TOPS-20
Monitor Calls
Quick Reference Guide
AV-P173A-TM

December, 1982

This guide provides a brief description of all of the TOPS-20 monitor calls and many of the blocks in the monitor's data base. It is intended for use by experienced MACRO-20 programmers who require a reminder of calling sequences and function codes. MACRO-20 programmers who require a more detailed description of the monitor calls should use the TOPS-20 Monitor Calls Reference Manual; those who desire a more introductory discussion on the use of monitor calls should refer to the TOPS-20 Monitor Calls User's Guide.

OPERATING SYSTEM: TOPS-20, V5.1
# TABLE OF CONTENTS

- **CONVENTIONS** ........................................ iv
- **MONITOR CALLS FUNCTIONAL ORGANIZATION** .......... 1
- **TOPS-20 MONITOR CALLS** ............................. 8
- **CONTROL CHARACTER OUTPUT CONTROL (CCOC) WORD** 155
- **COMMUNICATIONS PROTOCOLS** ......................... 156
- **DEVICE TYPES** ........................................ 157
- **DIRECTORY PROTECTION FIELDS** ....................... 157
- **FILE PROTECTION FIELDS** ............................. 157
- **FILE DESCRIPTOR BLOCK (FDB)** ....................... 158
- **FORK (PROCESS) HANDLES** ............................ 159
- **FLOATING-POINT FORMAT CONTROL** .................. 159
- **I/O IDENTIFIERS** ..................................... 160
- **JFN MODE WORD** ....................................... 161
- **JOB CAPABILITY WORD** ................................. 161
- **MAGTAPE DEVICE TYPES** ............................... 162
- **MAGTAPE DRIVE TYPES** ................................ 162
- **MAGTAPE HARDWARE DATA MODES** ...................... 162
- **MAGTAPE LABEL STATES** ............................... 163
- **MAGTAPE LABEL TYPES** ................................ 163
- **MAGTAPE RECORD SIZES** ............................... 163
- **MAGTAPE RECORDING DENSITIES** ....................... 163
- **PHYSICAL CARD PUNCH (PCDP:) STATUS BITS** ........ 163
- **PHYSICAL CARD READER (PCDR:) STATUS BITS** ....... 164
- **PHYSICAL LINE PRINTER (PLPT:) CONTROL CHARACTERS** 164
- **PHYSICAL LINE PRINTER (PLPT:) STATUS BITS** ...... 165
- **PHYSICAL MAGTAPE (MTA:) STATUS BITS** ............ 165
- **SOFTWARE DATA MODES** ................................ 165
- **SOFTWARE INTERRUPT CHANNELS** ...................... 166
- **SYSTEM PIDS** ......................................... 166
- **SYSTEM TABLES** ....................................... 167
- **TERMINAL CHARACTERISTICS** ......................... 170
- **TERMINAL INTERRUPT CODES** ......................... 172
- **TIME ZONES** .......................................... 173
- **TOPS-20 JSYS ERROR CODES** ........................... 174
- **TOPS-20 JSYS ERROR MNEMONICS** ..................... 180
- **POINTER FORMATS** .................................... 205
- **PDP-10 INSTRUCTION SET** ............................. 205
- **MACRO-20 PSEUDO-OPS** ................................ 216
**CONVENTIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bn or Bn-m</td>
<td>Bit n or bits n through m; bit positions are always decimal</td>
</tr>
<tr>
<td>(enabled priv)</td>
<td>Designates a capability that must be enabled for the specified function to be legal</td>
</tr>
<tr>
<td>filespec</td>
<td>Designates a complete TOPS-20 file specification</td>
</tr>
<tr>
<td>IPCF</td>
<td>IPCF capability required</td>
</tr>
<tr>
<td>underline</td>
<td>Designates a variable argument, as in <code>1Bn</code></td>
</tr>
<tr>
<td>MNT</td>
<td>MAINTENANCE capability required</td>
</tr>
<tr>
<td>mss.</td>
<td>Milliseconds</td>
</tr>
<tr>
<td>number</td>
<td>Designates an octal number</td>
</tr>
<tr>
<td>number.</td>
<td>Designates a decimal number</td>
</tr>
<tr>
<td>number.number</td>
<td>Designates a floating point number</td>
</tr>
<tr>
<td>OPR</td>
<td>OPERATOR capability required</td>
</tr>
<tr>
<td>OWGBP</td>
<td>A One Word Global Byte Pointer; see Pointer Formats for format</td>
</tr>
<tr>
<td>(priv)</td>
<td>Designates a capability that must exist for the specified function to be legal, but need not be enabled</td>
</tr>
<tr>
<td>R-J</td>
<td>The data is or should be right-justified in the specified field</td>
</tr>
<tr>
<td>&lt;value&gt;,&lt;value&gt;</td>
<td>The left and right half-word (18-bit) values of a full-word (36-bit) value</td>
</tr>
<tr>
<td>WHL</td>
<td>WHEEL capability required</td>
</tr>
</tbody>
</table>
MONITOR CALLS FUNCTIONAL ORGANIZATION

Accounting Functions

GACCT  Reads a job's account
GACTF  Reads a file's account
LOGIN  Logs a job into the system
SACTF  Sets a file's account
USAGE  Writes entries into the system's accounting file
VACCT  Validates an account

File Functions

ACCES  Allows access to a directory
BKJFN  Backspaces file's pointer
CHFDB  Changes a File Descriptor Block
CHKAC  Checks access to a file
CLOSF  Closes a file
CLZFF  Closes a process' files
CRLNM  Creates a logical name
DELF   Deletes a file
DELFN  Retains specified number of generations of file
DIRST  Translates directory or user number to a string
FFFFP  Finds first free file page
FFUFID  Finds first used file page
GACTF  Reads a file's account
GFUST  Reads the author or last writer name string
GNJFN  Assigns a JFN to the next file
GPJFN  Returns primary JFN's
GTJFN  Assigns a JFN to a file
GTSTS  Reads file's status
INLNM  Writes logical names
JFNS   Translates a JFN to a string
LNMST  Translates logical name to string
OPENF  Opens a file
RCDIR  Translates directory name to number
RCUSR  Translates user name to number
RFBSZ  Reads file's byte size
RFPTR  Reads file's pointer
RFTAD  Reads file's time and dates
RLJFN  Releases a JFN
RNAMF  Renames a file
SACTF  Sets a file's account
SFBSZ  Sets file's byte size
SPTR   Sets file's pointer
SFTAD  Sets file's time and dates
SPUST  Changes the author or last writer name string
SIZEF  Obtains file's length
SPJFN  Sets primary JFN's
STSTS  Sets file's status
SWJFN  Transposes two JFN's
UPFGS  Updates file's pages
WILD%  Compares a wild filespec against a non-wild filespec

I/O Functions

BIN    Reads the next byte
BOUT   Writes the next byte
DUMPI  Reads data in unbuffered data mode
DUMPO  Writes data in unbuffered data mode
FLIN   Reads a floating-point number
FLOUT  Writes a floating-point number
NIN    Reads a number
NOUT   Writes a number
PSOUT  Writes string to primary output designator
PBIN   Reads byte from primary input designator
PBOUT  Output byte to primary output designator
PMAP   Maps pages
RDTTY  Reads data from primary input designator
RIN    Reads a byte nonsequentially
ROUT   Writes a byte nonsequentially
RSCAN  Reads and outputs rescan buffer
SIN    Reads a string
SOUT   Writes a string
SINR   Reads a record
SOUTR  Writes a record
SMAP%  Maps sections
TEXTI  Reads data from terminal or file

Information Functions

ERSTR  Translates an error code to a string
ESSTR  Returns an error string
GETAB  Returns a word from a system table
GETER  Returns the last error condition
GETJI  Returns job information for specified job
GETNM  Returns the program name being used by the job
GUINF  Returns job information for current job
GTAD   Returns the system's date
GTDAL  Returns the disk allocation of a directory
GTRPI  Returns the paging trap information
GTRPW  Returns the trap words
HPTIM  Returns the high-precision clock values
RUNITM Returns the runtime of a job or process
SYSGST Returns values for a system table
TIME   Returns the time since the system was restarted
### Device Control Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASND</td>
<td>Assigns a device</td>
</tr>
<tr>
<td>ATACH</td>
<td>Attaches controlling terminal to a job</td>
</tr>
<tr>
<td>CFIBF</td>
<td>Clears terminal's input buffer</td>
</tr>
<tr>
<td>CFOBF</td>
<td>Clears terminal's output buffer</td>
</tr>
<tr>
<td>DEVST</td>
<td>Translates a device designator to a string</td>
</tr>
<tr>
<td>DIBE</td>
<td>Dismisses until terminal input buffer is empty</td>
</tr>
<tr>
<td>DDBE</td>
<td>Dismisses until terminal output buffer is empty</td>
</tr>
<tr>
<td>DTACH</td>
<td>Detaches controlling terminal from a job</td>
</tr>
<tr>
<td>DVCHR</td>
<td>Returns device characteristics</td>
</tr>
<tr>
<td>GDSKC</td>
<td>Returns disk usage</td>
</tr>
<tr>
<td>GDST</td>
<td>Returns the device status</td>
</tr>
<tr>
<td>GTTYP</td>
<td>Returns terminal type number</td>
</tr>
<tr>
<td>LPINI</td>
<td>Loads VFU or translation RAM</td>
</tr>
<tr>
<td>NSTR</td>
<td>Performs structure-dependent functions</td>
</tr>
<tr>
<td>MTOPR</td>
<td>Performs device-dependent functions</td>
</tr>
<tr>
<td>MTUX</td>
<td>Performs functions for logical tape devices</td>
</tr>
<tr>
<td>RELD</td>
<td>Releases a device</td>
</tr>
<tr>
<td>RFCOC</td>
<td>Returns control character output control words</td>
</tr>
<tr>
<td>RFMOD</td>
<td>Returns the JFN mode word</td>
</tr>
<tr>
<td>RPPDS</td>
<td>Returns current position of the terminal</td>
</tr>
<tr>
<td>SDSTS</td>
<td>Sets the device status</td>
</tr>
<tr>
<td>SFCOC</td>
<td>Sets control character output control words</td>
</tr>
<tr>
<td>SFMOD</td>
<td>Sets program-related fields in the JFN mode word</td>
</tr>
<tr>
<td>SFPDS</td>
<td>Sets position of the terminal's cursor</td>
</tr>
<tr>
<td>SIBE</td>
<td>Skips if input buffer is empty</td>
</tr>
<tr>
<td>SDBE</td>
<td>Skips if output buffer is empty</td>
</tr>
<tr>
<td>SDBF</td>
<td>Skips if output buffer is full</td>
</tr>
<tr>
<td>SPDDL</td>
<td>Defines and initializes input spooling</td>
</tr>
<tr>
<td>STDEV</td>
<td>Translates a string to a device designator</td>
</tr>
<tr>
<td>STPAR</td>
<td>Sets device-related fields in the JFN mode word</td>
</tr>
<tr>
<td>STTYP</td>
<td>Sets terminal type number</td>
</tr>
<tr>
<td>TLINK</td>
<td>Controls terminal linking</td>
</tr>
</tbody>
</table>

### Software Interrupt System Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC</td>
<td>Activates interrupt channels</td>
</tr>
<tr>
<td>ATI</td>
<td>Assigns terminal code to channel</td>
</tr>
<tr>
<td>CIS</td>
<td>Clears the interrupt system</td>
</tr>
<tr>
<td>DEBRK</td>
<td>Dismisses current interrupt</td>
</tr>
<tr>
<td>DIC</td>
<td>Deactivates interrupt channels</td>
</tr>
<tr>
<td>DIR</td>
<td>Disables the interrupt system</td>
</tr>
<tr>
<td>DTI</td>
<td>Deassigns terminal code</td>
</tr>
<tr>
<td>EIR</td>
<td>Enables the interrupt system</td>
</tr>
<tr>
<td>GTRPW</td>
<td>Returns trap words</td>
</tr>
<tr>
<td>IIC</td>
<td>Initiates interrupts on specific channels in a process</td>
</tr>
<tr>
<td>RCM</td>
<td>Reads activated channel word mask</td>
</tr>
<tr>
<td>RIR</td>
<td>Reads the interrupt table addresses for a single-section program</td>
</tr>
<tr>
<td>RIRCM</td>
<td>Reads inferior reserved channel mask</td>
</tr>
<tr>
<td>RTIW</td>
<td>Reads terminal interrupt word</td>
</tr>
<tr>
<td>RWM</td>
<td>Reads waiting channel word mask</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
Monitor Calls Functional Organization

SCTTY  Changes source of terminal interrupts
SIR    Sets the interrupt table addresses for a single-section process
SIRCM  Sets inferior reserved channel mask
SKPIR  Skips if the interrupt system is enabled
STIW   Sets terminal interrupt word
XTFW%  Returns page-fault words
XRIR%  Reads the interrupt table addresses for a multiple-section program
XSIR%  Sets the interrupt table addresses for a multiple-section process

