCED bugcheck • *Device Support (B),* 3–74, 3–78
UCB$B_DEVCLASS • *Device Support (A),* 6–3, 17–21, 17–25; *Device Support (B),* 2–25, 3–51
UCB$B_DEVTYPE • *Device Support (A),* 6–3, 17–21, 17–25; *Device Support (B),* 2–25, 3–51
UCB$B_DIPL • *Device Support (A),* 3–6, 6–2, 10–4; *Device Support (B),* 2–25
UCB$B_ERTCNT • *Device Support (A),* 10–3; *Device Support (B),* 3–69, 3–94
UCB$B_FIPL • *Device Support (B),* 1–73, 2–33
UCB$B_FLCK • *Device Support (A),* 3–6, 6–2, 10–1; *Device Support (B),* 2–25, 2–33
  initializing • *Device Support (A),* E–8
UCB$B_SLAVE • *Device Support (A),* 15–12 to 15–13
UCB$B_SLAVE+1 • *Device Support (A),* 15–12 to 15–13
UCB$B_TP_STAT • *Device Support (A),* 18–18
UCB$B_TT_DEPARI • *Device Support (A),* 18–22
UCB$B_TT_DETYPE • *Device Support (A),* 18–22

UCB$B_TT_MAINT • *Device Support (A)*, 18–15
UCB$B_TT_OUTYPE • *Device Support (A)*, 18–16, 18–21, 18–22, 18–23
UCB$B_TT_PARITY • *Device Support (A)*, 18–15, 18–22
UCB$L_AFFINITY • *Device Support (B)*, 3–71
UCB$L_CRB • *Device Support (A)*, 11–5, 15–13
UCB$L_DDB • *Device Support (A)*, 4–8
UCB$L_DDT • *Device Support (A)*, 18–9
UCB$L_DEVCHAR • *Device Support (A)*, 6–3, 11–9; *Device Support (B)*, 2–25
UCB$L_DLCK • *Device Support (A)*, 3–22
UCB$L_DUETIM • *Device Support (A)*, 4–16, 8–7, 10–5; *Device Support (B)*, 3–104, 3–105
UCB$L_EMB • *Device Support (A)*, 10–3; *Device Support (B)*, 3–8
UCB$L_FPC • *Device Support (A)*, 4–16, 4–17, 9–4, 10–1, 10–4
UCB$L_FR3 • *Device Support (A)*, 4–16, 4–17, 9–4, 10–1, 10–4
UCB$L_FR4 • *Device Support (A)*, 4–16, 4–17, 9–4, 10–1, 10–4
UCB$L_IOQFL • *Device Support (A)*, 10–3, E–14; *Device Support (B)*, 3–28
UCB$L_IRP • *Device Support (A)*, 4–5, 10–3; *Device Support (B)*, 3–71
UCB$L_LINK • *Device Support (A)*, 11–5
UCB$L_MAXBCNT • *Device Support (A)*, 17–14, 17–26
UCB$L_OPCNT • *Device Support (B)*, 3–5, 3–24, 3–94
  adjusted by IOC$REQCOM • *Device Support (B)*, 3–95
UCB$L_ORB • *Device Support (B)*, 1–44
UCB$L_PDT • *Device Support (A)*, 17–26
UCB$L_SCDT • *Device Support (A)*, 17–26
UCB$L_STS • *Device Support (A)*, 2–4, 8–5, 8–7
UCB$L_SVAPTE • *Device Support (A)*, 4–5, 8–2, 14–22, 15–3, 15–14, 16–19; *Device Support (B)*, 1–40, 3–71, 3–79
UCB$L_SVPN • *Device Support (B)*, 2–21, 3–67, 3–79
UCB$L_TT_CLASS • *Device Support (A)*, 18–9; *Device Support (B)*, 2–8
UCB$L_TT_GETNXT • *Device Support (A)*, 18–9
UCB$L_TT_LOGUCB • *Device Support (A)*, 18–22
UCB$L_TT_OUTADR • *Device Support (A)*, 18–16, 18–21, 18–22
UCB$L_TT_PORT • *Device Support (A)*, 18–9; *Device Support (B)*, 2–8
UCB$L_TT_PUTNXT • *Device Support (A)*, 18–9
UCB$L_TT_RTIMOU • *Device Support (A)*, 18–22
UCB$L_TT_WFLINK • *Device Support (A)*, 18–22
UCB$Q_DEVDEPEND • *Device Support (A)*, 6–3; *Device Support (B)*, 3–49, 3–51
UCB$V_BSY • *Device Support (A)*, 2–4, 4–5, 7–5, 10–4, 11–8; *Device Support (B)*, 3–28, 3–68, 4–5
UCB$V_CANCEL • *Device Support (A)*, 10–6, 10–7, 11–8; *Device Support (B)*, 3–68, 3–71, 4–5
UCB$V_DELMBX • *Device Support (A)*, 18–13
UCB$V_ECC • *Device Support (B)*, 3–67
UCB$V_ERLOGIP • *Device Support (A)*, 10–3, 11–10; *Device Support (B)*, 3–8, 3–95
UCB$V_INT • *Device Support (A)*, 8–7, 9–3, 9–7, 10–4, 15–10, 18–16
UCB$V_JOB • *Device Support (A)*, 9–6, 9–7, 9–8
UCB$V_ONLINE • *Device Support (A)*, 9–8, 11–2, 11–3, 16–13; *Device Support (B)*, 1–36
UCB$V_POWER • *Device Support (A)*, 8–5, 10–5, 11–1, 17–26, 18–13
UCB$V_TEMPLATE • *Device Support (B)*, 4–6
UCB$V_TIM • *Device Support (A)*, 8–7, 10–1, 10–4; *Device Support (B)*, 2–43, 3–30, 3–104
UCB$V_TIMOUT • *Device Support (A)*, 10–4; *Device Support (B)*, 3–71, 3–104
UCB$V_VALID • *Device Support (A)*, 9–8
UCB$W_BCNT • *Device Support (A)*, 8–2, 14–19, 14–22, 15–3, 15–4, 15–14, 16–19; *Device Support (B)*, 1–41, 1–79, 3–64, 3–66, 3–71
UCB$W_BOFF • *Device Support (A)*, 8–2, 14–19, 14–21, 14–22, 14–23, 15–3, 15–4, 15–14, 16–19; *Device Support (B)*, 1–41, 1–79, 3–64, 3–66, 3–71
UCB$W_BUFQUO
  in mailbox UCB • *Device Support (B)*, 3–61
UCB$W_DEVBUFSIZ • *Device Support (A)*, 6–3; *Device Support (B)*, 3–51
  in mailbox UCB • *Device Support (B)*, 3–61
UCB$W_DEVSTS • *Device Support (A)*, 10–3
UCB$W_EC1 • *Device Support (B)*, 3–67
UCB$W_EC2 • *Device Support (B)*, 3–67
UCB$W_ERRCNT • *Device Support (A)*, 11–10; *Device Support (B)*, 3–8
UCB$W_QLEN • *Device Support (B)*, 3–28
UCB$W_REFC • *Device Support (A)*, 9–6, 9–7, 11–6, 11–7; *Device Support (B)*, 4–4
UCB$W_STS • *Device Support (A)*, 17–26
UCB$W_TT_CURSOR • *Device Support (A)*, 18–22
UCB$W_TT_DESPEE • *Device Support (A)*, 18–22
UCB$W_TT_HOLD • *Device Support (A)*, 18–22
UCB$W_TT_OUTLEN • *Device Support (A)*, 18–16, 18–21, 18–22
UCB$W_TT_PRTCTL • *Device Support (A)*, 18–14, 18–16

UCB$W_TT_SPEED • *Device Support (A),* 18–15, 18–22
UCB$W_UNIT • *Device Support (A),* 15–12
UCB (unit control block) • *System Dump Analyzer,* SDA–87; *Device Support (A),* 1–5, 3–5, 4–5; *Device Support (B),* 1–12, 1–68 to 1–91
  See also SCSI device UCB
  See also SCSI port UCB
  address • *Device Support (A),* 8–7, 11–5
  as fork block • *Device Support (A),* 8–7
  as template • *Device Support (B),* 1–78
  cloned • *Device Support (B),* 1–31, 1–78
  creation • *Device Support (A),* 11–4, 12–4, 12–21, 15–7; *Device Support (B),* 1–37, 1–68
  dual path extension • *Device Support (B),* 1–69
  error log extension • *Device Support (A),* 11–9; *Device Support (B),* 1–69, 1–80 to 1–81
  extending • *Device Support (B),* 1–69 to 1–70
  initializing • *Device Support (A),* 11–3
  local disk extension • *Device Support (A),* 11–9; *Device Support (B),* 1–69, 1–82 to 1–84, 3–9, 3–67
  local tape extension • *Device Support (A),* 11–9; *Device Support (B),* 1–69, 1–81 to 1–82, 3–9
  logical • *Device Support (B),* 1–87
  number to be created • *Device Support (A),* 6–2
  physical • *Device Support (B),* 1–86
  reference count • *Device Support (B),* 1–78
  remote terminal extension • *Device Support (B),* 1–75
  size • *Device Support (B),* 1–33, 1–69 to 1–70, 1–72, 2–22
  storing data in • *Device Support (A),* 4–5, 5–2
  synchronizing access to • *Device Support (A),* 2–4, 3–5, 3–6, 3–16
  terminal extension • *Device Support (A),* 18–2 to 18–3; *Device Support (B),* 1–69, 1–84 to 1–91
$UCBDEF macro • *Device Support (B),* 1–69
UDA50 disk adapter • *I/O User's I,* 3–3
UFO (user-file open) • *Programming Resources,* 8–8
UFO (user-file open) option • *File Def Language,* FDL–25
  See also FAB$V_UFO option
UIC (user identification code) • *Routines Intro,* A–11t, A–12t; *File Applications,* 1–10; *File Def Language,* FDL–22
  delimiting in control block fields • *RMS,* 3–7
UIC-based protection • *File Applications,* 4–21
uic data type • *Routines Intro,* A–12t
UIF option • *File Def Language,* FDL–14
ULK option • *File Def Language,* FDL–11
Unaligned bit array descriptor • *Routines Intro,* 2–38
Unaligned bit string descriptor • *Routines Intro,* 2–37
Unaligned bit string with bounds descriptor • *Routines Intro,* 2–42
UNANCHOR keyword • *VAXTPU,* 7–530 to 7–531
  with SEARCH_QUIETLY • *VAXTPU,* 7–333
Unary operator • *System Dump Analyzer,* SDA–12; *MACRO,* 3–10
  summary • *MACRO,* C–7
Unbound code
  use of local variables in • *VAXTPU,* 3–34
UNDEFINED format • *File Def Language,* FDL–35
Undefined record format option
  See FAB$C_UDF option
UNDEFINED results • *MACRO,* 7–1
UNDEFINED_KEY keyword • *VAXTPU,* 7–490
"Undefined_key" string constant parameter to GET_INFO • *VAXTPU,* 7–204
UNDEFINE_KEY built-in procedure • *VAXTPU,* 7–532 to 7–533
Underflow detection • *RTL Math,* 2–9
UNDERLINE keyword
  with MARK • *VAXTPU,* 7–261
  with SELECT • *VAXTPU,* 7–337
  with SET (PROMPT_AREA) • *VAXTPU,* 7–446
  with SET (STATUS_LINE) • *VAXTPU,* 7–476
  with SET (VIDEO) • *VAXTPU,* 7–492
"Underline_status" string constant parameter to GET_INFO • *VAXTPU,* 7–225
"Underline_video" string constant parameter to GET_INFO • *VAXTPU,* 7–225
Ungrab routine
  global selection
    fetching • *VAXTPU,* 7–202
    specifying • *VAXTPU,* 7–389
  input focus
    fetching • *VAXTPU,* 7–202
    specifying • *VAXTPU,* 7–402
UNIBUS
  accomplishing a DMA transfer on • *Device Support (A),* 14–15 to 14–26
  address size • *Device Support (A),* 14–6
  example of driver designed for • *Device Support (A),* C–1 to C–29, D–1 to D–26
  example of read operation • *Device Support (A),* 14–12 to 14–13, 14–14
  example of write operation • *Device Support (A),* 14–12, 14–15
  I/O address space • *Device Support (A),* 19–1, 19–4, 19–7
  I/O space • *Device Support (A),* 14–4
  power failure • *Device Support (A),* 19–7