Process/Capability Handling Functions

ADBRK  Controls address breaks
CFORK  Creates interior process
DISMS  Dismisses process for specified amount of time
EPCAP  Enables process capabilities word
FFORK  Freezes one or more processes
GFHKH  Gets process handle
GFKKS  Gets current process structure
HALTF  Halts a process
HFKRH  Halts an inferior process
KFORK  Kills one or more processes
PRAG   Sets or returns process argument block
RESET  Resets and initializes current process
RFDKS  Returns process' accumulators
RFORK  Resumes one or more processes
RFKRM  Releases process handles
RFSTS  Returns process' status
RMAP   Obtains a handle on a page in a process
RPACS  Returns accessibility of page
RRCAPS Returns process capabilities word
RSMAP% Returns information about the mapping of one section of a process
RTFRK  Returns the handle of a process suspended because of a monitor call intercept
RWSET  Releases working set
SFACS  Sets process' accumulators
SFORK  Starts a process in section zero
SPACS  Sets accessibility of page
SPLFK  Splices a process structure
TFORK  Sets and removes monitor call intercepts
UTFRK  Resumes a process suspended because of a monitor call intercept
WAIT   Dismisses process until interrupt occurs
WFORK  Waits for process to terminate
XSFRK% Starts a process in a non-zero section
Save File Handling Functions

GCVEC  Gets compatibility package entry vector
GDVEC  Gets RMS entry vector
GET    Obtains a saved file
GEVEC  Gets process entry vector of a single-section program
SAVE   Saves a process as nonsharable
SCVEC  Sets compatibility package entry vector
SDVEC  Sets RMS entry vector
SEVEC  Sets the entry vector for a single-section program
SFRKV  Starts process using its entry vector
SSAVE  Saves a process as sharable
XGVEC% Gets process entry vector for a multiple-section program
XSRK%  Starts a process using a user-supplied, global PC
XSVEC% Sets the entry vector for a multiple-section program

Date/Time Conversion Functions

GTAD   Gets current date and time in internal format
IDCNV  Converts from day, month, year to internal date and time
IDTIM  Inputs date and time, converting to internal format
IDTNC  Inputs date and time without converting to internal format
ODCNV  Converts from internal date and time to day, month, year
ODTIM  Outputs date and time, converting from internal format to text
ODTNC  Outputs date and time in internal format

Archive/Virtual Disk Functions

ARCF   Performs archive/virtual-disk operations
CRDIR  Creates or modifies a directory
DELDIF Expunges deleted files
DELNIF  Retains specified number of generations of file
GTJFN Assigns a JFN to a file
GNJFN  Assigns a JFN to the next file
JFNS   Translates a JFN to a string
OPENF  Opens a file
RFTAD  Reads file's time and dates
SETJ  Sets job parameters
SFTAD  Sets file's time and dates
SMON   Sets monitor flags
TMON   Reads monitor flags
Privileged Functions

NOTE: Calls marked with an asterisk (*) require privileges for specific functions only.

ACCES* Accesses a directory
ALLOC Allocates a device to a particular job
ARCQ* Performs archive/virtual-disk operations
ASNQ* Assigns ARPA message queue
ATTACH* Attaches job to new controlling terminal
BOOT Performs functions required for loading front-end software
CRDIR* Creates or modifies a directory
CREJ* Creates a new job
DELDF* Expunges deleted files
DEL* Deletes files
DIAG Reserves and releases hardware channels
DSAS Assigns specific disk addresses
DSKP Allows hardware address specification in disk transfers
ENQ* Places a request in ENQ/DEQ resource queue
ENQQ* Returns status of a resource
FHLST Flushes an ARPA host
GACCT Returns job account information
GIVOK% Allows/denies access to a protected system resource
GTDIR* Returns directory information
HALTF* Halts a process
HYS Halts the monitor
LGOUT* Logs a job out
LPINI Loads line-printer VFU
MDDT* Enters MDDT program
MRCV* Retrieves IPCF message
MSEND* Sends IPCF message
MFRK Starts a process in monitor mode
MSTR* Performs structure-related functions
MTALN Associates magnetic tape drive with logical unit number
MTOPR* Performs device-related functions
MTU* Performs MT-device functions
MUTL* Performs IPCF functions
NDE* Performs DECN functions
NTMNX* Performs DECN network management functions
PEEK Reads monitor data
PLCK Locks physical pages
PMCTL Controls physical memory
RCVOK% Services GITOK% requests
SETUP* Sets job parameters
SFTAD* Sets file date/time
SFUST* Sets file author
SUPRI Sets job priority
SKED* Manipulates scheduler database
SMON Sets monitor flags
SNQPR Performs system performance analysis
SPOOL Performs spooling-related functions
SPRIW Sets process priority
STAD* Sets system date/time
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STI*</td>
<td>Simulates terminal input</td>
</tr>
<tr>
<td>SYERR</td>
<td>Places information in the System Error file</td>
</tr>
<tr>
<td>TTMSG*</td>
<td>Sends a message to a terminal</td>
</tr>
<tr>
<td>USAGE</td>
<td>Makes entries in accounting file</td>
</tr>
<tr>
<td>USRIO</td>
<td>Places program in user I/O mode</td>
</tr>
<tr>
<td>UTEST</td>
<td>Monitors executed instructions</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
ACCES

TOPS-20 MONITOR CALLS

ACCES JSYS 552

FUNCTION
Gives a particular type of access to a given directory.

RESTRICTIONS
Requires WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: BO(AC%CDN) Connect job to directory
     B1(AC%OWN) Give job owner access to directory
     B2(AC%REM) Relinquish owner access to directory
     B16-35 Length of argblk
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK
Word Symbol Meaning
0 .ACDIR 36-bit directory number or byte pointer to
         ASCIZ string containing full directory name
1 .ACP5W Byte pointer to ASCIZ string containing
         password of specified directory
2 .ACJOB Job # or -1 for current job (WHL/OPR if not
         -1)

ADBRK JSYS 570

FUNCTION
Controls address breaks.

RESTRICTIONS
Not available on KS-10 hardware.

CALLING SEQUENCE
AC1: <function code>..<process handle>
AC2: Address of location at which to break (.ABSET only)
AC3: Flags (.ABSET only)
     BO(AB%RED) Break on read reference
     B1(AB%WRT) Break on write reference
     B2(AB%XCT) Break on execute reference

RETURNS +1: Always, with
AC2: Address of break location
AC3: Flags (.ABRED only)
     BO(AB%RED) Break set for read
     B1(AB%WRT) Break set for write
     B2(AB%XCT) Break set for execute
FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ABSET</td>
<td>Set address break</td>
</tr>
<tr>
<td>1</td>
<td>ABRED</td>
<td>Read address break</td>
</tr>
<tr>
<td>2</td>
<td>ABCLR</td>
<td>Clear address break</td>
</tr>
<tr>
<td>3</td>
<td>ABGAD</td>
<td>Return address of break instruction</td>
</tr>
</tbody>
</table>

AIC  JSYS 131

FUNCTION
Activates specific software interrupt channels.

CALLING SEQUENCE
AC1: Process handle
AC2: 36-bit word (18n activates channel n)

RETURNS +1: Always

ALLOC  JSYS 520

FUNCTION
Allocates a device to a job or to the device pool.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code (.ALCAL)
AC2: Device designator
AC3: Job # to allocate designated device, -1 to deallocate designated device, or -2 to assign device to monitor's resource allocator

RETURNS +1: Failure, error code in AC1
        +2: Success

ARCF  JSYS 247

FUNCTION
Performs archive and virtual disk operations.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: UFN
AC2: Function code
AC3: Function-specific argument
FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.ARRAR</td>
<td>Set/clear user request for archival</td>
<td>AC3: 0(ARCLR) to clear; 1(ARSET) to set</td>
</tr>
<tr>
<td>1</td>
<td>.ARRIV</td>
<td>Set/clear system request for file migration</td>
<td>AC3: 0(ARCLR) to clear; 1(ARSET) to set</td>
</tr>
<tr>
<td>2</td>
<td>.AREXm</td>
<td>Set/clear exemption from involuntary migration (enabled WHL/OPR)</td>
<td>AC3: 0(ARCLR) to clear; 1(ARSET) to set</td>
</tr>
<tr>
<td>3</td>
<td>.ARRFR</td>
<td>Request that contents of file be restored to disk</td>
<td>AC3: 1BO(AR%NMS) Don't send msg when restored</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1B1(AR%WAT) Wait for file</td>
</tr>
<tr>
<td>4</td>
<td>.ARDIS</td>
<td>Discard tape information for file</td>
<td>AC3: 1BO(AR%R1) Clear run 1 information</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1B1(AR%R2) Clear run 2 information</td>
</tr>
<tr>
<td>5</td>
<td>.ARSST</td>
<td>Set tape information for file; (enabled WHL/OPR)</td>
<td>AC3: Pointer to argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AC3: JFN for a DUMPER temporary file</td>
</tr>
<tr>
<td>6</td>
<td>.ARRST</td>
<td>Restore file to disk; (enabled WHL/OPR)</td>
<td>AC3: Pointer to argblk</td>
</tr>
<tr>
<td>7</td>
<td>.ARGST</td>
<td>Get tape information for file</td>
<td>AC3: Pointer to argblk</td>
</tr>
<tr>
<td>10</td>
<td>.ARRF1</td>
<td>Retrieve for file failed; (WHL/OPR)</td>
<td>AC3: 0(ARCLR) to clear; 1(ARSET) to set</td>
</tr>
<tr>
<td>11</td>
<td>.ARNSR</td>
<td>Set/clear resist involuntary migration</td>
<td>AC3: 0(ARCLR) to clear; 1(ARSET) to set</td>
</tr>
</tbody>
</table>

Argument Block for Functions .ARSST and .ARGST

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.ARDF</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO(AR%O1) Set information for run 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(AR%O2) Set information for run 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(AR%DFL) Delete content of disk file when done</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B3(AR%ARC) Archive the file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B4(AR%CRQ) Clear archive and/or migration requests</td>
</tr>
<tr>
<td>1</td>
<td>.ARTP1</td>
<td>Tape 1 identification</td>
</tr>
<tr>
<td>2</td>
<td>.ARSF1</td>
<td>&lt;tape 1 saverset number&gt;,&lt;tape 1 file number&gt;</td>
</tr>
<tr>
<td>3</td>
<td>.ARTP2</td>
<td>Tape 2 identification</td>
</tr>
<tr>
<td>4</td>
<td>.ARSF2</td>
<td>&lt;tape 2 saverset number&gt;,&lt;tape 2 file number&gt;</td>
</tr>
<tr>
<td>5</td>
<td>.ARODT</td>
<td>Time/date of tape write in internal format</td>
</tr>
<tr>
<td>6</td>
<td>.ARPSZ</td>
<td>Number of pages in file</td>
</tr>
</tbody>
</table>
ASND _ JSYS 70

FUNCTION
Assigns a device to the caller

CALLING SEQUENCE
AC1: Device designator

RETURNS  +1: Failure, error code in AC1
          +2: Success

ASNS0 _ JSYS 752

FUNCTION
Assigns a special message queue to a job.

RESTRICTIONS
For ARPANET systems only; requires enabled NET WIZARD capability.

CALLING SEQUENCE
AC1: Mask
AC2: Header value

RETURNS  +1: Failure, error code in AC1
          +2: Success, special message queue assigned with
              queue handle in AC1

ATACH _ JSYS 116

FUNCTION
Detaches the specified job from its controlling terminal (if any) and optionally attaches it to a new controlling terminal.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: BO(ATXCCJ)  Generate a CTRL/C interrupt to attached job
       B1(AT%NAT)  Do not attach job
       B2(AT%XTRM) Attach job to terminal specified in AC4
       B18-35(AT%JOB) Job # of desired job
AC2: Logged-in user number of job to be attached
AC3: Byte pointer to ASCIZ password string
AC4: Number of terminal to be attached to specified job

RETURNS  +1: Failure, error code in AC1
          +2: Success
ATI JSYS 137

FUNCTION
Assigns a terminal code to a software interrupt channel.

CALLING SEQUENCE
AC1: <terminal interrupt code>,<channel number>

RETURNS +1: Always

ATNVT JSYS 274

FUNCTION
Creates the Network Virtual Terminal (NVT) connection.

RESTRICTIONS
For ARPANET systems only

CALLING SEQUENCE
AC1: Flags,,<JFN of opened receive connection>
   B2(ATNVT) Indicates new (1) or old (0) TELNET protocol
AC2: JFN of opened send connection

RETURNS +1: Failure, with error code in AC1
       +2: Success, with NVT-specific terminal designtor in AC1

BIN JSYS 50

FUNCTION
Inputs the next byte from the specified source.

CALLING SEQUENCE
AC1: Source designtor

RETURNS +1: Always, with the byte right-justified in AC2
       or 0 indicating EOF.

BKJFN JSYS 42

FUNCTION
Backs up the source designtor’s pointer by one byte.

RESTRICTIONS
Cannot be used with DECNET devices SRV: or DCN:.

CALLING SEQUENCE
AC1: Source designtor
RETURNS +1: Failure, errcr code in AC1
+2: Success, updated byte pointer in AC1, if pertinent

BOOT USYS 562

FUNCTION
Performs basic maintenance and utility functions required for loading and dumping communications software.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Always

FUNCTION CODES
KS-10 Processor Functions

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Argblk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.BTRM</td>
<td>Put line in MOP mode; activate front end bootstrap ROM</td>
</tr>
<tr>
<td>0</td>
<td>.BTPRT</td>
<td>Line number</td>
</tr>
<tr>
<td>1</td>
<td>.BTLD</td>
<td>Load secondary bootstrap program into front end</td>
</tr>
<tr>
<td>0</td>
<td>.BTPRT</td>
<td>Line number</td>
</tr>
<tr>
<td>1</td>
<td>.BTSEC</td>
<td>Address of bootstrap program</td>
</tr>
<tr>
<td>2</td>
<td>.BTLD</td>
<td>Load front-end memory using previously loaded secondary or tertiary bootstrap program</td>
</tr>
<tr>
<td>0</td>
<td>.BDTE</td>
<td>Line number</td>
</tr>
<tr>
<td>1</td>
<td>.BTCNT</td>
<td>Number of bytes to transfer</td>
</tr>
<tr>
<td>2</td>
<td>.BTLPT</td>
<td>Pointer to data to be loaded</td>
</tr>
<tr>
<td>4</td>
<td>.BTIPR</td>
<td>Generate and link DDCMP Station Table; start up lines/terminals not previously known to system</td>
</tr>
<tr>
<td>0</td>
<td>.BTPRT</td>
<td>Drop,&lt;line number&gt;</td>
</tr>
<tr>
<td>1</td>
<td>.BTPRV</td>
<td>Protocol version number to use</td>
</tr>
<tr>
<td>5</td>
<td>.BTTPR</td>
<td>Stop protocol currently running on front end or line</td>
</tr>
<tr>
<td>0</td>
<td>.BTPRT</td>
<td>Line number</td>
</tr>
<tr>
<td>6</td>
<td>.BTSTS</td>
<td>Return status type of protocol running on front end to specified DTE or line, and name of adjacent DECNET node for this front end</td>
</tr>
<tr>
<td>0</td>
<td>.BTPRT</td>
<td>Line number</td>
</tr>
<tr>
<td>1</td>
<td>.BTCOD</td>
<td>Returned protocol version type; or -1 if no protocol is running</td>
</tr>
<tr>
<td>10</td>
<td>.BTRMP</td>
<td>Read MOP message from front end using previously loaded secondary or tertiary</td>
</tr>
</tbody>
</table>
bootstrap program
0 .BTPRT  Line number
1  Not used; must be zero
2  Not used; must be zero
3  Not used; must be zero
4 .BTCNT  Number of bytes to transfer
5 .BTMPT  Pointer to MOP message destination

11 .BTKML  Load KMC11, or CRAM, DRAM, and 4 UNIBUS registers
0 .BTKMC  KMC11 address
1 .BTKER  <error flags>,<bad data word>
   (16-bit)
   BO(BTKCVE) CRAM verify error
   B1(BTKDVE) DRAM verify error
   B2(BTKRVE) Register verify error
2 .BTKCC  Count of CRAM data
3 .BTKCP  Pointer to CRAM data (16-bit)
4 .BTKDC  Count of DRAM data
5 .BTKDP  Pointer to DRAM data (8-bit)
6 .BTKRC  Count of register data
7 .BTKRP  Pointer to register data
   (16-bit)
8 .BTKSA  If $BO, $B18-$35 contain start address
   BO(BTKSAA) Right half >0; start
   KMC11

12 .BTKMD  Dump KMC11, or CRAM, DRAM, and registers if space provided
0 .BTKMC  KMC11 address
1  Not used; must be zero
2 .BTKCC  Count of CRAM data
3 .BTKCP  Pointer to CRAM data (16-bit)
4 .BTKDC  Count of DRAM data
5 .BTKDP  Pointer to DRAM data (8-bit)
6 .BTKRC  Count of register data
7 .BTKRP  Pointer to register data
   (16-bit)

13 .BTRLC  Return line counters
0 .BTPRT  Port number
1 .BTZTM  Time since counters were last zeroed
2 .BTSCC  # of status counts to return
3 .BTSCP  Pointer to area for status counters
4 .BTRCC  # of receive counts to return
5 .BTRCP  Pointer to area for receive counters
6 .BTCC  # of transmit counts to return
7 .BTTCP  Pointer to area for transmit counters

14 .BTCLI  Convert line ID to port number
0 .BTPRT  Port number

15 .BTCNP  Convert NSP port number to line ID
0 .BTPRT  Port number
16 .BTSTA Set station's polling state to activate/deactivate terminal polling
(Requires VT62)
  0 .BTPRT Drop,,<line number>
  1 .BTCOD Flags
      0 .BTACT Set line active
      1 .BTIDL Set line idle

17 .BTSSP Set start-up priority value (Requires VT62)
  0 .BTPRT Line number
  1 .BTSPR Start priority count

20 .BTSTP Set polling priority (Requires VT62)
  0 .BTPRT Drop,,<line number>
  1 .BTPRI Priority value: 1 (high) to 5

21 .BTSD Send a DDCMP message
  0 .BTPRT Drop,,<line number>
  1 .BTMSG Address of or byte pointer to message
  2 .BTLEN Byte count of message

22 .BTRDD Receive a DDCMP message; .BTLEN is set to zero if queue is empty
  0 .BTPRT Line number
  1 .BTMSG Address of or byte pointer to buffer
  2 .BTLEN Size of user buffer
      Returned in .BTLEN:
      100(BT%CTL) +
      .BTSUP (1) - station came up
      .BTSDW (2) - station went down
      .BTCOMP (3) - transmit complete
      .BTSSF (4) - start-up failed

23 .BTCHN Set interrupt channel
  0 .BTPRT Drop,,<line number>
  1 .BTCOD Software interrupt channel

24 .BTLS Set type of line service for synchronous communications lines
  0 .BTPRT Drop,,<line number>
  1 .BTCOD Define protocol

KL-10 Processor Functions

Code Symbol Meaning/Argblk
0 .BTRDM Put line in MDP mode; activate front end bootstrap ROM
  0 .BDTE DTE-20 number
  1 .BTERR Error flags on failure (RET)

1 .BTLD Load secondary bootstrap program into front end
  0 .BDTE DTE-20 number
  1 .BTERR Error flags on failure (RET)
  2 .BTSEC Address of bootstrap to load

2 .BTLOD Load front-end memory using previously loaded secondary or tertiary bootstrap program
  0 .BDTE DTE-20 number
  1 .BTERR Error flags on failure (RET)
  2 Not used; must be zero
  3 .BTFLG User-supplied flag word
      BO(BTXBEL) Send to -11 doorbell
TOPS-20 Monitor Calls Quick Reference Guide

BOOT

3 .BTDM
Dump front-end memory using ROM bootstrap program
0 .BTDE DTE-20 number
1 .BTERR Error flags on failure (RET)
2 .BLTPT Pointer to data to be loaded
3 .BTNCNT Number of bytes to transfer
4 .BDPT Pointer to dump data destination

4 .BTIPR
Initialize front-end protocol
0 .BTDE DTE-20 number
1 .BTIPR Protocol version number to use
5 .BTIPR Stop protocol currently running on front end or line

5 .BTMP
Read data from front-end using previously loaded secondary or tertiary bootstrap program
0 .BTDE DTE-20 number
1 .BTERR Error flags on failure (RET)
2 .BLTPT Pointer to data destination
3 .BTCNT Maximum # of bytes to transfer
4 .BTCNT Maximum # of bytes to transfer

7 .BTMP
Convert line ID to port number
1 .BTMP Pointer to ASCLL line ID

10 .BTMP
Convert NSP port number to line ID
1 .BTMP Pointer to ASCLL line ID

14 .BTMP
Send message to or receive message from DN60 front-end using .VND60 protocol (Requires DN60 on KL-10 Model B)
0 .BTDE DTE number
1 .BT6ERR Error flags (RET)
30 D6%BDP Byte pointer is bad
31 D6%ARD -11 attempted to send data
32 D6%TRD DTSRV timed out waiting for response header from -11
33 D6%TDI DTSRV timed out waiting for data from -11
34 D6%TPD DTSRV timed out

16
waiting for DTE to be free
35 D6%NT6 -11 is not running DN60 protocol
2 .BT6HBC B0-17 DN60 header byte count
  .BT6HDR B18-35 DN60 header address
3 .BT6DBC Number of bytes of data
4 .BT6PTR Pointer to first byte of data
5 .BT6TMR Time request was made (RET)
6 .BT6TAS Time DTE was assigned (RET)
7 .BT6THQ Time TOPS-20 queued header to DTE (RET)
10 .BT6TRD Time
11 .BT6TDD Time
12 .BT6TFR Time TOPS-20 satisfied request

BOUT JSYS 51

FUNCTION
Outputs a byte sequentially to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: Byte to be output, right-justified

RETURNS +1: Always

CACCT JSYS 4

FUNCTION
Changes the account for the current job.

RESTRICTIONS
In non-zero sections, OWGBP's must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Byte pointer to the new account string; in section 0, may contain <SB2+<account number>B35>

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC1

CFIBF JSYS 100

FUNCTION
Clears the designated file input buffer.

CALLING SEQUENCE
AC1: Source designator

17
TOPS-20 Monitor Calls Quick Reference Guide
CFIBF

RETURNS  +1: Always

CFGBF   JSYS 101

FUNCTION
Clears the designated file output buffer

CALLING SEQUENCE
AC1:  Destination designator

RETURNS  +1: Always

CFORK   JSYS 152

FUNCTION
Creates a process inferior to the calling process.

CALLING SEQUENCE
AC1:  BO(CR%MAP)  Make inferior process' map same as current process' map
       B1(CR%CAP)  Make inferior process' capabilities same as current process' capabilities
       B3(CR%ACS)  Set inferior process' ACs from block whose address is in AC2
       B4(CR%ST)  Set PC of inferior process to value in B1B-35 of AC1 and start process
       B18-35(CR%PCV)  PC value for inferior process if CR%ST is on
AC2:  Address of optional 20 word block containing AC values for inferior process

RETURNS  +1: Failure, error code in AC1
          +2: Success, relative process handle in AC1

CHFDB   JSYS 64

FUNCTION
Changes words in the File Descriptor Block for the specified file.

CALLING SEQUENCE
AC1:  BO(CF%NUD)  Don't wait for disk copy of directory to be updated
       B9-17(CF%DSP)  Index into FDB of word to be changed
       B1B-35(CF%JFN)  JFN for a disk file
AC2:  Mask indicating bits to be changed; -1 if changing a count value in AC3
AC3:  New values for changed bits corresponding to mask given in AC2
RETURNS +1: Always

FUNCTION
Checks if a user is allowed access to files in a given directory.

Restrictions
In non-zero sections, OWGBPs must specify 7-bit bytes.

Calling sequence
AC1: Flags, <length of argblk>
    BO(CK%JFN) JFN in word .CKAUD of the argblk
AC2: Address of argblk

Returns
+1: Failure, error code in AC1
    +2: Success, access check is completed, with AC1 containing -1 if access is allowed or 0 if access is not allowed

Argument Block

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.CKAAC</td>
<td>Code of desired access to files</td>
</tr>
<tr>
<td>1</td>
<td>.CKALD</td>
<td>Byte pointer to username string, or 36-bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>user number</td>
</tr>
<tr>
<td>2</td>
<td>.CKACD</td>
<td>Byte pointer to directory name string, or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36-bit directory number of user's connected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>directory</td>
</tr>
<tr>
<td>3</td>
<td>.CKAEC</td>
<td>Enabled capabilities of user</td>
</tr>
<tr>
<td>4</td>
<td>.CKAUD</td>
<td>Byte pointer to directory name string, or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36-bit directory number of directory being</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessed; if BO(CK%JFN) is on, contains JFN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for file being accessed</td>
</tr>
<tr>
<td>5</td>
<td>.CKAPR</td>
<td>Protection of files being accessed; (not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>required if a JFN is supplied in word .CKAUD</td>
</tr>
</tbody>
</table>

Access Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.CKARD</td>
<td>Read existing files</td>
</tr>
<tr>
<td>1</td>
<td>.CKAWR</td>
<td>Write existing files</td>
</tr>
<tr>
<td>2</td>
<td>.CKAXE</td>
<td>Execute existing files</td>
</tr>
<tr>
<td>3</td>
<td>.CKAAP</td>
<td>Append to existing files</td>
</tr>
<tr>
<td>4</td>
<td>.CKADL</td>
<td>Obtain directory listing of existing files</td>
</tr>
<tr>
<td>6</td>
<td>.CKADR</td>
<td>Read the directory</td>
</tr>
<tr>
<td>10</td>
<td>.CKACN</td>
<td>Connect to the directory</td>
</tr>
<tr>
<td>11</td>
<td>.CKACF</td>
<td>Create files in the directory</td>
</tr>
</tbody>
</table>
CIS JSYS 141

FUNCTION
Clears the software interrupt system for the current process.

RETURNS +1: Always

CLOSF JSYS 22

FUNCTION
Closes a specific file or all files.

CALLING SEQUENCE
AC1: B0(CO%NRJ) Do not release the JFN
      B6(CZ%ABT) Abort any output operations currently being done
      B7(CZ%NUD) Do not update copy of directory on disk
      B18-35(CD%JFN) JFN of file being closed

RETURNS +1: Failure, error code in AC1
          +2: Success

CLZFF JSYS 34

FUNCTION
Closes all files and/or releases all JFNs at or below a specified process.