UNIBUS adapter • *Device Support (A)*, 1–11, 1–13
  error interrupt from • *Device Support (A)*, 13–22, 19–7
  functions • *Device Support (A)*, 14–1 to 14–15
  interrupt service routine • *Device Support (A)*, 14–29
  nexus value of • *Device Support (A)*, 12–5
  obtaining resources of • *Device Support (A)*, 14–16
  prefetch function • *Device Support (A)*, 14–12, 14–13
  registers • *Device Support (A)*, 14–15
  scatter-gather map • *Device Support (A)*, 14–4 to 14–7
  synchronizing access to • *Device Support (A)*, 14–2
Uniprocessing device driver
  converting to multiprocessing device driver • *Device Support (A)*, E–8 to E–20
  incompatibility with multiprocessing device driver • *Device Support (A)*, 12–13, E–3
Uniprocessing environment
  contrasted with multiprocessing environment • *Device Support (A)*, 3–11, E–1
Uniprocessing synchronization image • *Device Support (A)*, 13–28
  loading • *Device Support (A)*, E–2
Unit control block
  See SCSI device UCB
  See SCSI port UCB
  See UCB
Unit delivery routine • *Device Support (B)*, 1–2
  address • *Device Support (A)*, 6–2, 12–21; *Device Support (B)*, 1–34, 2–22, 4–21
  context • *Device Support (A)*, 12–21; *Device Support (B)*, 4–21
  entry point • *Device Support (B)*, 4–21
  exit method • *Device Support (B)*, 4–21
  functions • *Device Support (A)*, 12–21; *Device Support (B)*, 4–21
  input • *Device Support (B)*, 4–21
  output • *Device Support (A)*, 12–21
  register usage • *Device Support (B)*, 4–21
  synchronization requirements • *Device Support (B)*, 4–21
Unit initialization routine • *Device Support (A)*, 1–3, 11–1 to 11–6, 12–4
  address • *Device Support (A)*, 4–6, 6–3, 6–4, 11–1, 14–30; *Device Support (B)*, 1–26, 1–30, 2–26, 4–22
  allocating controller data channel in • *Device Support (A)*, 8–4, 10–2
Unit initialization routine (Cont.)
  allocating permanent buffered data path in • *Device Support (A)*, 14–18
  allocating permanent map registers in • *Device Support (A)*, 14–20 to 14–21
  context • *Device Support (A)*, 11–1, 11–3; *Device Support (B)*, 4–22
  entry point • *Device Support (B)*, 4–22
  exit method • *Device Support (B)*, 4–23
  for connect to interrupt facility • *Device Support (A)*, 19–10, 19–15
  for generic VAXBI device • *Device Support (A)*, 16–12, 16–22
  forking in • *Device Support (A)*, 3–24, 11–6
  for MASSBUS device • *Device Support (A)*, 11–5, 15–12 to 15–13; *Device Support (B)*, 1–26
  for terminal port driver • *Device Support (A)*, 18–9, 18–12
  functions • *Device Support (A)*, 11–3; *Device Support (B)*, 4–23
  input • *Device Support (A)*, 11–3; *Device Support (B)*, 4–23
  of CONINTERR.EXE • *Device Support (A)*, 19–15
  of terminal port driver • *Device Support (B)*, 2–8
  of third-party SCSI class driver • *Device Support (A)*, 17–26 to 17–27
  register usage • *Device Support (B)*, 4–22
  synchronization requirements • *Device Support (A)*, E–11 to E–12; *Device Support (B)*, 4–22
Unit testing • *Modular Procedures*, 4–1
  black box • *Modular Procedures*, 4–2
  white box • *Modular Procedures*, 4–3
UNIVERSAL option
  See Linker Utility
Universal symbol • *Programming Resources*, 5–5; *Linker*, 1–5, 2–2, 2–8; *Patch*, PAT–8, PAT–9
  See Symbol
  declaring • *Patch*, PAT–8
  designation of • *Linker*, 1–9, 2–8, 3–12
  in shareable image creation • *Linker*, 1–11, 4–10
  reason for • *Linker*, 2–8
  referencing in a shareable image • *Patch*, PAT–8, PAT–9
  resolving • *Programming Resources*, 5–5
Unload function
  disk • *I/O User's I*, 3–32
  magnetic tape • *I/O User's I*, 6–22
UNLOCK macro • *Device Support (A)*, 3–10, E–4; *Device Support (B)*, 2–96, 3–114, 3–116
UNLOCK_SYSTEM_PAGES macro • *Device Support (B)*, 2–97

UNMANAGE_WIDGET built-in procedure • *VAXTPU*, 7–534
UNMAP built-in procedure • *VAXTPU*, 7–536 to 7–537
Unmodifiable record • *VAXTPU*, 7–448
  determining if present • *VAXTPU*, 7–175, 7–186, 7–193
  preventing or allowing erasing of • *VAXTPU*, 7–375
  sensing erasable state • *VAXTPU*, 7–169
"Unmodifiable_records" string constant parameter to GET_INFO • *VAXTPU*, 7–175, 7–186, 7–193
UNPREDICTABLE results • *MACRO*, 7–1
Unsegmented key • *File Def Language*, FDL–28
Unsolicited interrupt
  See Device interrupt
Unsolicited interrupt service routine • *Device Support (A)*, 9–5, 15–16; *Device Support (B)*, 1–30
  address • *Device Support (A)*, 6–4; *Device Support (B)*, 4–24
  context • *Device Support (B)*, 4–24
  entry point • *Device Support (B)*, 4–24
  exit method • *Device Support (B)*, 4–24
  input • *Device Support (B)*, 4–24
  register usage • *Device Support (B)*, 4–24
  synchronization requirements • *Device Support (B)*, 4–24
UNSPECIFIED data type • *VAXTPU*, 2–24
Unsupported terminals • *VAXTPU*, 2–29
UNSUPRTCPU bugcheck • *Device Support (B)*, 2–10
$UNWIND • *RTL Library*, 4–14, 4–21, 4–22 to 4–23, 4–29
Unwind condition handler • *Programming Resources*, 9–18
UP command • *File Applications*, 10–12; *Analyze/RMS_File*, ARMS–35
UPDATE attribute • *File Def Language*, FDL–3, FDL–37
UPDATE built-in procedure • *VAXTPU*, 6–9, 7–538 to 7–539
  compared with REFRESH • *VAXTPU*, 7–538
UPDATE command • *Patch*, PAT–2, PAT–6, PAT–30, PAT–89
Update file • *SUMSLP*, SUM–1
Update-if option • *File Applications*, 8–4
  See also RAB$V_UIF option
Update operation • *File Applications*, 3–9
/UPDATE qualifier • *Patch*, PAT–33 to PAT–35; *SUMSLP*, SUM–20
UPDATE secondary attribute • *File Applications*, 7–3, 7–4
Update service • *File Applications*, 8–1, 8–4; *RMS*, RMS–99, RMS–100
Update service (Cont.)
  comparing with Put service for stream format files • *RMS*, RMS–100
  condition values • *RMS*, RMS–101
  control block input fields • *RMS*, RMS–100
  control block output fields • *RMS*, RMS–101
  high-level language equivalents • *File Applications*, 8–1
  invoking • *RMS*, 5–11
  program example • *RMS*, 4–20
  requirements for using • *RMS*, RMS–100
  run-time options • *File Applications*, 9–19 to 9–20
  using with indexed files • *RMS*, RMS–100
Update sharing option
  See FAB$V_UPD option
"Update" string constant parameter to GET_INFO • *VAXTPU*, 7–208
UPDATE_IF attribute • *File Def Language*, FDL–14
UPDATE_IF secondary attribute • *File Applications*, 8–8
Updating windows • *VAXTPU*, 2–29
UPD option • *File Def Language*, FDL–3, FDL–37
UPI option • *File Def Language*, FDL–37
/UP qualifier • *Debugger*, CD–99, CD–111, CD–119
Upward compatibility • *Modular Procedures*, 6–1, A–7
User-action routine • *Modular Procedures*, 2–7
  interface • *Modular Procedures*, 3–11
  optional • *Modular Procedures*, 3–11
  passing • *Modular Procedures*, 3–11
User buffer
  address • *File Applications*, 9–17
  size • *File Applications*, 9–17
User classification • *File Def Language*, FDL–23
User context field
  See RAB$L_CTX field
User default library
  object module • *Linker*, 6–14
  shareable image • *Linker*, 6–14
User-defined condition code
  signaling • *Programming Resources*, 9–10
User-defined local label • *MACRO*, 3–7
  range • *MACRO*, 3–7
User-defined logical name tables • *System Services Intro*, 6–6
User-defined patch area
  accessing with SET PATCH_AREA • *Patch*, PAT–80
  creating and accessing • *Patch*, PAT–19
  default size • *Patch*, PAT–81
  resetting • *Patch*, PAT–19, PAT–43
  terminating use of • *Patch*, PAT–19
  when to use • *Patch*, PAT–19

User-defined symbol • *Patch*, PAT–5; *MACRO*, 3–5, 3–6
User-entered reply
  as used in example for selecting key path • *RMS*, 4–12
User-file open
  See UFO
User identification code
  See UIC
User identification code field
  See XAB$L_UIC field
User interface CSR space
  enabling interrupts from • *Device Support (A)*, 16–16
User library
  creating • *Linker*, 1–5
/USERLIBRARY qualifier • *Linker*, 2–4, LINK–21
User-mode (PSL$C_USER) constant
  for FAB$V_CHAN_MODE • *RMS*, 5–5
User number • *File Def Language*, FDL–22
User-open routine • *Programming Resources*, 8–58
User privilege • *System Services Intro*, 2–2
User procedure • *RTL Intro*, 3–1
User process interlock option
  See FAB$V_UPI option
User prompt string
  program example • *RMS*, 4–16
/USER qualifier • *Debugger*, CD–15, CD–18, CD–31, CD–212, CD–257; *System Dump Analyzer*, SDA–157
User record buffer address field
  See RAB$L_UBF field
User record buffer size field
  See RAB$W_USZ field
User stack
  displaying contents • *System Dump Analyzer*, SDA–157
User stack pointer • *System Dump Analyzer*, SDA–14
User window
  in EVE editor • *VAXTPU*, 4–16
User-written system service • *System Services Intro*, A–1
User-written VAXTPU routines
  See VAXTPU routines
user_arg data type • *Routines Intro*, A–13t
USER_FILE_OPEN attribute • *File Def Language*, FDL–25
USER_FILE_OPEN secondary attribute • *File Applications*, 7–4
USER_INTERLOCK • *File Applications*, 7–4, 7–7; *File Def Language*, FDL–37
/USER_VALUE qualifier
  in message definition • *Message*, MSG–22
/USE_CLAUSE qualifier • *Debugger*, CD–251
Using entry and display modes • *Patch*, PAT–14
Using patch area • *Patch*, PAT–17
Using procedure libraries • *Modular Procedures*, 5–11
Using symbols • *Patch*, PAT–7
Using the Patch Utility • *Patch*, PAT–1
USP symbol • *System Dump Analyzer*, SDA–14
Utility
  See also entries for each utility
  invoking from a program • *Programming Resources*, 1–24
Utility routine
  See CLI routine
Utility routines • *Programming Resources*, 1–34; *Modular Procedures*, 1–10
  See ACL Editor routine
  See CONV routine
  See DCX routines
  See EDT routines
  See FDL routines
  See LBR routines
  See PSM routines
  See SOR routines
  See VAXTPU routines
  defined • *Utility Routines*, 1–1
  forming the VAXTPU callable interface • *VAXTPU*, 4–1, 7–41
  See SMB routines • *Utility Routines*, SMB–1

V

VADD (Vector Floating Add) instruction • *MACRO*, 10–70
VADDL (Vector Integer Add) instruction • *MACRO*, 10–57
VAER (Vector Arithmetic Exception Register) • *MACRO*, 10–6
VALIDATE QUEUE command • *System Dump Analyzer*, SDA–164
Validity rules • *File Def Language*, FDL–39, FDL–40
Value • *VAXTPU*, 4–10, 7–494
  See also Built-in value type
  how to define • *Command Def*, CDU–6 to CDU–8
  symbol for last displayed value • *Delta/XDelta*, DELTA–9
VALUE clause
  for defining parameters, qualifiers, keywords • *Command Def*, CDU–6