CALLING SEQUENCE
AC1: B0(CZ%NIF) Do not close files of inferior processes
     B1(CZ%NSF) Do not close files of this process
     B2(CZ%NRJ) Do not release JFNs
     B3(CZ%NCL) Do not close any files; only release nonopen JFNs
     B4(CZ%UNR) Unrestrict files opened with restricted access for specified process
     B5(CZ%ARJ) Wait until file can be closed, close it, and release JFNs
     B6(CZ%ABT) Abort any output operations currently being done
     B7(CZ%NUD) Do not update copy of directory on disk
     B18-35(CZ%PRH) Process handle

RETURNS +1: Always
FUNCTION
Parses one or more fields of a command that is either typed by a user or contained in a file.

CALLING SEQUENCE
AC1: Address of the command state block
AC2: Address of first alternate function descriptor block

RETURNS +1: Always (unless a reparse is needed and the right half of .CMFLG is nonzero), with
AC1: Flags,<address of command state block>
AC2: Data obtained for field; or error code
if field could not be parsed (CM%NDP is on)
AC3: B0-17 Address of function descriptor block given
     B18-35 Address of function descriptor block used

COMMAND STATE BLOCK
Word Symbol Meaning
0 .CMFLG <flag bits>,<reparse dispatch address>
1 .CMIQJ <input JFN>,<output JFN>
2 .CMRTY Byte pointer to beginning of the prompting text
3 .CMBFP Byte pointer to beginning of the user's input
4 .CMPTR Byte pointer to beginning of next field to be parsed
5 .CMCNT Count of space remaining in buffer after .CMPTR pointer
6 .CMINC Count of number of unparsed characters in buffer after .CMPTR pointer
7 .CMABP Byte pointer to atom buffer containing last field parsed by CMDND
10 .CMABC Size of atom buffer in bytes
11 .CMGJB Address of 16 word, writable GTJFN argbix

Settable Bits in Word .CMFLG of the Command State Block
Bit Symbol Meaning
6 CM%RAI Convert lowercase input to uppercase
7 CM%XIF "#" is punctuation, not indirect file designator
8 CM%WKF Begin parsing after each field is terminated without waiting for action character (CRLF, ESC, CTRL/F, ?)

FUNCTION DESCRIPTOR BLOCK
Word Symbol Meaning
0 .CMFNP Function code and pointer to next function
descriptor block
BO-8(CM%FNC) Function code
B9-17(CM%FFL) Function-specific flags
B18-35(CM%LST) Address of next function descriptor block; or 0

21
if last
1. CMDAT Data for the specific function, if any
2. CMHLP Byte pointer to help text for this field
3. CMDEF Byte pointer to default string for this field
4. CMBRK Address of 4-word break mask that specifies which characters terminate a field

FUNCTIONS FOR WORD .CMFNP OF THE FUNCTION DESCRIPTOR BLOCK

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CMKEY</td>
<td>Parse a keyword</td>
</tr>
<tr>
<td>1</td>
<td>CMNUM</td>
<td>Parse a number</td>
</tr>
<tr>
<td>2</td>
<td>CMNOI</td>
<td>Parse a guide word string</td>
</tr>
<tr>
<td>3</td>
<td>CMSWI</td>
<td>Parse a switch</td>
</tr>
<tr>
<td>4</td>
<td>CMIFI</td>
<td>Parse an input filespec</td>
</tr>
<tr>
<td>5</td>
<td>CMOFI</td>
<td>Parse an output filespec</td>
</tr>
<tr>
<td>6</td>
<td>CMFLD</td>
<td>Parse a general (arbitrary) filespec</td>
</tr>
<tr>
<td>7</td>
<td>CMFLD</td>
<td>Parse an arbitrary field</td>
</tr>
<tr>
<td>10</td>
<td>CMCFM</td>
<td>Wait for user to confirm command with CRLF</td>
</tr>
<tr>
<td>11</td>
<td>CMDIR</td>
<td>Parse a directory name</td>
</tr>
<tr>
<td>12</td>
<td>CMUSR</td>
<td>Parse a user name</td>
</tr>
<tr>
<td>13</td>
<td>CMCMA</td>
<td>Parse a comma</td>
</tr>
<tr>
<td>14</td>
<td>CMINI</td>
<td>Initialize the command line</td>
</tr>
<tr>
<td>15</td>
<td>CMFLT</td>
<td>Parse a floating-point number</td>
</tr>
<tr>
<td>16</td>
<td>CMDEV</td>
<td>Parse a device name</td>
</tr>
<tr>
<td>17</td>
<td>CMTXT</td>
<td>Parse input text up to next carriage return, place text in atom buffer, and return</td>
</tr>
<tr>
<td>20</td>
<td>CMRAD</td>
<td>Parse a date and/or time field according to setting of bits CMXIDA and CMXITM</td>
</tr>
<tr>
<td>21</td>
<td>CMQST</td>
<td>Parse a quoted string up to terminating quote</td>
</tr>
<tr>
<td>22</td>
<td>CMUQS</td>
<td>Parse an unquoted string up to one of the specified break characters</td>
</tr>
<tr>
<td>23</td>
<td>CMTOC</td>
<td>Parse input and compare it with a given string</td>
</tr>
<tr>
<td>24</td>
<td>CMNUM</td>
<td>Parse a number and terminate on 1st nonnumeric character</td>
</tr>
<tr>
<td>25</td>
<td>CMACR</td>
<td>Parse an account string</td>
</tr>
<tr>
<td>26</td>
<td>CMNOD</td>
<td>Parse a network node name</td>
</tr>
</tbody>
</table>

Function-specific Flags in B9-B17 (CMXFFL) of Word .CMFNP

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>CM%NSF</td>
<td>Suffix is optional; functions .CMDEV and .CMNOD only</td>
</tr>
<tr>
<td>13</td>
<td>CM%BRK</td>
<td>Word .CMBRK of function descriptor block contains a pointer to a 4-word break mask</td>
</tr>
<tr>
<td>14</td>
<td>CM%PD</td>
<td>Field is parse only (no existence verification); functions .CMDEV, .CMDIR, .CMNOD, and .CMUSR only</td>
</tr>
<tr>
<td>15</td>
<td>CM%HPP</td>
<td>Byte pointer to program-supplied help message for field is in word 2 (.CMHLP) of function descriptor block</td>
</tr>
<tr>
<td>16</td>
<td>CM%DPP</td>
<td>Byte pointer to program-supplied default string for field is in word 3 (.CMDEF) of function descriptor block</td>
</tr>
<tr>
<td>17</td>
<td>CMXSDH</td>
<td>Suppress output of default help message if user types a question mark</td>
</tr>
</tbody>
</table>
### ADDITIONAL DATA IN WORD .CMDAT OF THE FUNCTION DESCRIPTOR BLOCK

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.CMKEY</td>
<td>Address of keyword symbol table whose entries point to argblk; B18-35 of Word 0 of argblk may contain flags: B33(CM%ABR) Keyword is abbreviation, B34(CM%NOR) Do not recognize keyword, B35(CM%INV) Make keyword invisible.</td>
</tr>
<tr>
<td>.CMNUM</td>
<td>Radix of the number (from 2 to 10)</td>
</tr>
<tr>
<td>.CMNDAI</td>
<td>Byte pointer to an ASCII string that contains the guide word</td>
</tr>
<tr>
<td>.CMSWI</td>
<td>Address of switch keyword table, whose entries point to argblk; B18-35 of Word 0 of argblk may contain flags: B33(CM%ABR) Keyword is abbreviation, B34(CM%NOR) Do not recognize keyword, B35(CM%INV) Make keyword invisible.</td>
</tr>
<tr>
<td>.CMDIR</td>
<td>Data bits</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>&lt;flag bits&gt;, &lt;address of 8-word block&gt;</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>B0(CM%IDA) Parse a date, B1(CM%ITM) Parse a time, B2(CM%NCE) Do not convert date/time to internal format</td>
</tr>
<tr>
<td>.CMUOQS</td>
<td>Address of 4-word block of 128. break character mask bits</td>
</tr>
<tr>
<td>.CMFTOK</td>
<td>Byte pointer to the given string</td>
</tr>
<tr>
<td>.CMNUX</td>
<td>The radix (from 2 to 10) of the number</td>
</tr>
</tbody>
</table>

### DEFAULT HELP MESSAGES

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.CMKEY</td>
<td>ONE OF THE FOLLOWING if no keyword matches the currently typed field</td>
</tr>
<tr>
<td>.CMNUM</td>
<td>OCTAL NUMBER (radix 8), DECIMAL NUMBER (radix 10)</td>
</tr>
<tr>
<td>.CMNDAI</td>
<td>A NUMBER IN BASE nn (radix nn)</td>
</tr>
<tr>
<td>.CMSWI</td>
<td>None</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>ONE OF THE FOLLOWING</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>Depending on flag settings for</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>GT/JFN call, OUTPUT FILESPEC or</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>INPUT FILESPEC</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>None</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>CONFIRM WITH CARRIAGE RETURN</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>DIRECTORY NAME</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>USER NAME</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>COMMA</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>None</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>NUMBER</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>DEVICE NAME</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>TEXT STRING</td>
</tr>
<tr>
<td>.CMFAD</td>
<td>Depending on bits set in .CMDAT,</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

CMDND

.DATE, TIME, or DATE AND TIME

. CMQST (quoted) QUOTED STRING

. CMUQS (unquoted) UNQUOTED STRING if "?" is a break character

. CMTOK (token) None

. CMNUM (number) Same as .CMNUM

. CMACT (account) None

. CMNOD (node) NODE NAME

Functions That Use Masks (Word .CMBRK)

Mask Symbol Function Changeable by User
KEYBO. - KEYB3. .CMKEY Yes
DEVBO. - DEVB3. .CMDEV Yes (if parse-only)
FLDBO. - FLDB3. .CMFLD Yes
EOLBO. - EOLB3. .CMTXT Yes
KEYBO. - KEYB3. .CMSWI Yes
User-specified .CMDAT Yes
USRBO. - USRB3. .CMUSR No
FILBO. - FILL, .CMFIL No
FILBO. - FILL3. .CMFIL No
FILBO. - FILL3. .CMDFI No
internal .CMNUM No
FILBO. - FILL3. .CMDFI No
internal .CMFLT No
ACTBO. - ACTB3. .CMACT No

RETURNED BITS IN WORD .CMFLG OF THE FUNCTION DESCRIPTOR BLOCK

Bit Symbol Meaning
0 CMXESC ESC was typed by user as terminator for this field
1 CMXNOD Field could not be parsed because it did not conform to specified function(s)
2 CMXEDC Field was terminated with a carriage return
3 CMXRT Characters already parsed need to be reparsed because user edited them
4 CMXSWT Switch field was terminated with a colon
5 CMXPF Previous field was terminated with an ESC

CRDIR USYS 240

FUNCTION
Creates, changes, or deletes a directory entry.

RESTRICTIONS
Enabled WHEEL or OPERATOR capability required for some functions.

CALLING SEQUENCE
AC1: Byte pointer to ASCIZ string containing str:<directory>
AC2: BO(CD%LEN) Set flags and length of argblk from values in word .CDLEN
      B1(CD%PSW) Set password from argblk
TOPS-20 Monitor Calls Quick Reference Guide
CRDIR

B2(CD%LIQ) Set working disk storage limit from argblk
B3(CD%PRV) Set capability bits from argblk
B4(CD%MOD) Set mode bits from argblk
B5(CD%LOQ) Set permanent disk storage limit from argblk
B6(CD%NUM) Set directory number from argblk
B7(CD%FPT) Set default file protection from argblk
B8(CD%PT) Set directory protection from argblk
B9(CD%RET) Set default retention count from argblk
B10(CD%LLD) Set last LOGIN date from argblk
B11(CD%UGP) Set user groups from argblk
B12(CD%DG) Set directory groups from argblk
B13(CD%SQ) Set subdirectory quota from argblk
B14(CD%UG) Set user groups assignable by directory from argblk
B15(CD%DAC) Set default account from argblk
B17(CD%DEL) Delete this directory entry
B18-35(CD%APB) Address of the argblk

AC3: Byte pointer to ASCIZ string containing password of directory

RETURNS +1: Always, with directory number in AC1

ARGUMENT BLOCK
Word Symbol Meaning
0 .CDLEN <flag bits>,<length of argblk>
   B0(CD%NSQ) On restore, do not update superior directory’s quotas
               (enabled WHL/DPR required)
   B1 (CD%NC) On restore or reconstruction, do not change directory parameters
               if directory currently exists
               (enabled WHL/DPR)
   B2(CD%NED) Set default on-line expiration date from word .CDDNE
   B3(CD%FE) Set default on-line expiration date from word .CDDFE
1 .CDPWSW Byte pointer to password string
2 .CDLIQ Working disk storage quota
3 .CDPRV Capabilities for this user
4 .CDMOD Mode word
   B0(CD%DIR) Directory is files-only
   B1(CD%ANA) Obsolete
   B2(CD%RLM) Repeat messages from file
               <SYSTEM>MAIL.TXT each time user logs in
   B7(CD%DAR) File should be archived rather than migrated when on-line expiration date reached
5 .CDLOQ Permanent disk storage quota
6 .CDNUM Directory number (valid only when creating a directory)
7 .CDEPT Default file protection (18 bits, R-J)
10 .CDPDP Directory protection (18 bits, R-J)
11 .CDRET Default generation retention count

25
CRDIR

12 .CDLLD  Date of last login
13 .CDUGP  Address of user group list for this directory
14 .CDDGP  Address of directory group list
15 .CDSQ   Maximum number of sub-directories allowed
16 .CDUG  Address of user group list
17 .CDDAC  Byte pointer to default account string
20 .CDDNE  Default on-line expiration date and time
21 .CDDFE  Default off-line expiration date and time

DEFAULT ARGUMENTS

Bit Symbol  Default Argument
  2  CD%L1Q  250 working pages
  3  CD%PRV  No special capabilities
  4  CD%MOD  Directory name for login
  5  CD%LQ  250 permanent pages
  6  CD%NUM  First unused directory number
  7  CD%FTP  Default file protection to 777700
  8  CD%DPT  Directory protection to 777700
  9  CD%RET  Default file retention count to 1
10  CD%LLD  Never logged in
11  CD%UGP  No user groups
12  CD%DGP  No directory groups
13  CD%SDQ  No ability to create inferior directories
14  CD%UG  No assignable user groups for inferior directories
15  CD%DAC  No default account

CRJOB  JSYS 2

FUNCTION
Creates a new job and optionally logs it in.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions; in non-zero sections, OWGBP must specify 7-bit bytes.

CALLING SEQUENCE

AC1: <flag bits>,0
AC2: Address of argblk
AC3: Job # of previously-created job if B17(CJ%DSN) is on in AC1

RETURNS  +1: Failure, with error code in AC1
           +2: Success, with number of new job in AC1

Flag Bits in AC1

Bit Symbol  Meaning
  0  CJ%LOG  Log in the new job
  1  CJ%NAM  Set user name and password from argblk
2-3  CJ%ACT  Account code for new job
         0  .CJUCA  Use current account of caller
         1  .CJUAA  Use account from the argblk
         2  .CJUDA  Use default account of caller

26
TOPS-20 Monitor Calls Quick Reference Guide
CRJOB

4 CJKETF Place TOPS-20 command processor in top process of new job
5 CJKFLIL Move the file pointed to by word .CUFLIL of the argblk into a process in new job
6 CJKACCS Load ACs from the address in argument block; loaded only if the program being run is not the command processor
7 CJKOWN Maintain ownership of the new job
8 CJKWTM Do not start new job until it is attached to a terminal
9 CJKNPW Do not check password given when new job is logged in
10 CJKNUD Do not update LOGIN date for user logging in to new job
11 CJKSPJ Set primary I/O designators from argblk before starting job
12 CJKCAP Set allowed capabilities of new job to be same as caller's currently enabled capabilities, until new job is logged in
13 CJKCAM Set allowed capabilities of new job to combination and function of capability mask in argblk and new job's user capabilities
14 CJKSLD Send IPCF message to PID supplied in argblk when new job is logged out
17 CJKDSN Release ownership of previously created job whose number is in AC3; if on, overrides all other bits set in AC1

ARGUMENT BLOCK

Word Symbol Meaning
0 .CJNAM Byte pointer to ASCIZ user name string
1 .CUFLS Byte pointer to ASCIZ password string
2 .CUFLT 5B2 + account number or byte pointer to account string
3 .CUFLS Byte pointer to name of file to be moved into a process of new job
4 .CJFSV Offset in entry vector to use as start address of the file to which word .CUFLIL points
5 .CUTTY TTY designator of new job's controlling terminal
6 .CUJTM Connect-time for new job before LGDUT is forced on it; 0 indicates no limit
7 .CUJACS Address of 16-word block to be loaded in new job's ACs if program other than Command Processor is being run
10 .CJXEF Flag bits to be passed to Command Processor in top-level process of new job
10 B0 Suppress herald printed by Command Processor
10 B1 Move file to which word .CUFLIL points into process whose handle is in PRARG block
10 B2 Start process at offset in entry vector given in word .CJFSV after Command
Processor is initialized

B3 Output text printed when LOGIN command is given

11 .CUPRI Primary input and output device designators for the inferior processes of the new job

12 .CFCPU Run-time limit for new job

13 .CUCAM Capability mask for new job; used only if CUCAM is set

14 .CUSLO PID to which IPCF message is to be sent when new job is logged out

Format of IPCF Logout Message

Word Contents
0 0,..,IPDLO
1 <count of remaining words>,<# of job logged out>
2 Flags,.reserved
   0 SP%BAT Job is controlled by batch
   1 SP%DFS Spooling is deferred
   2 SP%LLO Job executed LGOUT
   3 SP%LO Job was forced to logout
   4 SP%LO Job was logged out by another job
3 Job connect time
4 Job CPU time
5 TTY number of job at logout (-1 if detached)
6 Job # of job that did logout
7 Reserved
10 Most recent monitor call error code

CRLNM  JSYS 502

FUNCTION
Defines or deletes a logical name assignment.

CALLING SEQUENCE
AC1: Function code
AC2: Byte pointer to the logical name
AC3: Byte pointer to the logical name definition string

RETURNS  +1: Failure, error code in AC1
          +2: Success, updated byte pointer in AC3

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.CLNJ1</td>
<td>Delete one logical name from the job</td>
</tr>
<tr>
<td>1</td>
<td>.CLNS1</td>
<td>Delete one logical name from the system</td>
</tr>
<tr>
<td>2</td>
<td>.CLNJ2</td>
<td>Delete all logical names from the job</td>
</tr>
<tr>
<td>3</td>
<td>.CLNS2</td>
<td>Delete all logical names from the system</td>
</tr>
<tr>
<td>4</td>
<td>.CLNJ3</td>
<td>Create a logical name for the job</td>
</tr>
<tr>
<td>5</td>
<td>.CLNSY</td>
<td>Create a logical name for the system</td>
</tr>
</tbody>
</table>
CVHST JSYS 276

FUNCTION
Converts a host number to a primary name.

RESTRICTIONS
For use with ARPANET systems only.

CALLING SEQUENCE
AC1: Destination for ASCIZ host name string
AC2: Host number

RETURNS
+1: Failure
+2: Success

CVSKT JSYS 275

FUNCTION
Converts a local socket number to absolute form.

RESTRICTIONS
For use with ARPANET systems only.

CALLING SEQUENCE
AC1: JFN

RETURNS
+1: Failure, error code in AC1
+2: Success, absolute socket number in AC2

DEBRK JSYS 136

FUNCTION
Dismisses the software interrupt routine in progress and
resumes the process at the location specified by the PC
stored in the priority level table.

RETURNS
+1: Only if no software interrupt is currently in
progress and if an ERUMP or ERCAL instruction
follows the DEBRK

DELDIF JSYS 67

FUNCTION
Reclaims disk space by expunging deleted disk files.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some
functions.
CALLING SEQUENCE
AC1: B0(DF%NRJ) Delete temporary files (;T) also
    B1(DF%DNF) Delete nonexistent files that are not now open
    B2(DF%RST) Rebuild the symbol table
    B3(DF%CHK) Check internal consistency of directory
AC2: Directory number

RETURNS +1: Always

DELF   JSYS 26

FUNCTION
Deletes specified disk file and, if the file is closed, releases the JFN.

REstrictions
Requires enabled WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE
AC1: B0(DF%NRJ) Do not release the JFN
    B1(DF%EXP) Expunge file and delete FDB entry in directory
    B2(DF%FGT) Expunge file but do not deassign its addresses; (enabled WHL/DPR)
    B3(DF%DIR) Delete and expunge a directory file;
                (enabled WHL/DPR)
    B4(DF%ARC) Allow a file with archive status to be deleted
    B5(DF%CN0) Delete and expunge file but preserve filename and FDB (except for page count and page table address)
B18-35(DF%JFN) JFN of the file being deleted

RETURNS +1: Failure, error code in AC1
           +2: Success, JFN is released unless B0(DF%NRJ) is on or file is open

DELF   JSYS 317

FUNCTION
Marks for deletion all but the specified number of generations of a disk file.

CALLING SEQUENCE
AC1: B0(DF%NRJ) Do not release the JFN
    B4(DF%ARC) Allow a file with archive status to be deleted
    B5(DF%CNO) Delete and expunge file but preserve filename and FDB (except for page count and page table address)
B18-35    JFN of the file being deleted
AC2:  Number of generations to retain

RETURNS  +1: Failure, error code in AC1
         +2: Success, with number of files deleted in AC2

DEQ   USYS 514

FUNCTION
Removes a request for a specific resource from the queue
associated with that resource.

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1:  Function code
AC2:  Address of argblk

RETURNS  +1: Failure, error code in AC1
         +2: Success

FUNCTION CODES
Code  Symbol  Meaning
  0   .DEQDR  Remove specified requests from queue;
               requires argblk
  1   .DEQDA  Remove all requests for this process from
               queue
  2   .DEQID  Remove all requests corresponding to
               specified request ID

ARGUMENT BLOCK FOR FUNCTION .DEQDR

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.ENQLN</td>
<td>BO-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B6-17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35</td>
</tr>
<tr>
<td>1</td>
<td>.ENQID</td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td>2</td>
<td>.ENQLV</td>
<td>&lt;flags &amp; level number&gt;, [JFN][-1][-2][-3]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO(EN%SHR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(EN%BLN)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(EN%NST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B3(EN%LTL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B9-17(EN%LVL)</td>
</tr>
</tbody>
</table>
|      |        | B18-35  | JFN | Associated file has standard protection; or
|      |        |        | -1 | Resource accessible only by processes in job; or
|      |        |        | -2 | Resource accessible by any job on system; or

| |
-3 Resource accessible only by enabled WHL/QPR processes

3 .ENUC Pointer to string; or 5B2+33-bit user code
4 .ENQRS <# of resources in pool>,<# requested>; or 0,<group number> if only one resource of specific type
5 .ENOMS Address of a resource mask block
n-4 <flags & level number>,,[JFN][(J-1)][-2][-3]

n-3 Pointer to string; or 5B2+33-bit user code
n-2 <# of resources in pool>,,<# requested>
or 0,<group number>
n-1 Address of a resource mask block

DEVST USYS 121

FUNCTION
Translates the given device designator to its corresponding ASCIZ device name string (excluding colon).

CALLING SEQUENCE
AC1: Destination designator
AC2: Device designator

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC1, if pertinent

DFIN USYS 234

FUNCTION
Inputs a double-precision, floating-point number, rounding if necessary.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: Failure, error code in AC4 and updated string pointer in AC1, if pertinent
+2: Success, double-precision, floating-point number in AC2 and AC3 and updated byte pointer in AC1, if pertinent

DFOUT USYS 235

FUNCTION
Outputs a double-precision, floating-point number.

CALLING SEQUENCE
AC1: Destination designator
AC2: 1st word of a normalized, double-precision.
floating-point number
AC3: 2nd word of a normalized, double-precision, floating-point number
AC4: Format control word

RETURNS +1: Failure, error code in AC1 and updated string pointer in AC1, if pertinent.
+2: Success, updated byte pointer in AC1, if pertinent

DIAG  JSYS B30

FUNCTION
Reserves/releases a channel and either a single device or all devices attached to that channel.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: ,<length of argblk>,<address of argblk>

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Argblk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.DGACU</td>
<td>Assign channel and a single device; release device after time limit specified</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Device address</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Time limit in mss</td>
</tr>
<tr>
<td>2</td>
<td>.DGACH</td>
<td>Assign the channel and all devices</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Device address</td>
</tr>
<tr>
<td>3</td>
<td>.DGRCH</td>
<td>Release channel and all assigned devices</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Device address</td>
</tr>
<tr>
<td>4</td>
<td>.DGSCP</td>
<td>Set up channel program</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Device address</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Channel control word 0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Channel control word 1</td>
</tr>
<tr>
<td></td>
<td>n+2</td>
<td>Channel control word n</td>
</tr>
<tr>
<td>5</td>
<td>.DGRCP</td>
<td>Release channel program</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Device address</td>
</tr>
<tr>
<td>6</td>
<td>.DGGCS</td>
<td>Return status of channel</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Device address</td>
</tr>
<tr>
<td></td>
<td>2-5</td>
<td>4-word channel logout area</td>
</tr>
<tr>
<td>100</td>
<td>.DGGEM</td>
<td>Get memory (for TGHA)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>Function code</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1st page in user address space</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

DIAG

2  1st physical memory page
3  Number of pages
4  User address of AR/ARX parity trap routines

101 .DGREM Release memory (for TGHA)
0  Function code

102 .DGPDL Inform the monitor that a device previously unknown is now online
0  Function code
1  Primary channel number
2  Primary unit number
3  Primary controller number; -1 if no controller
4  Alternate channel number
5  Alternate unit number
6  Alternate controller number; -1 if no controller

DEVICE ADDRESS WORD
0 2 3 9 10 23 24 29 30 35

<table>
<thead>
<tr>
<th>address type</th>
<th>device code</th>
<th>0</th>
<th>unit</th>
<th>subunit</th>
</tr>
</thead>
</table>

DIBE  JSYS 212

FUNCTION
Dismisses the process until the designated file input buffer is empty.

CALLING SEQUENCE
AC1: File designator

RETURNS +1: Always

DIC  JSYS 133

FUNCTION
Deactivates the specified software interrupt channels.