VALUE clause (Cont.)
for PARAMETER clause • *Command Def,* CDU–24, CDU–32
for QUALIFIER clause • *Command Def,* CDU–25, CDU–34
/VALUE qualifier • *Debugger,* 8–6, CD–47
Variable
as override type • *Debugger,* 4–28
buffer • *VAXTPU,* 2–4
depositing into • *Debugger,* 4–3, 4–15
with DECwindows • *Debugger,* 1–25
examining • *Debugger,* 4–2, 4–15
with DECwindows • *Debugger,* 1–25
global • *VAXTPU,* 3–4
global section • *Debugger,* 10–17
initialized • *Debugger,* 4–1
initializing • *VAXTPU,* 2–24
local • *VAXTPU,* 3–4, 3–20, 3–34
nonstatic • *Debugger,* 3–19, 4–1
with DECwindows • *Debugger,* 1–25
optimized code • *Debugger,* 9–1
register • *Debugger,* 3–19, 4–1
with DECwindows • *Debugger,* 1–25
selecting from DECwindows window • *Debugger,* 1–23
stack local • *Debugger,* 3–19, 4–1
with DECwindows • *Debugger,* 1–25
static • *Debugger,* 3–19
uninitialized • *Debugger,* 3–23
watchpoint • *Debugger,* 3–17, 10–17
with DECwindows • *Debugger,* 1–25
Variable bit base address access type • *MACRO,* 8–17
Variable buffer descriptor • *Routines Intro,* 2–25
VARIABLE declaration • *VAXTPU,* 3–36
VARIABLE format • *File Def Language,* FDL–35
Variable-length bit field
bytes referenced • *MACRO,* 8–7
data type • *MACRO,* 8–6
Variable-length bit field instructions • *MACRO,* 9–36
Variable-length bit field routine • *RTL Library,* 2–11
Variable-length format option
See FAB$C_VAR option
Variable-length record • *File Def Language,* FDL–35
guidelines for specifying • *RMS,* 5–21
with D format • *File Applications,* 2–9
with V format • *File Applications,* 2–9
Variable name
address expression • *Debugger,* 4–8
with DECwindows • *Debugger,* 1–23
DEPOSIT command • *Debugger,* 4–3
EXAMINE command • *Debugger,* 4–2
Variable name (Cont.)
language expression • *Debugger,* 4–6
selecting from DECwindows window • *Debugger,* 1–23
SET WATCH command • *Debugger,* 3–17
Variables
recommended naming conventions • *VAXTPU,* 4–31
VARIABLES keyword
with EXPAND_NAME • *VAXTPU,* 7–135
Variable with fixed-length control field
See VFC
Varying character string data type • *Routines Intro,* 2–21
Varying length string • *RTL String Manipulation,* 2–1, 2–2, 2–3, STR–9, STR–24, STR–68
Varying string array descriptor • *Routines Intro,* 2–35
Varying string descriptor • *Routines Intro,* 2–34
varying_arg data type • *Routines Intro,* A–13t
VAX–11/725
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–6
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
VAX–11/730
bootstrap procedure for XDELTA • *Delta/XDelta,* DELTA–6
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
VAX–11/750
booting with XDELTA from • *Delta/XDelta,* DELTA–5
bootstrap procedure for XDELTA with TU58 console • *Delta/XDelta,* DELTA–6
inducing a crash • *System Dump Analyzer,* SDA–31
requesting interrupt • *Delta/XDelta,* DELTA–7
VAX–11/780
booting with XDELTA from • *Delta/XDelta,* DELTA–4
inducing a crash • *System Dump Analyzer,* SDA–30
requesting interrupt • *Delta/XDelta,* DELTA–6
VAX–11/785
booting with XDELTA from • *Delta/XDelta,* DELTA–4
inducing a crash • *System Dump Analyzer,* SDA–30
requesting interrupt • *Delta/XDelta,* DELTA–6

VAX 6200
  inducing a crash • *System Dump Analyzer,* SDA–29
VAX 8200
  booting with XDELTA from • *Delta/XDelta,* DELTA–4
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–7
VAX 8230
  inducing a crash • *System Dump Analyzer,* SDA–29
VAX 8250
  booting with XDELTA from • *Delta/XDelta,* DELTA–4
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–7
VAX 8300
  booting with XDELTA from • *Delta/XDelta,* DELTA–4
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–7
VAX 8350
  booting with XDELTA from • *Delta/XDelta,* DELTA–4
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–7
VAX 8530
  booting with XDELTA from • *Delta/XDelta,* DELTA–2
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–6
VAX 8550
  booting with XDELTA from • *Delta/XDelta,* DELTA–2
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–6
VAX 8600
  booting with XDELTA from • *Delta/XDelta,* DELTA–3
  inducing a crash • *System Dump Analyzer,* SDA–30
  requesting interrupt • *Delta/XDelta,* DELTA–6
VAX 8650
  booting with XDELTA from • *Delta/XDelta,* DELTA–3
  inducing a crash • *System Dump Analyzer,* SDA–30
VAX 8650 (Cont.)
  requesting interrupt • *Delta/XDelta,* DELTA–6
VAX 8700
  booting with XDELTA from • *Delta/XDelta,* DELTA–2
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–6
VAX 8800
  booting with XDELTA from • *Delta/XDelta,* DELTA–2
  inducing a crash • *System Dump Analyzer,* SDA–29
  requesting interrupt • *Delta/XDelta,* DELTA–6
VAX 8830
  inducing a crash • *System Dump Analyzer,* SDA–29
VAX 8850
  inducing a crash • *System Dump Analyzer,* SDA–29
VAX 9000
  bus architecture • *Device Support (A),* 1–16
  hardware • *Device Support (A),* 1–16
  I/O address space • *Device Support (A),* 16–5
VAX Ada • *Programming Resources,* 1–5
VAX APL • *Programming Resources,* 1–6
VAX BASIC • *Programming Resources,* 1–6
  USEROPEN routine • *File Applications,* 5–10, 9–5
VAXBI
  displaying bus assignments • *Device Support (A),* 12–10
  displaying mapped addresses • *Device Support (A),* 12–9
VAXBI bus • *Device Support (A),* 1–13
  address • *Device Support (A),* 16–2 to 16–5
  arbitration mode of • *Device Support (A),* 16–25
  errors • *Device Support (A),* 16–26
  I/O address space • *Device Support (A),* 16–2, 16–17, 19–1
  master of • *Device Support (A),* 16–10
  memory space • *Device Support (A),* 16–2
VAXBI node
  See also Generic VAXBI device, Node ID
  definition • *Device Support (A),* 16–1
  determining self-test status of • *Device Support (A),* 16–13
  enabling BIIC options on • *Device Support (A),* 16–16
  enabling error interrupts from • *Device Support (A),* 16–16
  mapping window space of • *Device Support (A),* 16–16 to 16–18; *Device Support (B),* 3–107

VAXBI node (Cont.)
- setting interrupt destination of • *Device Support (A)*, 16–15
- setting interrupt vector for • *Device Support (A)*, 16–15

VAXBI-to-UNIBUS adapter
- See DWBUA
- See DWMUA

VAX BLISS
- using JSB entry point • *RTL Intro*, 2–2

VAX BLISS-32 • *Programming Resources*, 1–6; *System Services Intro*, 2–4; *File Def Language*, FDL–41
- example in • *RTL Parallel Processing*, 6–4

VAX C • *Programming Resources*, 1–7
- example in • *RTL Parallel Processing*, 6–14

VAXcluster • *File Applications*, 3–28
- base address of loadable code • *System Dump Analyzer*, SDA–13
- displaying SDA information • *System Dump Analyzer*, SDA–82
- locking considerations • *File Applications*, 3–29

VAX COBOL • *Programming Resources*, 1–7

VAX common language environment • *Programming Resources*, 1–5

VAX compilers
- See Compilers

VAX condition • *Routines Intro*, 2–44

VAX condition codes • *MACRO*, 10–17

VAX Condition Handling Standard • *Routines Intro*, 2–44
- exception • *Routines Intro*, 2–44

VAX data type • *Routines Intro*, 1–8

VAX DEC/CMS (Code Management System) • *Modular Procedures*, 1–12

VAX DEC/MMS (Module Management System) • *Modular Procedures*, 1–12

VAX DEC/Test Manager • *Modular Procedures*, 1–12

VAX DIBOL • *Programming Resources*, 1–8

VAX FORTRAN • *Programming Resources*, 1–8; *File Def Language*, FDL–33
- /BLAS qualifier • *RTL Math*, 2–1
- example in • *RTL Parallel Processing*, 6–9

VAX FORTRAN-HPO compiler • *RTL Math*, 2–1, 2–10

VAX instruction set
- accessing through Run-Time Library • *RTL Library*, 2–9

VAX language
- use with control blocks • *RMS*, 2–1

VAX language extension • *Routines Intro*, 2–6

VAX language implementation table
- See Implementation table

VAX Language-Sensitive Editor • *Debugger*, CD–78

VAX LISP • *Programming Resources*, 1–8

VAX MACRO • *Programming Resources*, 1–9; *System Services Intro*, 2–1, 2–4, 2–5; *File Applications*, 3–12, 3–15, 3–27, 4–2
- See Addressing mode
- See also Macro
- See Directive
- and VMS RMS • *File Applications*, 9–5
- using JSB entry point • *RTL Intro*, 2–2

VAX MACRO instruction
- as used in device driver • *Device Support (A)*, 5–1 to 5–5
- entering • *Patch*, PAT–21
- formatting memory with SDA • *System Dump Analyzer*, SDA–51
- INSERT command • *Patch*, PAT–68
- with same opcode • *Patch*, PAT–21

VAX object language • *Linker*, 7–1 to 7–37

VAX Pascal • *Programming Resources*, 1–9

VAX PL/I • *Programming Resources*, 1–10

VAX Procedure and Condition Handling Standard
- for calling services • *RMS*, 3–3

VAX procedure calling conventions • *System Services Intro*, 2–1

VAX Procedure Calling Standard • *Routines Intro*, 2–1
- address • *Routines Intro*, 2–3
- argument list • *Routines Intro*, 2–3
- argument list format • *Routines Intro*, 2–4
- calling sequence • *Routines Intro*, 2–4
  - argument list • *Routines Intro*, 2–4
- condition value • *Routines Intro*, 2–3
  - severity code • *Routines Intro*, 2–9
- condition value format • *Routines Intro*, 2–8
- data type • *Routines Intro*, 2–15
  - atomic • *Routines Intro*, 2–15
  - COBOL intermediate temporary • *Routines Intro*, 2–20
  - miscellaneous • *Routines Intro*, 2–18
  - string • *Routines Intro*, 2–17
- descriptor • *Routines Intro*, 2–3
- descriptor formats • *Routines Intro*, 2–21
- exception condition • *Routines Intro*, 2–3
- for high-level languages • *Routines Intro*, 2–6
- function • *Routines Intro*, 2–3
- function value • *Routines Intro*, 2–7
- goals • *Routines Intro*, 2–2
- immediate value • *Routines Intro*, 2–3
- introduction • *Routines Intro*, 2–1
- language support procedures • *Routines Intro*, 2–4
- library procedures • *Routines Intro*, 2–4

VAX Procedure Calling Standard (Cont.)
- procedure • *Routines Intro*, 2–3
- reference • *Routines Intro*, 2–3
- registers • *Routines Intro*, 2–12
- stacks
  - use of • *Routines Intro*, 2–14
- subroutine • *Routines Intro*, 2–3
- VAX language extensions • *Routines Intro*, 2–6

VAX RMS Journaling
- error caused by active recovery units • *Analyze/RMS_File*, ARMS–9
- how to turn off • *Analyze/RMS_File*, ARMS–8

VAX RMS Journaling errors
- how to handle • *Analyze/RMS_File*, ARMS–8

VAX RMS Journaling recovery units
- how to turn off • *Analyze/RMS_File*, ARMS–9

VAX RPG II • *Programming Resources*, 1–10

VAX scalar
- See Scalar

VAX SCAN • *Programming Resources*, 1–11

VAX standard data type • *Routines Intro*, 1–8

VAXstation
- See Workstation

VAXstation 2000
- bootstrap procedure for XDELTA • *Delta/XDelta*, DELTA–5
- requesting interrupt • *Delta/XDelta*, DELTA–7

VAXstation 3520/3540 system
- support for SCSI devices • *Device Support (A)*, 1–18, 1–19

VAXstation II
- inducing a crash • *System Dump Analyzer*, SDA–31

VAX Text Processing Utility (VAXTPU) • *Programming Resources*, 1–4
- built-in procedures • *VAXTPU*, 1–2
- DECwindows • *VAXTPU*, 1–2
- EVE editor • *Programming Resources*, 1–5
- file support • *VAXTPU*, F–1
- journaling methods • *VAXTPU*, 1–11
- relationship with DECwindows features • *VAXTPU*, 1–2
- running from a subprocess
  - example • *VAXTPU*, A–5
- used with UIL • *VAXTPU*, 1–4