AC1: Process handle
AC2: 36-bit word (15n means deactivate channel n)

RETURNS +1: Always

34
**DIR JSYS 130**

**FUNCTION**
Disables the software interrupt system for a process.

**CALLING SEQUENCE**
AC1: Process handle

**RETURNS** +1: Always

---

**DIRST JSYS 41**

**FUNCTION**
Translates the specified 36-bit user or directory number to its corresponding string and writes it to the given destination.

**CALLING SEQUENCE**
AC1: Destination designator
AC2: User or directory number

**RETURNS** +1: Failure, with error code in AC1.
+2: Success, string written to destination, updated string pointer, if pertinent, in AC1

---

**DISMS JSYS 167**

**FUNCTION**
Dismisses this process for the specified amount of time.

**CALLING SEQUENCE**
AC1: Number of mss. for which the process is to be dismissed

**RETURNS** +1: When the elapsed time is up

---

**DDBE JSYS 104**

**FUNCTION**
Dismisses the process until the designated file output buffer is empty.

**CALLING SEQUENCE**
AC1: Destination designator

**RETURNS** +1: Always
TOPS-20 Monitor Calls Quick Reference Guide

DSKAS

DSKAS  JSYS 244

FUNCTION
Assigns or deassigns specific disk addresses.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1:  B0(DA%DEA)
       B1(DA%ASF)
       B2(DA%CNV)
       B3(DA%HWA)
       B4(DA%INI)
       B5(DA%WRT)
       B18-35(DA%ADR) Disk address
AC2:  Device designator of structure; not required if DA%CNV is on in AC1

RETURNS  +1:  Failure, address already assigned or cannot be assigned
         +2:  Success, address assigned in AC1

DSKOP  JSYS 242

FUNCTION
Allows the process to reference physical disk addresses when performing disk transfers.

RESTRICTIONS
Requires WHEEL, OPERATOR, or MAINTENANCE capabilities enabled.

CALLING SEQUENCE
AC1:  B0-1(DOP%AT)
       B2-6(DOP%CN)
       B7-12(DOP%UN)
       B13-35(DOP%UA) unit address
       2(,DOPSR) for structure or relative addresses, with
       B2-10(DOP%SN) structure designator
       0 PS:  -1 structure designator in AC4
       B11-35(DOP%RA) relative address
AC2:  <control flags>,<number of words to transfer>
       B9(DOP%NF)  Use channel/controller/unit #s in AC4
       B10(DOP%ED) Error if unit off-line
       B11(DOP%IL) Inhibit error logging
B12(DOP%IR) Inhibit error recovery
B14(DOP%WR) Write data to disk; if off, read data from disk
B18-35(DOP%CT) Word count

AC3: Address in caller's address space from which data is read or into which data is written
AC4: Device designator of structure if -1 in DDP%SN; physical channel, controller, and unit numbers if 1B9(DOP%NF) with
B0-11(DOP%C2) Channel number
B12-23(DOP%K2) Controller number
B13-35(DOP%U2) Unit number

RETURNS +1: Always, AC1 is nonzero if an error occurred, or zero if no error occurred.

DTACH Jsys 115

FUNCTION
Detaches the controlling terminal from the current job.

RETURNS +1: Always

DTI Jsys 140

FUNCTION
Deassigns a terminal interrupt code.

CALLING SEQUENCE
AC1: Terminal interrupt code

RETURNS +1: Always

DUMPI Jsys 66

FUNCTION
Reads data words into memory in unbuffered data mode.

RESTRICTIONS
File must be open for data mode 17

CALLING SEQUENCE
AC1: JFN
AC2: B0(DMP%NWT) Do not wait for completion of requested operation
B10-35(DMP%PTR) Address of command list in memory

RETURNS +1: Failure, error code in AC1, pointer to bad command in AC2
+2: Success, pointer in AC2 updated to last
TOPS-20 Monitor Calls Quick Reference Guide
DUMPI

command

COMMAND LIST FORMAT

Entry   Meaning

IOWD  n, loc  Causes n words to be transferred from file to locations loc through loc+n-1 of process address space.

XWD  0, y    Causes next command to be taken from location y.

O       Terminates the command list.

DUMPO  USYS 66

FUNCTION

Writes data words from memory in unbuffered data mode.

RESTRICTIONS

File must be open for data mode 17.

CALLING SEQUENCE

AC1: JFN
AC2: BO(DM%AWT)       Do not wait for completion of requested operation

B18-35(DM%PTR)       Address of command list in memory

RETURNS

+1: Failure, error code in AC1, pointer to bad command in AC2.
+2: Success, pointer in AC2 updated to last command.

COMMAND LIST FORMAT

Entry   Meaning

IOWD  n, loc  Causes n words to be transferred from file to locations loc through loc+n-1 of process address space.

XWD  0, y    Causes next command to be taken from location y.

O       Terminates command list.

DVCHR  USYS 117

FUNCTION

Returns the characteristics of the specified device.

CALLING SEQUENCE

AC1: JFN or device designator

RETURNS

+1: Always, with

AC1: Device designator (even if JFN given)
AC2: Device characteristics word
AC3: <job # to which assigned>,<unit #>

<job #>,-1 if no units

-1,<[unit #][1]> if not assigned

-2,<[unit #][1]> if assigned to
DEVICE CHARACTERISTICS WORD

<table>
<thead>
<tr>
<th>Bit</th>
<th>DV%O</th>
<th>DV%I</th>
<th>DV%D</th>
<th>DV%A</th>
<th>DV%S</th>
<th>DV%M</th>
<th>DV%N</th>
<th>DV%T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OUT</td>
<td>IN</td>
<td>DIR</td>
<td>AV</td>
<td>AS</td>
<td>MNT</td>
<td>TYP</td>
<td>MOD</td>
</tr>
<tr>
<td>1</td>
<td>Device can do output</td>
<td>Device can do input</td>
<td>Device has a directory</td>
<td>Device is available or assigned to this job</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Device is assignable with ASND</td>
<td>Device has multiple directories</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Device type

| 0 | .DVDSK | Disk |
| 1 | .DVMTA | Magtape |
| 2 | .DVLPT | Line printer |
| 3 | .DVCOR | Card reader |
| 4 | .DVFE | Front-end pseudo-device |
| 5 | .DVTY | Terminal |
| 6 | .DVTPY | Pseudo-terminal |
| 7 | .DVNUL | Null device |
| 8 | .DVNET | ARPA network |
| 9 | .DVDCN | DECnet active component |
| 10| .DVSRC | DECnet passive component |

Data mode in which device can be opened

<table>
<thead>
<tr>
<th>20-35</th>
<th>DV%MOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>DV%M17</td>
</tr>
<tr>
<td>27</td>
<td>DV%M10</td>
</tr>
<tr>
<td>34</td>
<td>DV%M1</td>
</tr>
<tr>
<td>35</td>
<td>DV%M0</td>
</tr>
</tbody>
</table>

EIR USYS 126

FUNCTION
Enables the software interrupt system for a process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

ENO USYS 513

FUNCTION
Requests access to a specific resource by placing a request in the queue for that resource.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions. In non-zero sections, OWGBPs must specify 7-bit bytes.
CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Failure, error code in AC1
         +2: Success

FUNCTION CODES
Code  Symbol  Meaning
0     .ENQBL  Queue requests and block process until all
           requested locks are acquired
1     .ENQA   Queue requests and acquire locks only if all
           requested resources are immediately available
2     .ENQSI  Queue requests
3     .ENQMA  Modify access of a previously queued request

ARGUMENT BLOCK
Word  Symbol  Meaning
0     .ENQLN  BO-5  Header length
              B6-17  # of locks
              B18-35 Length of argblk
1     .ENQID  <PSI channel number>,<request ID>
2     .ENQLV  <flags & level number>,<JFN>[[-1][-2][-3]>
             BO(ENKSHR) Access to this resource is to
             be shared
             B1(EN%BLN) Ignore level number of
             resource
             B2(EN%NST) Allow ownership of this lock
             to be nested
             B3(ENKTL)  Allow a long-term lock on this
             resource
             B9-17(EN%LVL) Level number associated with
             this resource
             B18-35  JFN Associated file has standard
             protection
                       -1  Resource accessible only by
                       processes in job
                       -2  Resource accessible by any job
                       on system
                       -3  Resource accessible only by
                       enabled WHL/DPR processes
3     .ENQUC  Pointer to string or a 582+33-bit user code
4     .ENQRS  <# of resources in pool>,<# requested>
             or 0,<group number> if only one resource of
             type exists
5     .ENQMS  Address of a resource mask block
n-4  <flags & level number>,<JFN>[[-1][-2][-3]>
n-3  Pointer to string or 582+33-bit user code
n-2  <# of resources in pool>,<# requested>
     or 0,<group number>
40
ENQC  SYS  515

FUNCTION
Returns the current status of the given resource and obtains information about the state of the queues.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability for some functions. In non-zero sections, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk
AC3: Address of block for status information (.ENQCS only)

RETURNS  +1: Failure, error code in AC1
          +2: Success

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.ENQCS</td>
<td>Returns status of specified resources</td>
</tr>
<tr>
<td>1</td>
<td>.ENQCG</td>
<td>Return ENQ/DEQ quota for specified job</td>
</tr>
<tr>
<td>2</td>
<td>.ENQCC</td>
<td>Change ENQ/DEQ quota for specified job (enabled WHL)</td>
</tr>
<tr>
<td>3</td>
<td>.ENQCD</td>
<td>Dump ENQ/DEQ locks and queue entries into argblk (enabled WHL)</td>
</tr>
</tbody>
</table>

STATUS BLOCK

<table>
<thead>
<tr>
<th>Word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Resource status flags</td>
</tr>
<tr>
<td>1</td>
<td>Process owns the lock</td>
</tr>
<tr>
<td>2</td>
<td>Process is in queue waiting for this resource</td>
</tr>
<tr>
<td>3</td>
<td>Lock has been allocated for exclusive access</td>
</tr>
<tr>
<td>4</td>
<td>Process is in queue waiting for exclusive access to resource</td>
</tr>
<tr>
<td>5-17</td>
<td>Level number of the resource</td>
</tr>
<tr>
<td>18-35</td>
<td>36-bit time stamp</td>
</tr>
<tr>
<td>2</td>
<td>Job # of lock owner</td>
</tr>
<tr>
<td>1-16</td>
<td># of processes with lock&gt;.,&lt;request ID</td>
</tr>
</tbody>
</table>

ARGUMENT BLOCK

<table>
<thead>
<tr>
<th>Function</th>
<th>Word</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ENQCS</td>
<td>0</td>
<td>See ENQ SYS for argblk</td>
</tr>
<tr>
<td>.ENQCG</td>
<td>0</td>
<td>&lt;ignored&gt;,&lt;job #&gt;</td>
</tr>
<tr>
<td>.ENQCC</td>
<td>0</td>
<td>&lt;new quota&gt;,&lt;job #&gt;</td>
</tr>
<tr>
<td>.ENQCD</td>
<td>n</td>
<td>Length of block</td>
</tr>
</tbody>
</table>

41
TOPS-20 Monitor Calls Quick Reference Guide

ENQC

Data Returned by Function .ENQCD

Lock Data

Word Symbol Meaning
0 .ENQDF B0-8 Flags
   B9-17 Level number
   B18-35 Ofn, 40000+job #, -2, or -3
1 .ENQDR <total resources in pool>,<# remaining>
   or 0,<group number>
2 .ENQDT Time stamp of last request locked
3 .ENQDC User code of lock or beginning of string

Queue Data

0 .ENQDF B0-8 Flags
   B9-17 PSI channel
   B18-35 job # queue entry creator
1 .ENQDI <group # or number requested>,<request ID>

Flags Returned in Word 0 for Function .ENQCD

Bit Symbol Meaning
B0 .ENQCLI Block contains lock data (if off, queue data)
B1 .ENQCOD Process owns the lock
B2 .ENQCST Lock contains a text string
B3 .ENQCX Lock is for exclusive access
B4 .ENQCOB Process is blocked until exclusive access is available

EPCAP .SYS 151

FUNCTION
Enables the capabilities for the specified process.

CALLING SEQUENCE
AC1: Process handle
AC2: Capabilities the process can enable
AC3: Capabilities to enable

RETURNS +1: Always

ERSTR .SYS 11

FUNCTION
Translates a TOPS-20 error number to its corresponding text string and writes the string to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: <process handle>,<error number>; -1 for most recent
AC3: <-maximum number of bytes to transfer>,0;
     or 0 for no limit

RETURNS +1: Failure, undefined error number
+2: Failure, string size out of bounds or invalid
destination designator
+3: Success

ESOUT JSYS 313

FUNCTION
Outputs an error string.

CALLING SEQUENCE
AC1: Byte pointer to ASCIZ error string

RETURNS +1: Always, with updated byte pointer in AC1

FFFP JSYS 31

FUNCTION
Finds the 1st free page in the specified file.

CALLING SEQUENCE
AC1: <starting page number>,JFN

RETURNS +1: Always, with
AC1: JFN,,<page number>
or -1 if there is no free page

FFORK JSYS 154

FUNCTION
Freezes one or more processes.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

FFUFJ JSYS 211

FUNCTION
Finds the first used page of the file at or beyond the
specified page number.

CALLING SEQUENCE
AC1: JFN,,<starting page number>

RETURNS +1: Failure, error code in AC1
+2: Success, page number in the right half of AC1

43
TOPS-20 Monitor Calls Quick Reference Guide
FLHST

FLHST  SYS 277

FUNCTION
"Flushes" an ARPANET host, causing the NCP tables containing that host's status information to be purged of all information regarding previous partially terminated connections.

RESTRICTIONS
For ARPANET systems only. Requires enabled WHEEL, OPERATOR, or NET WIZARD capability.

CALLING SEQUENCE
AC1: Number of host to be flushed

RETURNS  +1: Always

FLIN  SYS 232

FUNCTION
Inputs a floating-point number from the specified source.

CALLING SEQUENCE
AC1: Source designator

RETURNS  +1: Failure, with
        AC1: Updated byte pointer, if pertinent
        AC3: Error code
        +2: Success, with
        AC1: Updated byte pointer, if pertinent
        AC2: Single-precision, floating-point number

FLOUT  SYS 233

FUNCTION
Outputs a floating-point number to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: Normalized, single-precision, floating-point number
AC3: Format control word

RETURNS  +1: Failure, with
        AC1: Updated byte pointer, if pertinent
        AC3: Error code
        +2: Success, with
        AC1: Updated byte pointer, if pertinent

44
GACCT JSYS 546

FUNCTION
Returns the current account for the specified job.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or CONFIDENTIAL INFORMATION ACCESS capability.

CALLING SEQUENCE
AC1: Job #, or -1 for current job
AC2: Byte pointer to string for alphanumeric account designator (if any)

RETURNS +1: Always, with updated pointer to account string in AC2

GACTF JSYS 37

FUNCTION
Returns the account designator to which the specified file is being charged.

CALLING SEQUENCE
AC1: JFN
AC2: Byte pointer to string for account (if any)

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC2
+3: Success, SB2=account number returned in AC2

GCVEC JSYS 300

FUNCTION
Returns the entry vector and the UU0 locations for the compatibility package.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: B0-17 Entry vector length
     B18-35 Entry vector address
AC2: <UU0 location>,<PC location>
GDSKC  JSYS 214

FUNCTION
Returns information on the given structure's disk usage and availability.

CALLING SEQUENCE
AC1: Device designator (structure) or DSK: for connected structure

RETURNS  +1:  Always, with
          AC1: Number of pages in use
          AC2: Number of pages available

GDSTS  JSYS 145

FUNCTION
Returns the status of a device for user I/O.

CALLING SEQUENCE
AC1: JFN

RETURNS  +1:  Always, with
          AC2: Device-dependent status bits
          AC3: Device-dependant
                For magtape:
                <# of hardware bytes transferred>,0
                For lineprinter:
                last value of page counter register or
                -1 if no page counter register
                For ARPANET network-connection files:
                AC2: Connection state (01-16) in B0-3
                AC3: Foreign host number (octal)
                AC4: Foreign socket number (octal)

GDVEC  JSYS 542

FUNCTION
Returns the entry vector for the Record Management System (RMS).

RESTRICTIONS
Requires RMS software (currently available only with BASIC and COBOL).

CALLING SEQUENCE
AC1: Process handle

RETURNS  +1:  Always, with
          AC2: B0-17  Entry vector length
          B18-35  Entry vector address

46
GET  JSYS 200

FUNCTION
Gets a save file, copying or mapping it into the process as appropriate, and updates the monitor's data base for the process by copying the entry vector and the list of program data vector addresses (PDVA's) from the save file.

RESTRICTIONS
Some functions require WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1:  B0-17 Process handle
       B19(GT%ADR) Use memory address limits given in AC2
       B20(GT%PRL) Preload pages being mapped
       B21(GT%NDV) Do not overlay existing pages; return error
       B22(GT%AAR) If on, AC2 contains address of argblk
       B24-35(GT%JFN) JFN of the save file
AC2:  <lowest process page #>,<highest process page #>
or address of argblk

RETURNS  +1: Always

ARGUMENT BLOCK

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.GFLAG</td>
<td>Flags for remainder of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 GT%LOW .GLOW contains lowest page number in process to use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 GT%GHG .GHIGH contains highest page number in process to use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 GT%BAS .GBASE contains the section number to use</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 GT%CCCH Clear system's program cache (WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 GT%CSH Place in cache the program name being loaded into memory (WHL/OPR)</td>
</tr>
</tbody>
</table>

| 1    | .GLOW  | Lowest process page number into which file page gets loaded |
| 2    | .GHIGH | Highest process page number into which file page gets loaded |
| 3    | .GBASE | Section number into which file pages are loaded (single-section save files only) |

GETAB  JSYS 10

FUNCTION
Returns a word from the specified system table.

RESTRICTIONS
Requires GETAB capability (bit SC%GTB in process capability word).
TOPS-20 Monitor Calls Quick Reference Guide

GETAB

CALLING SEQUENCE
AC1: <index into table>, <table number>

RETURNS
  +1: Failure, error code in AC1
  +2: Success, 36-bit word from table in AC1

GETER  JSYS 12

FUNCTION
Returns the most recent error condition encountered in a process.

CALLING SEQUENCE
AC1: Process handle

RETURNS
  +1: Always, with
      AC2: <process handle>, <most recent error>

GETJ  JSYS 507

FUNCTION
Obtains information about the specified job.

CALLING SEQUENCE
AC1: Job #: -1 for current job; or 400000+TTY number
AC2: <length of destination block>, <address of block>
AC3: Offset of 1st entry desired from job information table

RETURNS
  +1: Failure, error code in AC1
      +2: Success, with updated pointer in AC2 and requested entries stored in specified block

JOB INFORMATION TABLE

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.JNJO</td>
<td>Job #</td>
</tr>
<tr>
<td>1</td>
<td>.JITNO</td>
<td>Job's terminal number; -1 if detached</td>
</tr>
<tr>
<td>2</td>
<td>.JUNO</td>
<td>Job's user number</td>
</tr>
<tr>
<td>3</td>
<td>.JIDNO</td>
<td>Job's connected directory number</td>
</tr>
<tr>
<td>4</td>
<td>.JISNM</td>
<td>Subsystem name (SIXBIT)</td>
</tr>
<tr>
<td>5</td>
<td>.JIPNM</td>
<td>Program name (SIXBIT)</td>
</tr>
<tr>
<td>6</td>
<td>.JIRT</td>
<td>Runtime (in mss.)</td>
</tr>
<tr>
<td>7</td>
<td>.JCPJ</td>
<td>Controlling PTY job #: -1 if no PTY</td>
</tr>
<tr>
<td>10</td>
<td>.JRTL</td>
<td>Runtime limit; -1 if no time limit</td>
</tr>
<tr>
<td>11</td>
<td>.JBAT</td>
<td>If -1, job is controlled by batch</td>
</tr>
<tr>
<td>12</td>
<td>.JIDEN</td>
<td>Default magtape density</td>
</tr>
<tr>
<td>13</td>
<td>.JPAR</td>
<td>Default magtape parity</td>
</tr>
<tr>
<td>14</td>
<td>.JDM</td>
<td>Default magtape data mode</td>
</tr>
<tr>
<td>15</td>
<td>.JRS</td>
<td>Default magtape record size in bytes</td>
</tr>
<tr>
<td>16</td>
<td>.JDFS</td>
<td>If 1, deferred spooling in effect</td>
</tr>
<tr>
<td>17</td>
<td>.JILNO</td>
<td>Job's logged-in directory number</td>
</tr>
<tr>
<td>20</td>
<td>.JISRM</td>
<td>Byte pointer to destination for job's session remark</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

GETJUI

21  .JILLN  Date and time of user's last login before the current job
22  .JISRT  Job CPU time at start of last session
23  .JISCT  Console time at start of last session
24  .JIT20  0 if job is at EXEC level; -1 if at program level
25  .JISTM  Time when job was created; -1 if system time/date not set when job created
26  .JIBCH  Batch stream number and batch flags
            BO-1  OB%WTO  Write-to-operator capabilities
            0  .OBALL  WTO and WTDOR
            1  .OBWR  No WTDOR allowed
            2  .OBDM  No message allowed
            B10  OB%CSS  OB%BSN contains batch-stream #
            B11-17  OB%BSN  Batch-stream #
27  .JILLO  Logical location (node name)

GETNM  JSYS 177

FUNCTION
Returns the name of the program currently being used by the job.

RETURNS  +1: Always, with SIXBIT program name in AC1

GETOK%  JSYS 574

FUNCTION
Requests access to the specified system resource from the installation's access-control program.

CALLING SEQUENCE
AC1:  Function code
AC2:  Address of argblk
AC3:  Length of the argblk
AC4:  Job # or user number request is for

RETURNS  +1: Always, with
            0 in Word 0 of status block if access granted
            1B18+error number in Word 0 of status block
            if request denied

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Argblk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.GOASD</td>
<td>Assign a device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .GEERB Error block address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .GEADD Device designator</td>
</tr>
<tr>
<td>2</td>
<td>.GDCAP</td>
<td>Enable capabilities (right half privileges only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .GEERB Error block address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .GENCP New capability word</td>
</tr>
<tr>
<td>3</td>
<td>.GOCJB</td>
<td>Allow CRJOB JSYS to be executed</td>
</tr>
</tbody>
</table>
4 .Golog Error block address
   Allow LOGIN
   1 .GEERB Error block address
5 .Gofk Allow CFORK JSYS to be executed
   0 .GEERB Error block address
   1 .GEFCT # of forks already in use by job
6 .Gotbr Allow setting of terminal baud rate
   0 .GEERB Error block address
   1 .GELIN Line number
   2 .GESPD <input speed>, <output speed>
7 .Golgo Inform access-control program of a logout
   0 .GEERB Error block address
   1 .GEUSD Number of pages used
   2 .GEQU Directory quota
   3 .GERLG Job # logging out; -1 if caller
10 .Gdenq Allow setting of ENQ quota
   0 .GEERB Error block address
   1 .GEEQU Desired quota
   2 .GEEUN Job #
11 .Gocrd Allow directory creation
12 .Gosmt Allow MOUNT of structure
   0 .GEERB Error block address
   1 .GESDE Device designator
13 .Gomdd Allow entry to MDDT
14 .Gocl Set scheduler class for a job
   0 .GEERB Error block address
   1 .GEJDB Job #
   2 .GECLS Class desired
15 .Goclo Set scheduler class at login
16 .Gota MT: access request
   0 .GEERB Error block address
   1 .GEACC Access code from HDR1 label
   2 .GEUSN User number
   3 .GEUNT MT: unit number
   4 .GEOCD Desired access bits (FPXxxx)
   5 .GELTP Label type (.LTxxx)
17 .Goad Error block address
   0 .GOCO Flags from ACCES JSYS
   2 .GDC Directory number
20 .Godea Allow device assignment due to OPENF
   0 .GEERB Error block address
   1 .GEADD Device designator
21 .Godna Allow DECNET access
22 .Gona Allow ARPANET access
   0 .GEERB Error block address
23 .Gatja Allow ATTACH
   0 .GOTJB Target job #
   1 .GEADD Source TTY number
400000+n Customer-reserved functions
ERROR BLOCK FORMAT (RET)

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.GESIZ</td>
<td>Count of words in block (including this word)</td>
</tr>
<tr>
<td>1</td>
<td>.GEERN</td>
<td>Error number</td>
</tr>
<tr>
<td>2</td>
<td>.GEPTR</td>
<td>Byte pointer to error string location</td>
</tr>
<tr>
<td>3</td>
<td>.GEBSZ</td>
<td>Maximum bytes user can accept in error string</td>
</tr>
</tbody>
</table>

FUNCTION

Returns the section-relative entry vector of the specified process.

RESTRICTIONS

Process must run in a single section of memory.

CALLING SEQUENCE

AC1: Process handle

RETURNS +1: Always, with entry vector word in AC2

FUNCTION

Gets a handle on a process that currently is not known to the calling process but is known to another process.

CALLING SEQUENCE

AC1: Handle of process that has handle on desired process
AC2: Process handle relative to process in AC1 that refers to desired process

RETURNS +1: Failure, with error code in AC1 +2: Success, with AC1: Relative handle of the desired process

FUNCTION

Returns the process structure of the current job from a given process downward.

RESTRICTIONS

Requires WHEEL or OPERATOR capability for some functions.

CALLING SEQUENCE

AC1: Process handle of the starting point
AC2: BO(GF%GH) Return relative process handles for each process
     B1(GF%GS) Return status for each process
AC3: <word count in PSB>,<address of PSB>

RETURNS +1: Failure, error code in AC1
         +2: Success, all process handles are returned

GFUST JSYS 550

FUNCTION
Returns the name of either the author of the file or the
user who last wrote to the file.

CALLING SEQUENCE
AC1: <function code>,JFN
AC2: Pointer to author/user string

RETURNS +1: Always, with an updated byte pointer in AC2

FUNCTION CODES
Code   Symbol  Meaning
0       .GFAUT  Return name of author of file
1       .GFLWR  Return name of user who last wrote to file

GIVOK% JSYS 576

FUNCTION
Allows a privileged access-control program to permit or
refuse a user program’s access to a specified system
resource.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Request number (from RCVOK% message)
AC2: 0 to permit request
     1B18 + error number to refuse request
AC3: Pointer to ASCIZ message string (80 characters
      maximum); or 0

RETURNS +1: Always

GUINF JSYS 13

FUNCTION
Returns information pertaining to the current job.