VAX Text Processing Utility Routines
- See VAXTPU routines

VAXTPU
- See VAX Text Processing Utility

VAXTPU callable interface
- See VAXTPU routines

VAXTPU routines
- callable VAXTPU • *Utility Routines*, TPU–1
  - error handling • *Utility Routines*, TPU–3
  - full interface • *Utility Routines*, TPU–2, TPU–6
  - overview • *Utility Routines*, TPU–1
  - simplified interface • *Utility Routines*, TPU–2, TPU–5
- condition handler
  - condition codes • *Utility Routines*, TPU–4
  - default • *Utility Routines*, TPU–4
  - return values • *Utility Routines*, TPU–4
  - universal symbols • *Utility Routines*, TPU–4
- examples • *Utility Routines*, TPU–5, TPU–8 to TPU–25
- introduction • *Utility Routines*, TPU–1
- parameter
  - bound procedure value • *Utility Routines*, TPU–4
- shareable image • *Utility Routines*, TPU–1, TPU–3
  - constants • *Utility Routines*, TPU–3
  - symbols • *Utility Routines*, TPU–3
- user-written
  - FILEIO • *Utility Routines*, TPU–51
  - HANDLER • *Utility Routines*, TPU–53
  - INITIALIZE • *Utility Routines*, TPU–54
  - requirements • *Utility Routines*, TPU–8
  - USER • *Utility Routines*, TPU–55

VAX vector
- See Vector

VAX Vector Instruction Emulation Facility
- See VVIEF

VBIC (Vector Bit Clear) instruction • *MACRO*, 10–64

VBIS (Vector Bit Set) instruction • *MACRO*, 10–64

VBN (virtual block number) • *Analyze/RMS_File*, ARMS–6

VCB (volume control block) • *System Dump Analyzer*, SDA–99; *Device Support (B)*, 1–74, 1–78

VCMP (Vector Floating Compare) instruction • *MACRO*, 10–72

VCMPL (Vector Integer Compare) instruction • *MACRO*, 10–59

%VCR
- See VCR

VCR (Vector Count Register) • *Debugger*, 11–4, D–3; *MACRO*, 10–3, 10–88, 10–90

VDIV (Vector Floating Divide) instruction • *MACRO*, 10–78

VEC$B_DATAPATH • *Device Support (A)*, 14–17, 14–18, 14–21, 14–25

VEC$B_NUMREG • *Device Support (A)*, 14–20

VEC$L_IDB • *Device Support (A)*, 4–6, 15–13

VEC$L_INITIAL • *Device Support (A)*, 4–6, 12–4; *Device Support (B)*, 4–8
VEC$L_ISR • *Device Support (A)*, 4–6, E–5; *Device Support (B)*, 4–13
VEC$L_RTINTD • *Device Support (A)*, 14–34, 14–35
VEC$L_UNITINIT • *Device Support (A)*, 4–6, 12–4; *Device Support (B)*, 4–22
VEC$Q_DISPATCH • *Device Support (B)*, 1–25
VEC$V_LWAE • *Device Support (A)*, 14–15, 14–21; *Device Support (B)*, 3–78
VEC$V_MAPLOCK • *Device Support (A)*, 14–20; *Device Support (B)*, 3–90
VEC$V_PATHLOCK • *Device Support (A)*, 14–17, 14–18; *Device Support (B)*, 3–87
VEC$W_MAPALT • *Device Support (A)*, 14–21, 14–23
VEC$W_MAPREG • *Device Support (A)*, 14–20, 14–22
VEC$W_NUMALT • *Device Support (A)*, 14–21
VEC (interrupt transfer vector) • *Device Support (A)*, 14–29, 14–30, 14–30 to 14–33; *Device Support (B)*, 1–9, 1–22 to 1–27
  initializing • *Device Support (A)*, 14–31
  multiple • *Device Support (B)*, 1–23
$VECEND macro • *Device Support (A)*, 18–6; *Device Support (B)*, 2–99
  example • *Device Support (B)*, 2–100
$VECINI macro • *Device Support (A)*, 18–6; *Device Support (B)*, 2–98, 2–100
  example • *Device Support (B)*, 2–100
$VEC macro • *Device Support (A)*, 18–6; *Device Support (B)*, 2–98
  example • *Device Support (B)*, 2–100
VECTAB
  See Adapter dispatch table
Vector
  applying Givens plane rotation • *RTL Math*, MTH–173
  copying • *RTL Math*, MTH–160
  fixed space • *Device Support (A)*, 12–14
  floating space • *Device Support (A)*, 12–14
  generating the elements for a Givens plane rotation • *RTL Math*, MTH–178
  multiplying • *RTL Math*, MTH–155
  obtaining the Euclidean norm of • *RTL Math*, MTH–170
  obtaining the index of • *RTL Math*, MTH–149
  obtaining the inner product of • *RTL Math*, MTH–165
  obtaining the sum of the absolute values of • *RTL Math*, MTH–152
  processor synchronization • *Routines Intro*, 2–13
  register usage • *Routines Intro*, 2–12
Vector (Cont.)
  scaling • *RTL Math*, MTH–183
  swapping • *RTL Math*, MTH–187
Vector address translation • *MACRO*, 10–47
Vector code
  assembling • *MACRO*, 6–23
Vector control word • *MACRO*, 10–9, 10–13, 10–17
  EXC (Exception Enable) bit • *MACRO*, 10–11, 10–12, 10–13, 10–17, 10–28, 10–58, 10–61, 10–63, 10–68, 10–71, 10–76, 10–79, 10–81, 10–83
  MI (Modify Intent) bit • *MACRO*, 10–11, 10–12, 10–18, 10–50, 10–53
  MOE (Masked Operations Enable) bit • *MACRO*, 10–11, 10–12, 10–18
  MTF (Match True/False) bit • *MACRO*, 10–11, 10–12, 10–18
  register specifier fields • *MACRO*, 10–13
Vector Count Register
  See VCR
Vector exception
  delivery of • *Debugger*, 11–19, 11–20, 11–23
Vector instruction • *Debugger*, 11–8
  CANCEL BREAK/VECTOR_INSTRUCTION command • *Debugger*, 11–3, CD–18
  CANCEL TRACE/VECTOR_INSTRUCTION command • *Debugger*, 11–3, CD–31
  decoding • *MACRO*, 10–18
  delivery of vector exception • *Debugger*, 11–19, 11–20, 11–23
  depositing • *Debugger*, 11–13
  displaying • *Debugger*, 11–8
  EXAMINE/OPERANDS command • *Debugger*, 11–9
  examining • *Debugger*, 11–9
  execution • *MACRO*, 10–21
  formats • *MACRO*, 10–9
  masked operation • *Debugger*, 11–10, 11–15
  operand • *Debugger*, 11–9
  replacing • *Debugger*, 11–13
  SET BREAK/VECTOR_INSTRUCTION command • *Debugger*, 11–3, CD–134
  SET STEP VECTOR_INSTRUCTION command • *Debugger*, 11–3, CD–180
  SET TRACE/VECTOR_INSTRUCTION command • *Debugger*, 11–3, CD–191
  STEP/VECTOR_INSTRUCTION command • *Debugger*, 11–3, CD–267
Vectorization of a loop
  preventing • *RTL Math*, MTH–192, MTH–197, MTH–201, MTH–205

- Vectorized program
  - CALL/[NO]SAVE_VECTOR_STATE command • *Debugger*, 11–23, CD–11
  - controlling and monitoring execution • *Debugger*, 11–2
  - debugging • *Debugger*, 11–1
    - with DECwindows • *Debugger*, 1–30
  - delivery of vector exception • *Debugger*, 11–19, 11–20, 11–23
  - depositing into vector register • *Debugger*, 11–4, 11–7
  - depositing vector instruction • *Debugger*, 11–13
  - EXAMINE/FMASK command • *Debugger*, 11–13
  - EXAMINE/OPERANDS command • *Debugger*, 11–9, CD–87
  - EXAMINE/TMASK command • *Debugger*, 11–13
  - examining vector instruction • *Debugger*, 11–9
  - examining vector register • *Debugger*, 11–4, 11–7
  - masked operation • *Debugger*, 11–6, 11–10, 11–13, 11–14
  - obtaining information about • *Debugger*, 11–2
  - setting breakpoint • *Debugger*, 11–3
  - setting tracepoint • *Debugger*, 11–3
  - setting watchpoint • *Debugger*, 11–3
  - SET VECTOR_MODE command • *Debugger*, 11–20, CD–198
  - SHOW PROCESS/FULL command • *Debugger*, 11–2
  - SHOW VECTOR_MODE command • *Debugger*, 11–20, CD–260
  - specifying vector register • *Debugger*, 11–4
  - SYNCHRONIZE VECTOR_MODE command • *Debugger*, 11–20, CD–273
  - synchronizing scalar and vector processors • *Debugger*, 11–20
  - V0 to V15 • *Debugger*, 11–7
  - VCR • *Debugger*, 11–4
  - VLR • *Debugger*, 11–5
  - VMR • *Debugger*, 11–6, 11–10, 11–13, 11–14
  - with DECwindows • *Debugger*, 1–30
- Vectorizing FORTRAN compiler • *RTL Math*, 2–8
- Vector jump table
  - See Adapter dispatch table
- Vector Length Register
  - See VLR
- Vector Logical Functions • *MACRO*, 10–64
- Vector Mask Register
  - See VMR
- Vector memory
  - accessing page tables • *MACRO*, 10–47
  - access mode • *MACRO*, 10–20, 10–49
  - alignment • *MACRO*, 10–49
- Vector memory (Cont.)
  - HALT considerations • *MACRO*, 10–43
  - indicating intent to modify • *MACRO*, 10–12
  - instructions • *MACRO*, 10–49
  - management
    - See Memory management
  - required use of synchronization instructions • *MACRO*, 10–42
  - scalar/vector synchronization of • *MACRO*, 10–38
  - stride • *MACRO*, 10–49
- Vector Memory Activity Check Register
  - See VMAC
- Vector mode
  - SET VECTOR_MODE [NO]SYNCHRONIZED command • *Debugger*, 11–20
  - SYNCHRONIZE VECTOR_MODE command • *Debugger*, 11–20
- Vector opcode • *MACRO*, D–1
- Vector processor
  - exception handling • *Routines Intro*, 2–51
- Vector processor disabled • *MACRO*, 10–31, 10–32
- Vector Processor Status Register
  - See VPSR
- Vector register • *MACRO*, 10–1
  - See also Register
  - built-in symbol • *Debugger*, 11–4, D–3
  - composite address expression • *Debugger*, 11–17
  - depositing into • *Debugger*, 11–4, 11–7
  - display, screen mode • *Debugger*, 11–24
  - examining • *Debugger*, 11–4, 11–7
  - scope • *Debugger*, 11–1
  - V0 to V15 • *Debugger*, 11–7, D–3
  - VCR • *Debugger*, 11–4, D–3
  - VLR • *Debugger*, 11–5, D–3
  - VMR • *Debugger*, 11–6, 11–10, 11–13, 11–14, D–3
  - watchpoint • *Debugger*, 11–3
- Vector routines
  - table of entry points • *RTL Math*, B–1 to B–4
- Vector State Address Register
  - See VSAR
- vector_byte_signed data type • *Routines Intro*, A–13t
- vector_byte_unsigned data type • *Routines Intro*, A–13t
- /VECTOR_INSTRUCTION qualifier • *Debugger*, 11–3, CD–18, CD–31, CD–134, CD–191, CD–267
- vector_longword_signed data type • *Routines Intro*, A–13t
- vector_longword_unsigned data type • *Routines Intro*, A–13t
- vector_quadword_signed data type • *Routines Intro*, A–13t

vector_quadword_unsigned data type • *Routines Intro*, A–13t
vector_word_signed data type • *Routines Intro*, A–13t
vector_word_unsigned data type • *Routines Intro*, A–13t
Verb
See also DEFINE VERB statement
how to define • *Command Def*, CDU–8 to CDU–9
Verification of NCS library operations
See /LOG qualifier
Verify
SET OUTPUT VERIFY command • *Debugger*, CD–159
VERIFY command • *Patch*, PAT–90
VERSION keyword • *VAXTPU*, 7–141
with FILE_SEARCH • *VAXTPU*, 7–144
Version number • *File Def Language*, FDL–20; *VAXTPU*, 4–2
"Version" string constant parameter to GET_INFO • *VAXTPU*, 7–208
VFC (variable with fixed-length control) field • *File Applications*, 2–11, 3–9, 3–10
record • *File Def Language*, FDL–34, FDL–35
converting • *Convert*, CONV–15
format of • *File Def Language*, FDL–35
record format • *File Applications*, 1–2
VFC record format option
See FAB$C_VFC option
VGATH (Gather Memory Data into Vector Register) instruction • *MACRO*, 10–12, 10–16, 10–44
Video attribute • *Programming Resources*, 7–10, 7–16, 7–20
current • *Programming Resources*, 7–16
default • *Programming Resources*, 7–16
marker • *VAXTPU*, 2–9, 7–261
PROMPT_AREA • *VAXTPU*, 7–446
range • *VAXTPU*, 2–22
SET (VIDEO) built-in procedure • *VAXTPU*, 7–492
with STATUS_LINE • *VAXTPU*, 7–476
VIDEO keyword • *VAXTPU*, 7–492
"Video" string constant parameter to GET_INFO • *VAXTPU*, 7–187, 7–193, 7–226
_VIELD macro • *Device Support (B)*, 1–70, 2–102 to 2–103
example • *Device Support (B)*, 2–103
$VIELD macro • *Device Support (B)*, 2–102 to 2–103
VIEW command • *File Def Language*, FDL–67
Viewport • *Programming Resources*, 7–17; *RTL Screen Management*, 1–6, 2–12
changing characteristics • *RTL Screen Management*, 2–14
creating • *RTL Screen Management*, 2–13
Viewport (Cont.)
deleting • *RTL Screen Management*, 2–13
moving • *RTL Screen Management*, 2–13
pasting • *RTL Screen Management*, 2–13
scrolling • *RTL Screen Management*, 2–13
unpasting • *RTL Screen Management*, 2–13
VIRTCONS spin lock • *Device Support (A)*, 3–14
Virtual address • *MACRO*, 8–1
Virtual address operator (@) • *System Dump Analyzer*, SDA–12
Virtual address register
See MBA$L_VAR
Virtual address space • *System Services Intro*, 11–2, 11–3
adding page to • *System Services*, SYS–102, SYS–163
creating • *System Services*, SYS–102
deleting page from • *System Services*, SYS–134
increasing and decreasing • *System Services Intro*, 11–2
layout • *System Services Intro*, 11–2
mapping section of • *System Services Intro*, 11–13
specifying array • *System Services Intro*, 11–5
sufficient for system dump analysis • *System Dump Analyzer*, SDA–6
VAXTPU restriction concerning • *VAXTPU*, 5–1
Virtual block
dump • *Analyze/RMS_File*, ARMS–25
Virtual block number
See VBN
Virtual-block-position option • *File Applications*, 4–31
Virtual display • *Programming Resources*, 7–10; *RTL Screen Management*, 1–5
See also Viewport
changing rendition of • *RTL Screen Management*, 2–9
checking occlusion of • *Programming Resources*, 7–12
creating • *Programming Resources*, 7–10
creating a subprocess from • *Programming Resources*, 7–16
cursor movement • *Programming Resources*, 7–20
deleting • *Programming Resources*, 7–14
deleting text • *Programming Resources*, 7–21
drawing lines • *Programming Resources*, 7–20
erasing • *Programming Resources*, 7–14
ID • *Programming Resources*, 7–10, 7–32
inserting text • *Programming Resources*, 7–18, 7–20
list pasting order of • *Programming Resources*, 7–14

Virtual display (Cont.)
- logical cursor position • *Programming Resources,* 7–17
- modifying • *Programming Resources,* 7–15
- obtaining the pasting order • *Programming Resources,* 7–14
- outputting through • *RTL Screen Management,* 2–5
- overwriting text • *Programming Resources,* 7–18, 7–20
- pasting • *Programming Resources,* 7–11
- physical cursor position • *Programming Resources,* 7–18
- popping • *Programming Resources,* 7–15
- reading data from • *Programming Resources,* 7–23
- reading from • *RTL Screen Management,* 2–12
- rearranging • *Programming Resources,* 7–13
- saving • *RTL Screen Management,* 2–15
- scrolling • *Programming Resources,* 7–20
- sharing • *Programming Resources,* 7–32
- specifying double-width characters • *Programming Resources,* 7–20
- specifying video attributes • *Programming Resources,* 7–10
- viewport • *Programming Resources,* 7–17
- writing double-width characters • *Programming Resources,* 7–19
- writing text to • *Programming Resources,* 7–17

Virtual I/O • *System Services Intro,* 7–8
- canceling requests for • *System Services,* SYS–39

Virtual I/O function • *Device Support (B),* 1–40, 1–41
- translation to logical function from • *Device Support (A),* 2–3

Virtual keyboard • *RTL Screen Management,* 1–7
- definition of • *RTL Screen Management,* 3–1
- inputting through • *RTL Screen Management,* 3–1
- obtaining data from • *RTL Screen Management,* 3–1
- reading data from • *Programming Resources,* 7–23, 7–24

Virtual keyboard characteristics
- setting and retrieving • *RTL Screen Management,* 3–1

Virtual memory address
- See Memory address

Virtual memory allocation
- See Memory allocation

Virtual memory zone
- creating • *RTL Parallel Processing,* 3–4
- deleting • *RTL Parallel Processing,* 3–4

VIRTUAL option • *File Applications,* 4–31

VIRTUALPAGECNT parameter • *System Dump Analyzer,* SDA–6

Visibility
- fetching display value of record or window • *VAXTPU,* 7–186, 7–222
- of record
  - using display value to determine • *VAXTPU,* 7–370
- setting record • *VAXTPU,* 7–448

Visible process • *Debugger,* 10–2, 10–3, 10–8
- field and buttons in main window
  - with DECwindows • *Debugger,* 1–8

/VISIBLE qualifier • *Debugger,* CD–162, CD–183, CD–236

"Visible" string constant parameter to GET_INFO • *VAXTPU,* 7–226

"Visible_bottom" string constant parameter to GET_INFO • *VAXTPU,* 7–226

"Visible_length" string constant parameter to GET_INFO • *VAXTPU,* 7–202, 7–226

%VISIBLE_PROCESS • *Debugger,* 10–12

%VISIBLE_TASK • *Debugger,* D–10

"Visible_top" string constant parameter to GET_INFO • *VAXTPU,* 7–226

"Vk100" string constant parameter to GET_INFO • *VAXTPU,* 7–202

VLD (Load Memory Data into Vector Register) instruction • *MACRO,* 10–12, 10–16, 10–44, 10–50

%VLR
- See VLR

VLR (Vector Length Register) • *Debugger,* 11–4, 11–5, D–3; *MACRO,* 10–2, 10–88, 10–90

VMAC (Vector Memory Activity Check) Register • *MACRO,* 10–7, 10–20, 10–40, 10–42, 10–44, 10–48

VMERGE (Vector Merge) instruction • *MACRO,* 10–84

%VMR
- See VMR

VMR (Vector Mask Register) • *Debugger,* 11–4, 11–6, 11–10, 11–13, 11–14, D–3; *MACRO,* 10–3, 10–24, 10–88, 10–90

VMS data type • *Routines Intro,* 1–7, A–1; *System Services Intro,* 1–6

VMS Debugger
- See Debugger

VMS executive image
- global symbols • *System Dump Analyzer,* SDA–59

VMS Linker
- See Linker Utility

VMS operating system • *File Def Language,* FDL–38

VMS print symbiont
- See Symbiont

Index

VMS RMS (Record Management Services) • *Programming Resources*, 1–35 to 1–38; *Modular Procedures*, 1–11; *System Services Intro*, 7–1; *File Applications*, 1–10; *File Def Language*, FDL–42

- allocating buffers • *File Applications*, 3–12, 3–14
- Analyze/RMS_File Utility • *Programming Resources*, 1–38
- applicable macro programming rules • *RMS*, 3–6
- argument delimiters • *RMS*, 3–10
- block I/O processing services • *RMS*, 3–5
- bucket splits • *File Applications*, 3–23
- calculating extension size • *File Applications*, 3–10
- calculating file extension size • *File Applications*, 3–5
- calling sequence • *RMS*, 2–4
- calling services • *RMS*, 1–1
- connect-time options • *File Applications*, 4–2
- control block • *File Applications*, 1–11, 4–15; *File Def Language*, FDL–2; *RMS*, 1–2
  - FAB • *Programming Resources*, 1–36
  - NAM • *Programming Resources*, 1–36
  - XAB • *Programming Resources*, 1–36
- Convert/Reclaim Utility • *Programming Resources*, 1–39
- Convert Utility • *Programming Resources*, 1–39
- Create/FDL Utility • *Programming Resources*, 1–39
- creation-time options • *File Applications*, 4–2, 4–17; *File Def Language*, FDL–41
- data structures • *File Applications*, 1–11
- data structures shown by SDA • *System Dump Analyzer*, SDA–76
- default • *Convert*, CONV–19; *File Def Language*, FDL–19
- deferred-write operation • *File Applications*, 3–15, 3–27
- device support • *Programming Resources*, 1–36
- displaying data structures • *System Dump Analyzer*, SDA–127, SDA–147
- Edit/FDL Utility • *Programming Resources*, 1–39
- error
  - recommended method for signaling • *RMS*, 2–6
- file organizations • *RMS*, 1–1
- global symbols • *System Dump Analyzer*, SDA–60, SDA–61
- how to use • *RMS*, 2–1
- Image activation • *File Applications*, 5–5
- in indexed files • *File Applications*, 3–15
- macro capabilities listed • *RMS*, 4–1
- MACRO parameter • *File Applications*, 3–12
- macros • *Programming Resources*, 1–37

VMS RMS (Record Management Services) (Cont.)

- opening file for mapping • *System Services Intro*, 11–9
- option
  - selection • *File Applications*, 9–1
- overflow into P0 • *File Applications*, 7–17
- passing arguments to • *RMS*, 1–2
- placing file information in prolog • *File Applications*, 3–15
- program interface description • *RMS*, 2–1
- Put service • *Convert*, CONV–11
- record access modes • *RMS*, 1–1
- record formats • *RMS*, 1–1
- role in reclaiming buckets • *Convert*, CONV–4
- security features • *RMS*, 1–1
- service
  - allowable program execution modes • *RMS*, 2–7
  - calling example • *RMS*, 3–11
  - naming conventions • *RMS*, 3–3
  - optional arguments to • *RMS*, 3–11
  - restrictions to calling • *RMS*, 2–7
- supporting file operations • *RMS*, 1–2
- supporting record operations • *RMS*, 1–2
- use of DEC Multinational Character Set • *RMS*, 2–7
- use of multiblocks • *File Applications*, 3–11
- use of reserved event flags • *RMS*, 2–7
- using with languages • *File Applications*, 1–10
- utilities
  - ANALYZE/RMS_FILE • *File Applications*, 1–12
  - CONVERT • *File Applications*, 1–14
  - CONVERT/RECLAIM • *File Applications*, 1–14
  - CREATE/FDL • *File Applications*, 1–14
  - EDIT/FDL • *File Applications*, 1–14
- with Prolog 3 files • *File Applications*, 10–30

VMS Symbolic Debugger

- See Debugger

VMS system image

- global symbols • *System Dump Analyzer*, SDA–59

VMS Usage • *Modular Procedures*, B–1; *Routines Intro*, 1–7, A–1; *System Services Intro*, 1–6; *RTL Intro*, 2–6

- description of • *Routines Intro*, A–1

VMS Usage entry • *Routines Intro*, 1–7

VMS Usage implementation table

- See Implementation table

VMUL (Vector Floating Multiply) instruction • *MACRO*, 10–80

VMULL (Vector Integer Multiply) instruction • *MACRO*, 10–61

Voice characteristics • *RTL DECtalk*, 1–2

- Voice characteristics (Cont.)
  - comma pause • *RTL DECtalk*, 1–2, DTK–31
  - period pause • *RTL DECtalk*, 1–2, DTK–31
  - speech rate • *RTL DECtalk*, 1–2, DTK–31
- Voice identifier
  - See DECtalk device
- Volume • *File Applications*, 1–4; *Device Support (B)*, 1–78
  - dismounting • *System Services*, SYS–143
  - getting information about
    - asynchronously • *System Services*, SYS–203
    - synchronously • *System Services*, SYS–221
  - mounting • *System Services Intro*, 7–24; *System Services*, SYS–350
  - multidisk • *File Applications*, 3–23
  - positioning • *File Applications*, 3–23
- VOLUME attribute • *File Def Language*, FDL–8
- Volume control block
  - See VCB
- Volume-number option • *File Applications*, 4–32
- Volume protection • *System Services Intro*, 7–4
- /VOLUME qualifier • *Patch*, PAT–36
- VOLUME secondary attribute • *File Applications*, 4–32
- Volume set • *File Applications*, 1–5
  - for improving performance • *File Applications*, 3–6
  - to minimize disk head competition • *File Applications*, 3–23
- Volume valid bit
  - See UCB$V_VALID
- Vote • *System Dump Analyzer*, SDA–82
- VPSR (Vector Processor Status Register) • *MACRO*, 10–4, 10–5, 10–6
  - AEX (Arithmetic Exception) bit • *MACRO*, 10–5, 10–31, 10–32, 10–33, 10–34
  - BSY (Busy) bit • *MACRO*, 10–4, 10–5, 10–6, 10–8, 10–20, 10–21, 10–33, 10–39, 10–47, 10–48
  - IMP (Implementation-Specific Hardware Error) bit • *MACRO*, 10–5, 10–31, 10–32, 10–33, 10–34, 10–47, 10–48
  - IVO (Illegal Vector Opcode) bit • *MACRO*, 10–5, 10–17, 10–31, 10–32, 10–33, 10–34
  - MF (Memory Fault) bit • *MACRO*, 10–4, 10–19, 10–30, 10–34
  - PMF (Pending Memory Fault) bit • *MACRO*, 10–4, 10–19, 10–30, 10–33, 10–34
  - RLD (State Reload) bit • *MACRO*, 10–4, 10–5, 10–34
  - RST (State Reset) bit • *MACRO*, 10–4, 10–5, 10–6, 10–8, 10–33, 10–41
  - STS (State Store) bit • *MACRO*, 10–4, 10–5, 10–33
- VPSR (Vector Processor Status Register) (Cont.)
  - VEN (Enable) bit • *MACRO*, 10–4, 10–5, 10–6, 10–18, 10–20, 10–31, 10–33, 10–34, 10–47, 10–48
- VSAR (Vector State Address Register) • *MACRO*, 10–7
- VSCAT (Scatter Vector Register Data into Memory) instruction • *MACRO*, 10–12, 10–16, 10–44, 10–56
- VSL (Vector Shift Logical) instruction • *MACRO*, 10–67
- VST (Store Vector Register Data into Memory) instruction • *MACRO*, 10–12, 10–16, 10–44, 10–54
- VSUB (Vector Floating Subtract) instruction • *MACRO*, 10–82
- VSUBL (Vector Integer Subtract) instruction • *MACRO*, 10–63
- VSYNC (Synchronize Vector Memory Access) instruction • *MACRO*, 10–41, 10–42, 10–44, 10–91
- "Vt100" string constant parameter to GET_INFO • *VAXTPU*, 7–202
- "Vt200" string constant parameter to GET_INFO • *VAXTPU*, 7–202
- "Vt300" string constant parameter to GET_INFO • *VAXTPU*, 7–202
- VTBIA (Vector TB Invalidate All) instruction • *MACRO*, 10–7, 10–8, 10–32, 10–34, 10–41, 10–47
- VVCVT (Vector Convert) instruction • *MACRO*, 10–75
- VVIEF (VAX Vector Instruction Emulation Facility)
  - SHOW PROCESS/FULL command • *Debugger*, 11–2
- VXOR (Vector Exclusive Or) instruction • *MACRO*, 10–64

W

- Wait for interrupt macro
  - See WFIKPCH macro, WFIRLCH macro
- $WAIT macro
  - format difference • *RMS*, 3–12
- Wait option
  - See RAB$V_WAT option
- Wait primitive operation • *RTL Parallel Processing*, 4–10
- /WAIT qualifier • *Debugger*, CD–263
- Wait service • *File Applications*, 8–5; *RMS*, RMS–102
  - and asynchronous operations • *File Applications*, 8–18

Wait service (Cont.)
  condition values • *RMS*, RMS–103
  control block input and output fields • *RMS*, RMS–102
WAIT_FOR_RECORD attribute • *File Def Language*, FDL–15
WAIT_FOR_RECORD secondary attribute • *File Applications*, 7–12
WAKE system service
  use of • *RTL Parallel Processing*, 5–5
Wakeup
  canceling request • *System Services*, SYS–44
  scheduling with RUN command • *System Services Intro*, 9–6
.WARN directive • *MACRO*, 6–99
Warning message • *Convert*, CONV–3
/WARNING qualifier
  in message definition • *Message*, MSG–23
Watchpoint
  aggregate • *Debugger*, 3–18, 11–3
  canceling • *Debugger*, CD–34
  defined • *Debugger*, 3–17
  displaying • *Debugger*, CD–261
  global section • *Debugger*, 10–17
  multiprocess program • *Debugger*, 10–17
  nonstatic (stack or register) variable • *Debugger*, 3–19
  register • *Debugger*, 3–19
  setting • *Debugger*, 3–17, CD–200
  source display at • *Debugger*, 6–7
  static variable • *Debugger*, 3–19
  vector register • *Debugger*, 11–3
  with DECwindows • *Debugger*, 1–25
WAT option • *File Def Language*, FDL–15
WBH option • *File Def Language*, FDL–15
WCB (window control block) • *System Dump Analyzer*, SDA–77; *Device Support (A)*, 4–10; *Device Support (B)*, 1–12, 1–39
WCK option • *File Def Language*, FDL–25
Weak definition • *Linker*, 2–9, 2–10
.WEAK directive • *MACRO*, 6–101
Weak reference • *Linker*, 2–9, 2–10
WFIKPCH macro • *Device Support (A)*, 4–16, 8–5, 8–6, 10–7, 15–14, E–10; *Device Support (B)*, 2–66, 2–104 to 2–105, 3–104, 4–19
WFIRLCH macro • *Device Support (A)*, 4–16, 8–5, 8–6; *Device Support (B)*, 2–104 to 2–105, 3–104, 4–19
WHEN clause
  example • *Debugger*, 3–14
  format • *Debugger*, CD–4
WHILE command • *Debugger*, 8–11, CD–277
White box testing • *Modular Procedures*, 4–3
Widget
  callback_parameters • *VAXTPU*, 7–209
  case sensitivity of name • *VAXTPU*, 7–74
  controlling mapping • *VAXTPU*, 7–418
  creating • *VAXTPU*, 7–72
  defining a class of • *VAXTPU*, 7–105
  deleting • *VAXTPU*, 7–108
  fetching callback routine for • *VAXTPU*, 7–214
  fetching children of in VAXTPU • *VAXTPU*, 7–210
  fetching class of in VAXTPU • *VAXTPU*, 7–214
  fetching name of • *VAXTPU*, 7–215
  finding out if managed in VAXTPU • *VAXTPU*, 7–214
  getting information about • *VAXTPU*, 7–216
  listing of • *VAXTPU*, 4–5
  main window • *VAXTPU*, 4–16
  managing • *VAXTPU*, 7–258
  membership in subclass
    finding out in VAXTPU • *VAXTPU*, 7–214
  menu bar
    in VAXTPU • *VAXTPU*, 4–16
  menu position of in VAXTPU • *VAXTPU*, 7–210
  parent of
    fetching in VAXTPU • *VAXTPU*, 7–215
  realizing in VAXTPU • *VAXTPU*, 7–306
  resource
    fetching class and data type of in VAXTPU • *VAXTPU*, 7–215
  scroll bar • *VAXTPU*, 7–224, 7–462
  scroll bar slider • *VAXTPU*, 7–224
  setting resource values of • *VAXTPU*, 7–494
  title bar • *VAXTPU*, 4–16
  unmanaging • *VAXTPU*, 7–534
  using callback data structure in VAXTPU • *VAXTPU*, 7–496
  widget_id • *VAXTPU*, 7–209
Widget children
  managing • *VAXTPU*, 7–258
  unmanaging • *VAXTPU*, 7–534
WIDGET data type • *VAXTPU*, 2–24 to 2–25
Widget resources
  data types of • *VAXTPU*, 4–12
  specifying • *VAXTPU*, 4–12
WIDGET_CALL_DATA parameter to SET built-in procedure • *VAXTPU*, 7–496
%WIDTH • *Debugger*, C–6
WIDTH parameter to SET built-in procedure • *VAXTPU*, 7–501
/WIDTH qualifier • *Debugger*, 7–23, CD–185; *Librarian*, LIB–45

- "Width" string constant parameter to GET_INFO • *VAXTPU*, 7–202
- Wildcard character • *Librarian*, LIB–5; *Convert*, CONV–5
  - See also File specification
  - and multiple file locations • *File Applications*, 5–8
  - program preprocessing • *File Applications*, 5–8 to 5–14
  - use of • *National Char Set*, NCS–27, NCS–28, NCS–38
  - use restriction • *National Char Set*, NCS–34, NCS–36
  - use with Remove service • *RMS*, RMS–82
  - use with Search service • *RMS*, 4–10
  - using with ANALYZE/RMS_FILE • *Analyze/RMS_File*, ARMS–10
- Wildcard characters
  - in file names • *VAXTPU*, 5–20
  - with CONV routines • *Utility Routines*, CONV–12
- Wildcard context field
  - See NAM$L_WCC field
- Wildcard substitution
  - specifying NAM$L_RSA field • *RMS*, 6–9
- Window • *File Applications*, 9–8 to 9–10
  - See also Display
  - adjusting size • *VAXTPU*, 7–19
  - attribute, DECwindows • *Debugger*, 1–10
  - attributes • *VAXTPU*, 7–78
  - automatic (AUTO), DECwindows • *Debugger*, 1–10
  - bottom
    - example of fetching • *VAXTPU*, B–16 to B–19
  - changing position • *VAXTPU*, 7–20
  - command
    - in EVE editor • *VAXTPU*, 4–16
  - creating • *VAXTPU*, 2–26
  - current • *VAXTPU*, 2–27, 7–77
  - debugger, separate (on workstation using VWS) • *Debugger*, 9–5, CD–154
  - default configuration, DECwindows • *Debugger*, 1–4
  - definition • *VAXTPU*, 2–25
  - deleting • *VAXTPU*, 6–4, 7–108
  - determining bottom of • *VAXTPU*, 7–222
  - determining boundaries and size of • *VAXTPU*, 7–222
  - determining last column of • *VAXTPU*, 7–224
  - determining leftmost column of • *VAXTPU*, 7–222
  - determining length of • *VAXTPU*, 7–223
  - determining top of • *VAXTPU*, 7–225
  - determining width of • *VAXTPU*, 7–226
  - dimensions • *VAXTPU*, 2–25
  - enlarging • *VAXTPU*, 7–19

- Window (Cont.)
  - fetching display value of • *VAXTPU*, 7–222
  - function of
    - in VAXTPU compared with DECwindows • *VAXTPU*, 4–16
  - getting information • *VAXTPU*, 2–29
  - instruction (INST), DECwindows • *Debugger*, 1–11, 1–22
  - key map list
    - example of fetching • *VAXTPU*, B–19 to B–22
  - length • *VAXTPU*, 2–26
    - example of fetching • *VAXTPU*, B–16 to B–19
  - making current • *VAXTPU*, 6–2
  - mapping • *VAXTPU*, 2–27, 6–3
  - message
    - in EVE editor • *VAXTPU*, 4–16
  - output (OUT), DECwindows • *Debugger*, 1–10
  - predefined, DECwindows • *Debugger*, 1–9
  - reducing • *VAXTPU*, 7–20
  - register (REG), DECwindows • *Debugger*, 1–11
  - removing • *VAXTPU*, 2–28
  - screen management • *VAXTPU*, 6–2 to 6–4
  - screen-mode, creating definition for • *Debugger*, 7–15, CD–207
  - screen-mode, defined • *Debugger*, 7–2
  - screen-mode, deleting definition of • *Debugger*, 7–15, CD–35
  - screen-mode, identifying • *Debugger*, 7–15, CD–262
  - screen-mode, predefined • *Debugger*, CD–262, C–7
  - screen-mode, specifying • *Debugger*, 7–14
  - screen updates • *VAXTPU*, 6–7
  - scroll bar in • *VAXTPU*, 7–224, 7–462
  - scroll bar slider in • *VAXTPU*, 7–224
  - selecting address expression from, DECwindows • *Debugger*, 1–23
  - setting display value of • *VAXTPU*, 7–370
  - size
    - with terminal display • *VAXTPU*, 6–4
    - with terminal emulator • *VAXTPU*, 6–4
  - source (SRC), DECwindows • *Debugger*, 1–9, 1–21
  - top
    - example of fetching • *VAXTPU*, B–16 to B–19
  - unmapping • *VAXTPU*, 2–28
  - unsupported terminals • *VAXTPU*, 2–29
  - updating • *VAXTPU*, 2–29
  - user
    - in EVE editor • *VAXTPU*, 4–16
  - values • *VAXTPU*, 2–27
  - width • *VAXTPU*, 2–26

Window
  width (Cont.)
    example of fetching • *VAXTPU,* B–19 to B–22
  window width • *VAXTPU,* 6–4
Window control block
  See WCB
WINDOW data type • *VAXTPU,* 2–25 to 2–29
Window size • *File Applications,* 10–29
Window space • *Device Support (A),* 16–5
  mapping • *Device Support (A),* 16–16 to 16–18
  starting address • *Device Support (A),* 16–17
WINDOW_SIZE attribute • *File Def Language,* FDL–25
"Within_range" string constant parameter to GET_INFO • *VAXTPU,* 7–187
Word count register • *Device Support (A),* 14–23
Word data type • *MACRO,* 8–2
.WORD directive • *MACRO,* 6–102
WORD mode • *Patch,* PAT–16
/WORD qualifier • *Debugger,* CD–63, CD–89
/WORD qualifier
  with ALIGN command • *Patch,* PAT–38
  with DELETE command • *Patch,* PAT–52
  with DEPOSIT command • *Patch,* PAT–55
  with EVALUATE command • *Patch,* PAT–59
  with EXAMINE command • *Patch,* PAT–62
  with REPLACE command • *Patch,* PAT–71
  with SET MODE command • *Patch,* PAT–76
  with VERIFY command • *Patch,* PAT–90
Word separators • *VAXTPU,* 7–146
Word storage directive (.WORD) • *MACRO,* 6–102
word_signed data type • *Routines Intro,* A–13t
word_unsigned data type • *Routines Intro,* A–13t
Working set • *File Applications,* 1–16
  adjusting for optimal sort performance • *Convert,* CONV–22
  adjusting limit • *System Services,* SYS–10
  adjusting size • *Programming Resources,* 10–3; *System Services Intro,* 11–6
  locking page into • *Programming Resources,* 10–3; *System Services Intro,* 11–6; *System Services,* SYS–337
  paging • *System Services Intro,* 11–6
  purging • *System Services,* SYS–370
  unlocking page from • *System Services,* SYS–528
Working set limit • *Device Support (B),* 3–35, 3–41
  insufficient • *Device Support (B),* 3–33
Working set list
  displaying • *System Dump Analyzer,* SDA–128
Working set quota
  how to determine • *Convert,* CONV–22
/WORKING_SET qualifier • *System Dump Analyzer,* SDA–128
WORKING_SET_MANAGEMENT.EXE
  global symbols • *System Dump Analyzer,* SDA–61
Work item
  deleting • *RTL Parallel Processing,* 4–18
  inserting • *RTL Parallel Processing,* 4–17
  removing • *RTL Parallel Processing,* 4–18
Work queue
  creating • *RTL Parallel Processing,* 4–16
  definition of • *RTL Parallel Processing,* 4–16
  deleting • *RTL Parallel Processing,* 4–17
  deleting work item from • *RTL Parallel Processing,* 4–18
  first in first out • *RTL Parallel Processing,* 4–16, 4–18
  inserting an item into • *RTL Parallel Processing,* 4–17
  reading • *RTL Parallel Processing,* 4–17
  removing work item from • *RTL Parallel Processing,* 4–18
Work queue processing software model • *RTL Parallel Processing,* 1–5
Work queue synchronization
  advantages and disadvantages • *RTL Parallel Processing,* 5–9
  PPL$ routines for • *RTL Parallel Processing,* 4–16 to 4–18
Workstation
  debugger commands for (when using VWS) • *Debugger,* CD–6
  debugger DECwindows interface • *Debugger,* 1–1
  debugging DECwindows application • *Debugger,* 1–36
  debugging screen-oriented program (when using VWS) • *Debugger,* 9–5
  popping window (when using VWS) • *Debugger,* CD–166
  screen size (when using VWS) • *Debugger,* 7–23, CD–185
  separate, for debugger DECwindows interface • *Debugger,* 1–36
  separate debugger window (when using VWS) • *Debugger,* 9–5, CD–154
Workstation device • *Device Support (B),* 1–76
/WORK_FILES qualifier • *Convert,* CONV–12, CONV–27
WORLD category • *File Def Language,* FDL–23
WRITE access • *File Def Language,* FDL–23
Write access type • *MACRO,* 8–17
Write attention AST function • *I/O User's I,* 7–9
Write-back section • *System Services Intro,* 11–19

Write-behind option
See RAB$V_WBH option
Write breakthrough function • *I/O User's I*, 8–36
Write check
enabling • *Device Support (B)*, 1–75
Write check option
See FAB$V_WCK option
Write end-of-file function
magnetic tape • *I/O User's I*, 6–21
message • *I/O User's I*, 7–9
Write function
FDT routine for • *Device Support (A)*, 7–9
Write protection
hardware • *I/O User's I*, 10–4
/WRITE qualifier • *VAXTPU*, 5–17
Write service • *RMS*, RMS–104, RMS–105
condition values • *RMS*, RMS–106
control block input fields • *RMS*, RMS–105
control block output fields • *RMS*, RMS–105
"Write" string constant parameter to GET_INFO • *VAXTPU*, 7–178
WRITE_BEHIND attribute • *File Def Language*, FDL–15
WRITE_CHECK attribute • *File Def Language*, FDL–25
/WRITE_CHECK qualifier • *Convert*, CONV–28
WRITE_CLIPBOARD built-in procedure • *VAXTPU*, 7–540
example of use • *VAXTPU*, B–11 to B–13
WRITE_FILE built-in procedure • *VAXTPU*, 7–543 to 7–545
WRITE_GLOBAL_SELECT built-in procedure • *VAXTPU*, 7–546
example of use • *VAXTPU*, B–31 to B–33
Writing operations • *RTL Screen Management*, 2–8

X

X4 symbol • *Delta/XDelta*, DELTA–9
X5 symbol • *Delta/XDelta*, DELTA–9
XAB$B_AID field • *File Applications*, 4–30; *File Def Language*, FDL–6; *RMS*, 8–2
XAB$B_ALN field • *File Def Language*, FDL–8; *RMS*, 8–2
options • *File Applications*, 4–31
XAB$B_AOP field • *File Def Language*, FDL–6, FDL–7; *RMS*, 8–3
options • *File Applications*, 4–30; *RMS*, 8–4
XAB$B_ATR field • *RMS*, 10–2
options • *RMS*, 10–2
XAB$B_BKZ field • *File Applications*, 3–24, 4–28, 7–19, 7–20
as output • *RMS*, 8–5
default logic • *RMS*, 8–5
determining bucket size • *RMS*, 8–5
in allocation XAB (XABALL) • *RMS*, 8–4
in file header characteristics allocation XAB (XABFHC) • *RMS*, 10–3
RMS–11 restriction • *RMS*, 8–5
size requirements for multiple index areas • *RMS*, 8–5
XAB$B_BLN field
in allocation XAB (XABALL) • *RMS*, 8–5
in date and time XAB (XABDAT) • *RMS*, 9–2
in file header characteristics XAB (XABALL) • *RMS*, 10–3
in item list XAB (XABITM) • *RMS*, 11–2
in key XAB (XABKEY) • *RMS*, 13–2
in protection XAB (XABPRO) • *RMS*, 14–4
in revision date and time XAB (XABRDT) • *RMS*, 15–2
in summary XAB (XABSUM) • *RMS*, 17–1
in terminal XAB (XABTRM) • *RMS*, 18–2
XAB$B_COD field
See also COD field
in allocation XAB (XABALL) • *RMS*, 8–5
in date and time XAB (XABDAT) • *RMS*, 9–3
in file header characteristics XAB (XABFHC) • *RMS*, 10–3
in item list XAB (XABITM) • *RMS*, 11–2
in key XAB (XABKEY) • *RMS*, 13–2
in protection XAB (XABPRO) • *RMS*, 14–4
in revision date and time XAB (XABRDT) • *RMS*, 15–2
in summary XAB (XABSUM) • *RMS*, 17–1
in terminal XAB (XABTRM) • *RMS*, 18–2
XAB$B_DAN field • *File Def Language*, FDL–27; *RMS*, 13–4
XAB$B_DBS field • *RMS*, 13–4
XAB$B_DPT field • *File Def Language*, FDL–32
XAB$B_DTP field • *RMS*, 13–5
data formats • *RMS*, 13–6
data type restrictions • *RMS*, 13–5
options • *RMS*, 13–5
use with search key • *RMS*, 7–13, 7–14
value prefixes for sorting • *RMS*, 13–5
XAB$B_FLG field • *File Def Language*, FDL–26, FDL–27, FDL–28, FDL–29; *RMS*, 13–8, B–21
option allowable combinations listed • *RMS*, 13–9
options • *RMS*, 13–8
XAB$B_HSZ field • *RMS*, 10–4
use restriction • *RMS*, 10–4

XAB$B_IAN field • *File Def Language*, FDL–28; *RMS*, 13–10
  conditional usage • *RMS*, 13–10
  indicating index level • *RMS*, 8–5
XAB$B_IBS field • *RMS*, 13–10
XAB$B_LAN field • *File Def Language*, FDL–28; *RMS*, 13–11
  indicating index level • *RMS*, 8–5
  relationship to XAB$B_AID field • *RMS*, 13–11
  requirement for compatibility with XAB$B_IAN field • *RMS*, 13–11
  use restriction • *RMS*, 13–11
XAB$B_LVL field • *RMS*, 13–12
XAB$B_MTACC field • *File Def Language*, FDL–22; *RMS*, 14–5
  default logic • *RMS*, 14–5
  valid character codes • *RMS*, 14–5
XAB$B_NOA field • *RMS*, 17–2
XAB$B_NOK field • *RMS*, 17–2
XAB$B_NSG field • *RMS*, 13–12
XAB$B_NUL field • *File Def Language*, FDL–29; *RMS*, 13–12
  use restrictions • *RMS*, 13–12
XAB$B_PROLOG field • *File Def Language*, FDL–30; *RMS*, 13–13
  default logic • *RMS*, 13–13
  service usage • *RMS*, 13–13
  use restriction • *RMS*, 13–13
XAB$B_PROT_OPT field • *RMS*, 14–7
XAB$B_REF field • *File Def Language*, FDL–26
XAB$B_RFO field • *RMS*, 10–5
  values listed • *RMS*, 10–6
XAB$B_SIZ0 field • *File Def Language*, FDL–28, FDL–30
XAB$B_SIZ0 through XAB$B_SIZ7 field • *RMS*, 13–14
  default logic • *RMS*, 13–15
  requirement for compatibility with XAB$W_POS0 through XAB$W_POS7 field • *RMS*, 13–14
  with segmented key • *RMS*, 13–14
  with simple key • *RMS*, 13–14
XAB$B_TKS field • *RMS*, 13–15
XAB$C_ALLEN value • *RMS*, 8–5
XAB$C_ALL value • *RMS*, 8–6
XAB$C_DATLEN value • *RMS*, 9–2
XAB$C_DAT value • *RMS*, 9–3
XAB$C_FHCLEN value • *RMS*, 10–3
XAB$C_FHC value • *RMS*, 10–3
XAB$C_ITMLEN value • *RMS*, 11–2
XAB$C_ITM value • *RMS*, 11–2
XAB$C_KEYLEN value • *RMS*, 13–2
XAB$C_KEY value • *RMS*, 13–2
XAB$C_PROLEN value • *RMS*, 14–4
XAB$C_PRO value • *RMS*, 14–4
XAB$C_RDTLEN value • *RMS*, 15–2
XAB$C_RDT value • *RMS*, 15–2
XAB$C_SUMLEN value • *RMS*, 17–1
XAB$C_SUM value • *RMS*, 17–2
XAB$C_TRMLEN value • *RMS*, 18–2
XAB$C_TRM value • *RMS*, 18–2
XAB$L_ACLBUF field • *RMS*, 14–2
  determining value for Create service • *RMS*, 14–2
  determining value for Open and Display service • *RMS*, 14–2
  handling ACE • *RMS*, 14–2
XAB$L_ACLCTX field • *RMS*, 14–2, 14–3
XAB$L_ACLSTS field • *RMS*, 14–3
  error-handling guidelines • *RMS*, 14–3
  use restriction • *RMS*, 14–4
XAB$L_ALQ field • *File Applications*, 4–30; *File Def Language*, FDL–6; *RMS*, 8–3
XAB$L_COLNAM field • *RMS*, 13–2
XAB$L_COLSIZ field • *RMS*, 13–3
XAB$L_COLTBL field • *RMS*, 13–3
XAB$L_DVB field • *RMS*, 13–7
XAB$L_EBK field • *RMS*, 10–3
XAB$L_HBK field • *RMS*, 10–4
  comparing with FAB$L_ALQ field • *RMS*, 10–4
XAB$L_ITEMLIST field • *RMS*, 11–2
XAB$L_ITMLST field • *RMS*, 18–2
  requirement for valid terminal driver • *RMS*, 18–1
XAB$L_KNM field • *File Def Language*, FDL–29; *RMS*, 13–11
XAB$L_LOC field • *File Applications*, 4–31; *File Def Language*, FDL–8; *RMS*, 8–6
  determining value • *RMS*, 8–6
  requirement for alignment option • *RMS*, 8–6
XAB$L_MODE field • *RMS*, 11–2
XAB$L_NXT field
  in XABALL • *RMS*, 8–6
  in XABDAT • *RMS*, 9–3
  in XABFHC • *RMS*, 10–5
  in XABKEY • *RMS*, 13–12
  in XABPRO • *RMS*, 14–5
  in XABRDT • *RMS*, 15–2
  in XABSUM • *RMS*, 17–2
  in XABTRM • *RMS*, 18–3
XAB$L_RVB field • *RMS*, 13–14
XAB$L_SBN field • *RMS*, 10–6
XAB$L_UIC field • *RMS*, 14–4, 14–8
  combining the XAB$W_GRP and XAB$W_MBM fields • *RMS*, 14–8
  order of determining value • *RMS*, 14–8
  setting XAB$W_GRP field • *RMS*, 14–4

XAB$L_UIC field (Cont.)
setting XAB$W_MBM field • *RMS*, 14–5
XAB$NXT field
in XABITM • *RMS*, 11–2
XAB$Q_BDT field • *File Def Language*, FDL–15; *RMS*, 9–2
XAB$Q_CDT field • *File Def Language*, FDL–16; *RMS*, 9–2
XAB$Q_EDT field • *File Def Language*, FDL–16; *RMS*, 9–3
XAB$Q_RDT field • *File Def Language*, FDL–16; *RMS*, 9–3, 15–2
XAB$V_BLK option • *RMS*, 10–2
XAB$V_CBT option • *RMS*, 8–4
XAB$V_CHG option • *RMS*, 13–8
use restriction • *RMS*, 13–8
XAB$V_CR option • *RMS*, 10–2
XAB$V_CTG option • *RMS*, 8–4
XAB$V_DAT_NCMPR option • *RMS*, 13–8
XAB$V_DUP option • *RMS*, 13–8
XAB$V_FTN option • *RMS*, 10–2
XAB$V_HRD option • *RMS*, 8–4
use restrictions • *RMS*, 8–4
XAB$V_IDX_NCMPR option • *RMS*, 13–8
use in defining string keys • *RMS*, 13–8
use restriction • *RMS*, 13–8
XAB$V_KEY_NCMPR option • *RMS*, 13–8
use in defining string keys • *RMS*, 13–8
use restriction • *RMS*, 13–9
XAB$V_NUL option • *RMS*, 13–9
setting for various data types • *RMS*, 13–6
use in defining string keys • *RMS*, 13–8
use restriction • *RMS*, 13–9
with XAB$B_NUL field • *RMS*, 13–9
XAB$V_ONC option • *RMS*, 8–4
XAB$V_PRN option • *RMS*, 10–2
XAB$V_PROPAGATE option • *RMS*, 14–7
XAB$W_ACLLEN field • *RMS*, 14–3
determining value • *RMS*, 14–3
limitation • *RMS*, 14–3
XAB$W_ACLSIZ field • *RMS*, 14–3
limitations imposed by MAXBUF • *RMS*, 14–3
limitations imposed by user's BYTLM quota • *RMS*, 14–3
XAB$W_DEQ field • *File Applications*, 4–31; *File Def Language*, FDL–7; *RMS*, 8–6
XAB$W_DFL field • *File Def Language*, FDL–27; *RMS*, 13–4
advantages of using • *RMS*, 13–4
comparing for primary and alternate keys • *RMS*, 13–4
determining value • *RMS*, 13–4

XAB$W_DFL field (Cont.)
use with RAB$V_LOA option • *RMS*, 7–13
XAB$W_DXQ field
in XABFHC • *RMS*, 10–3
XAB$W_FFB field • *RMS*, 10–4
XAB$W_GBC field
in XABFHC • *RMS*, 10–4
XAB$W_GRP field • *File Def Language*, FDL–23; *RMS*, 14–4
XAB$W_IFL field • *File Def Language*, FDL–28; *RMS*, 13–10
advantages of using • *RMS*, 13–11
XAB$W_ITMLST_LEN field • *RMS*, 18–2
requirement for valid terminal driver • *RMS*, 18–1
XAB$W_LRL field • *RMS*, 10–4
use restriction • *RMS*, 10–5
XAB$W_MBM field • *File Def Language*, FDL–23; *RMS*, 14–5
XAB$W_MRL field • *RMS*, 13–12
comparing primary key and alternate keys • *RMS*, 13–12
XAB$W_MRZ field
in XABFHC • *RMS*, 10–5
XAB$W_MRZ field in XABFHC
determining value • *RMS*, 10–5
XAB$W_POS0 field • *File Def Language*, FDL–29, FDL–30
XAB$W_POS0 through XAB$W_POS7 field • *RMS*, 13–12
requirement to be compatible with XAB$B_SIZ0 through XAB$B_SIZ7 field • *RMS*, 13–13
XAB$W_PRO field • *File Def Language*, FDL–23; *RMS*, 14–6
default logic • *RMS*, 14–7
organization • *RMS*, 14–6
required ordering of arguments • *RMS*, 14–6
subfield offsets • *RMS*, 14–6
user classes • *RMS*, 14–7
XAB$W_PVN field • *RMS*, 17–2
XAB$W_RFI field • *File Def Language*, FDL–8; *RMS*, 8–7
as argument to $XABALL_STORE macro • *RMS*, B–14
requirement for XAB$C_RFI • *RMS*, 8–7
specifying • *RMS*, 8–7
XAB$W_RVN field • *File Def Language*, FDL–24; *RMS*, 9–3, 15–3
XAB$W_VERLIMIT field
in XABFHC • *RMS*, 10–6
XAB$W_VOL field • *File Applications*, 4–32; *File Def Language*, FDL–8; *RMS*, 8–7
use restriction • *RMS*, 8–7

XAB$_REF field • *RMS*, 13–14
XAB (extended attribute block) • *Programming Resources*, 1–36; *File Applications*, 1–11, 4–2; *System Dump Analyzer*, SDA–77
  See also XAB block
  date and time fields • *File Applications*, 4–28
  description • *RMS*, 1–3
  key definition fields • *File Applications*, 4–29
  naming conventions for FAB • *RMS*, 1–3
  program example • *RMS*, 4–8
  protection fields • *File Applications*, 4–28
  types • *RMS*, 1–3
  types for VMS RMS file operations • *RMS*, 1–3
XABALL block • *RMS*, 1–3, 8–1
  relationship to FAB fields • *RMS*, 8–1
  summary of fields • *RMS*, 8–1
$XABALL macro • *RMS*, B–13
  argument categories • *RMS*, B–13
$XABALL_STORE macro • *RMS*, B–14
  argument categories • *RMS*, B–14
  comparing with $XABALL macro • *RMS*, B–14
  requirements • *RMS*, B–14
XAB block
  naming conventions for RAB • *RMS*, 1–4
XABDAT block • *RMS*, 9–1
  brief description • *RMS*, 1–3
  summary of fields • *RMS*, 9–1
  value selection logic • *RMS*, 9–2
$XABDAT macro • *RMS*, B–15
$XABDAT_STORE macro • *RMS*, B–16
  argument categories • *RMS*, B–16
  argument variations • *RMS*, B–16
  example of use • *RMS*, 3–9
  requirements • *RMS*, B–16
XABFHC block • *RMS*, 10–1
  brief description • *RMS*, 1–3
  summary of fields • *RMS*, 10–1
  use exception • *RMS*, 10–1
  values for shared sequential files • *RMS*, 10–1
$XABFHC macro • *RMS*, B–17
$XABFHC_STORE macro • *RMS*, B–18
  argument categories • *RMS*, B–18
  requirements • *RMS*, B–18
XABITM block • *RMS*, 11–1
  brief description • *RMS*, 1–3
  summary of fields • *RMS*, 11–1
$XABITM macro • *RMS*, B–19
XABJNL block • *RMS*, 12–1
  brief description • *RMS*, 1–3
XABKEY block • *RMS*, 13–1
  brief description • *RMS*, 1–3

XABKEY block (Cont.)
  data type options • *RMS*, 13–5
  default logic • *RMS*, 13–9
  summary of fields • *RMS*, 13–1
  XAB$W_MRL field • *RMS*, 13–12
$XABKEY macro • *RMS*, B–20, B–21
  argument categories • *RMS*, B–21
  position and size options • *RMS*, B–21
$XABKEY_STORE macro • *RMS*, B–22
  argument categories • *RMS*, B–23
  requirements • *RMS*, B–23
XABPRO block • *RMS*, 14–1
  brief description • *RMS*, 1–3
  summary of fields • *RMS*, 14–1
  XAB$B_BLN field • *RMS*, 14–4
  XAB$W_GRP field • *RMS*, 14–4
$XABPRO macro • *RMS*, B–24
  ASCII radix indicator requirement in MTACC argument • *RMS*, B–24
  describing UIC argument • *RMS*, B–25
  example of MTACC argument • *RMS*, B–24
  listing user classes • *RMS*, B–25
  XAB$W_PRO field requirements • *RMS*, B–24
$XABPRO_STORE macro • *RMS*, B–26
  argument categories • *RMS*, B–26
  argument exceptions to general rules • *RMS*, B–26
  requirements • *RMS*, B–26
XABRDT block • *RMS*, 15–1
  brief description • *RMS*, 1–3
  comparing with XABDAT • *RMS*, 15–1
  default logic • *RMS*, 15–1
  service use of XAB$Q_RDT and XAB$W_RVN fields • *RMS*, 15–1
  summary of fields • *RMS*, 15–1
  use restriction • *RMS*, 15–1
$XABRDT macro • *RMS*, B–27
$XABRDT_STORE macro • *RMS*, B–28
  argument categories • *RMS*, B–28
  requirements • *RMS*, B–28
XABRU block • *RMS*, 16–1
  brief description • *RMS*, 1–3
XABSUM block • *RMS*, 17–1
  brief description • *RMS*, 1–3
  summary of fields • *RMS*, 17–1
  use restriction • *RMS*, 17–1
$XABSUM macro • *RMS*, B–29
$XABSUM_STORE macro • *RMS*, B–30
  argument categories • *RMS*, B–30
  requirements • *RMS*, B–30
XABTRM block • *RMS*, 18–1
  brief decription • *RMS*, 1–4
  requirements to use • *RMS*, 18–1

XABTRM block (Cont.)
  summary of fields • *RMS*, 18–1
$XABTRM macro • *RMS*, B–31
$XABTRM_STORE macro • *RMS*, B–32
  argument categories • *RMS*, B–32
  requirements • *RMS*, B–32
XADRIVER.MAR • *Device Support (A)*, D–1 to D–26
;X command • *Delta/XDelta*, DELTA–40
XDELTA
  See Delta/XDelta Utility
XDELTA entry IPL • *Device Support (A)*, 3–9
XE base register • *Delta/XDelta*, DELTA–9, DELTA–38
XF base register • *Delta/XDelta*, DELTA–9, DELTA–38
XFC (Extended Function Call) instruction • *MACRO*, 9–81
XFMAXRATE parameter • *I/O User's II*, 4–22
%X format • *Analyze/RMS_File*, ARMS–25
XMI
  displaying mapped addresses • *Device Support (A)*, 12–11
XMI bus
  memory space • *Device Support (A)*, 16–5
Xn symbol • *Delta/XDelta*, DELTA–9
XORB2 (Exclusive OR Byte 2 Operand) instruction • *MACRO*, 9–32
XORB3 (Exclusive OR Byte 3 Operand) instruction • *MACRO*, 9–32
XORL2 (Exclusive OR Long 2 Operand) instruction • *MACRO*, 9–32
XORL3 (Exclusive OR Long 3 Operand) instruction • *MACRO*, 9–32
XOR operator • *VAXTPU*, 3–7
XOR operator (\) • *System Dump Analyzer*, SDA–13
XORW2 (Exclusive OR Word 2 Operand) instruction • *MACRO*, 9–32
XORW3 (Exclusive OR Word 3 Operand) instruction • *MACRO*, 9–32
XQP (extended QIO processor) • *I/O User's I*, 1–1; *System Dump Analyzer*, SDA–99; *Device Support (B)*, 1–12, 1–74
  default • *Device Support (B)*, 1–28
X resource
  fetching value of • *VAXTPU*, 7–151

Y

YES logical value • *File Def Language*, FDL–2

Z

Zero condition code (Z) • *MACRO*, 8–15
Zone • *RTL Library*, 5–6
  See Virtual memory zone
  allocation algorithm • *RTL Library*, 5–15
  attribute • *RTL Library*, 5–8
  creating • *RTL Library*, 5–6
  default • *RTL Library*, 5–12
  deleting • *RTL Library*, 5–6
  identifier • *RTL Library*, 5–12
  resetting • *RTL Library*, 5–14
  user-created • *RTL Library*, 5–6

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal[1]	————	USASSB Order Processing - WMO/E15 *or* U.S. Area Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

[1] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

VMS Programming
Master Index
AA–LA56B–TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

I would like to see more/less ______________________________

____

____

What I like best about this manual is ______________________________

____

____

What I like least about this manual is ______________________________

____

____

I found the following errors in this manual:

Page	Description
______	______________________________
______	______________________________
______	______________________________
______	______________________________
______	______________________________

Additional comments or suggestions to improve this manual:

____

____

____

____

I am using **Version** ______ of the software this manual describes.

Name/Title ______________________________ Dept. ______________

Company ______________________________ Date ______________

Mailing Address ______________________________

______________________________ Phone ______________

— 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

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01–3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-— Do Not Tear - Fold Here