RETURNS +1: Always, with
AC1: User number under which job is running
AC2: Directory number to which job is connected

52
AC3: Job #
AC4: TTY # attached to job; or -1 if none

**GNJFN JSYS 17**

**FUNCTION**
Assigns the JFN to the next file in a group of files that have been specified with wildcard characters.

**CALLING SEQUENCE**
AC1: Indexable file handle returned by GTJFN (flags, JFN)

**RETURNS**
+1: Failure; occurs on 1st call to GNJFN with no flags indicating wildcard fields on in B18-35 of AC1 (JFN released if no more files in group)
+2: Success, same JFN is assigned to next file in group, with
AC1: B13 GN%STR Structure changed
     B14 GN%DIR Directory changed
     B15 GN%NAM Name changed
     B16 GN%EXT File type changed

**GPJFN JSYS 206**

**FUNCTION**
Returns the primary JFNs of the specified process.

**CALLING SEQUENCE**
AC1: Process handle

**RETURNS**
+1: Always, with
AC2: BO-17 Primary input JFN
     B18-35 Primary output JFN

**GTAD JSYS 227**

**FUNCTION**
Returns the current date in the internal system format.

**RETURNS**
+1: Always, with
AC1: Day,,<fraction of day> or -1 if system date not set
**GTDAL**  JES3 305

**FUNCTION**
Returns the disk allocation for the specified directory.

**CALLING SEQUENCE**
AC1: Directory number; -1 for connected directory

**RETURNS**  +1: Always, with
AC1: Working storage limit for directory
AC2: Number of pages being used
AC3: Permanent storage limit for directory

---

**GTDIR**  JES3 241

**FUNCTION**
Returns information about the given directory.

**RESTRICTIONS**
Requires enabled WHEEL or OPERATOR capability.

**CALLING SEQUENCE**
AC1: Directory number; or 0 for system default settings
AC2: Address of block to store directory information
AC3: Byte pointer to password string

**RETURNS**  +1: Always, with updated byte pointer in AC3

**ARGUMENT BLOCK (RET)**

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.CDLEN</td>
<td>Length of argblk; defaults to 15</td>
</tr>
<tr>
<td>1</td>
<td>.CDSW</td>
<td>Byte pointer to password string</td>
</tr>
<tr>
<td>2</td>
<td>.CDLIQ</td>
<td>Working disk storage quota</td>
</tr>
<tr>
<td>3</td>
<td>.CDPRV</td>
<td>Capabilities for this user</td>
</tr>
<tr>
<td>4</td>
<td>.CDMD</td>
<td>Mode word</td>
</tr>
<tr>
<td>5</td>
<td>.CDLOQ</td>
<td>Permanent disk storage quota</td>
</tr>
<tr>
<td>6</td>
<td>.CDNUM</td>
<td>Directory number</td>
</tr>
<tr>
<td>7</td>
<td>.CDFTP</td>
<td>Default file protection (18 bits, R-J)</td>
</tr>
<tr>
<td>10</td>
<td>.CDGPT</td>
<td>Directory protection (18 bits, R-J)</td>
</tr>
<tr>
<td>11</td>
<td>.CDRET</td>
<td>Default generation retention count</td>
</tr>
<tr>
<td>12</td>
<td>.CDLD</td>
<td>Date of last login</td>
</tr>
<tr>
<td>13</td>
<td>.CDUGP</td>
<td>Address of user group list for this directory</td>
</tr>
<tr>
<td>14</td>
<td>.CDGDP</td>
<td>Address of directory group list</td>
</tr>
<tr>
<td>15</td>
<td>.CDDQ</td>
<td>Maximum number of sub-directories allowed</td>
</tr>
<tr>
<td>16</td>
<td>.CDCUG</td>
<td>Address of user group list</td>
</tr>
<tr>
<td>17</td>
<td>.CDDAC</td>
<td>0</td>
</tr>
</tbody>
</table>

54
20 .CDDNE  Default on-line expiration date and time
21 .CDDFE  Default off-line expiration date and time

GTFDB  JSYS 63

FUNCTION
Returns some or all of the file descriptor block (FDB) for
the specified file.

CALLING SEQUENCE
AC1:  JFN
AC2:  <# of FDB words to read>,<offset of 1st word desired>
AC3:  Address of block for returned data

RETURNS  +1: Always

GTHST  JSYS 273

FUNCTION
Obtains information about ARPANET hosts.

RESTRICTIONS
For ARPANET systems only

CALLING SEQUENCE
AC1:  Function code
AC2:  Function-specific argument
AC3:  Function-specific argument
AC4:  Function-specific argument

RETURNS  +1: Failure, error code in AC1
          +2: Success, data returned in ACs

FUNCTION CODES
Code  Symbol  Function
0 .GTHS2  Returns host table sizes
   AC2:  <number host names>,0
   AC3:  <length of HSTSTS table>,0
   AC4:  Local host number (in 32-bit Internet
         format)
1 .GTHIX  Returns name string associated with host
         Arguments
   AC2:  Byte pointer to destination for name
         string
   AC3:  Index into name table (returned by
         GETAB)
         Returns
   AC2:  Updated byte pointer
   AC3:  Host number
   AC4:  Host status; if name is a nickname, 
         HS%NCK is on

55
TOPS-20 Monitor Calls Quick Reference Guide
GTHST%

2 .GTHNS Returns primary name for given host number
   Arguments
   AC2: Byte pointer to destination for primary
   name
   AC3: Host number
   Returns
   AC2: Updated byte pointer
   AC3: Host number
   AC4: Host status

3 .GTHSN Translates specified host name string to its
   host number
   Arguments
   AC2: Byte pointer to host name string
   Returns
   AC2: Updated byte pointer
   AC3: Host number
   AC4: Host status

4 .GTHHN Returns current status of given host
   Arguments
   AC3: Host number
   Returns
   AC4: Host status

5 .GTHHI Returns host number and host status
   Arguments
   AC3: Index into HSTSTS (returned by GETAB)
   Returns
   AC3: Host number
   AC4: Host status

FLAGS IN HOST STATUS WORD
Bits Symbol Meaning
180 HS%UP Host is up
181 HS%VAL Valid status
B2-4 HS%DAY Day when up if currently down
B5-9 HS%HR Hour
B10-13 HS%MIN 5 minute interval
B14-17 HS%RSN Reason
1B18 HS%SRV Host is server
1B19 HS%USR Host is user
1B20 HS%NK Nickname
B21-26 HS%STY System type mask
1B27 HS%NEW RAS, RAR, RAP, etc

System Type Flags (HS%STY)
Bits Symbol Meaning
1B26 .HS1OX TENEX
2B26 .HSITS ITS
3B26 .HSDEC TOPS-10
4B26 .HSTIP TIP
5B26 .HSMTIP MTIP
6B26 .HSELF ELF
7B26 .HSANT ANTS
1OB26 .HSMT MULTICS
11B26 .HST2O TOPS-20
12B26 .HSUX UNIX

56
TOPS-20 Monitor Calls Quick Reference Guide

GTJFN

GTJFN JSYS 20

FUNCTION
Returns a JFN for the specified file. The short form accepts
the filespec from a string in memory or from a file, but not
from both; the long form accepts the filespec from either
memory or a file (if both are provided, the string in memory
is used first).

CALLING SEQUENCE
AC1: Flags, generation (short form)
0, <address of argblk> (long form)
AC2: Source designator from which to obtain filespec (short
form)
Byte pointer to ASCIZ filespec string; or 0 if none
(long form)

RETURNS
+1: Failure, error code in AC1
+2: Success, with
AC1: Flags, JFN
AC2: Updated byte pointer, if pertinent

GTJFN FLAG BITS

Bit Symbol Meaning
0 GU%FDU File is to be assigned next higher generation
1 GU%NEW File must not exist (no effect on a
parse-only JFN)
2 GU%OLD File must exist (no effect on a parse-only
JFN)
3 GU%MSG Print message after filespec if user types
ESC; possible messages:
!NEW FILE!
!NEW GENERATION!
!OLD GENERATION!
!OK!
!CONFIRM!
4 GU%CFM Require confirmation from user (if GU%FNS is
on) to verify filespec
5 GU%TMP File specified is a temporary file
6 GU%NS Search only the 1st specification in a
multiple logical name assignment for file
7 GU%ACC JFN cannot be accessed by inferior processes
8 GU%DEL Consider deleted files when searching for
file
9-10 GU%JFN Associate JFN in word 10 (.GUJFN) of argblk
with filespec according to value (long form
only)
0 (.GJDU) Ignore JFN supplied
2 (.GJERR) Assign JFN supplied; return error
if not available
3 (.GJALT) Assign JFN supplied; assign
alternate if not available
11 GU%IFG Allow use of wildcards in fields of filespec
12 GU%OFG Associate JFN with filespec string only, not
file
13 GU%FLG Return flags in the left half of AC1 if
TOPS-20 Monitor Calls Quick Reference Guide
GTJFN

14 GJ%PHY
Ignore job-wide logical names

15 GJ%XTN
Argblk contains more than 10 words (long form only)

16 GJ%FNS
If on, AC2 contains <input JFN>,<output JFN>
if off, AC2 contains byte pointer to ASCIZ filespec string (short form only)

17 GJ%SHT
Must be on for short form GTJFN; must be off for long form GTJFN

18-35 Generation of file or one of:
0(.GJDEF) Use next higher generation if
180(GJ%FDU); use highest existing
 generation if OBO(GJ%FDU)
-1(.GJN+G) Use next higher generation if
 none supplied
-2(.GUEG) Use lowest existing generation
-3(.GUALL) Use all generations and assign
 JFN to 1st file in group (GJ%FIG
 must be set)

ARGUMENT BLOCK (Long Form Only)
Word Symbol Meaning
0 .GUGEN Flags,<generation>
1 .GUSRC <input JFN>,<output JFN>
2 .GJDEV Byte pointer to ASCIZ default device string;
on 0 for user's connected structure
3 .GJDIR Byte pointer to ASCIZ default directory
 string; or 0 for user's connected directory
4 .GJNAM Byte pointer to ASCIZ default filename
 string; if 0, string in AC2 or input JFN must
 supply filename
5 .GJEXT Byte pointer to ASCIZ default file type
 string; or 0 for null file type
6 .GJPRO Byte pointer to ASCIZ default protection
 string; or 0 for default directory protection
 or protection of next lower generation
7 .GJACT Byte pointer to ASCIZ default account string;
on 0 for user's LOGIN account (unless
 changed)
10 .GUJFN JFN to associate with filespec if GJ%JFN is
 set in word 0 (.GUGEN)
11 .GJF2 Flags,<count of remaining words in block>
   if GJ%XTN is set in word 0 (.GUGEN) (OPT)
   B0(GJ%RND) Return if filename buffer empty
   and user attempts to delete character
   B2(GJ%NLN) Filenames limited to 6
   characters, file types to 3
   characters; generation, temporary
   status, protection, and account
   fields not allowed in string or
   input data
   B3(GJ%RCM) Return confirmation message in
   destination buffer
   B4(GJ%RLE) Return if input buffer empty, and
   user attempts to delete character

58
B5(G1%IIN) Consider invisible files when searching for file.

B6(G1%SLN) Prohibit expansion of logical names.

12 .GUCPP Byte pointer to destination string for copy of user’s typescript.

13 .GUCPC Number of bytes available in destination string; if 0, 130 bytes assumed.

14 .GURTYP Byte pointer to CTRL/R buffer.

15 .GJBF PSObsolete.

16 .GJATR Pointer to filespec attribute block.

ATTRIBUTE BLOCK (Long Form Only)

Word Contents
0 Count of words in block (including this word)
1 Byte pointer to argument string
1+n Byte pointer to argument string

ATTRIBUTE VALUES (Long Form Only)

Keyword Attribute Value
A: Installation-defined account string
BDATA: DECnet binary optional data
BLOCK-LENGTH: Magnetic-tape block length (in bytes)
BPASSWORD: DECnet binary password
CHARGE: DECnet account string
DATA: DECnet optional data
EXPIRATION-DATE: Magnetic-tape expiration date
FORMAT: Magnetic-tape record format
P: Fixed-length records
D Variable-length records
S Spanned records
U Binary files with 36-bit words
OFF-LINE NDONE - display-only keyword
PASSWORD: DECnet password string
POSITION: File sequence number for positioning magnetic-tape
RECORD-LENGTH: Magnetic-tape record length (in bytes)
T NDONE - display-only keyword
USERID: DECnet user ID string

Flags Returned in AC1

Bit Symbol Meaning
0 G1%DEV Device field of filespec contained wildcards
1 G1%UNIT Unit field of filespec contained wildcards
2 G1%DIR Directory field of filespec contained wildcards
3 G1%NAME Filename field of filespec contained wildcards
4 G1%EXT File type field of filespec contained wildcards
5 G1%VER Generation field of filespec contained wildcards
6 G1%UHV File used has highest generation
7 G1%HNV File used has next higher generation
8 G1%UHV File used has lowest generation
TOPS-20 Monitor Calls Quick Reference Guide
GTJFN

9   GU%PRO  Protection field of filespec was given
10  GU%ACT  Account field of filespec was given
11  GU%TFS  Filespec is for temporary file
12  GU%GND  Deleted files were not considered when assigning JFNs
17  GU%INV  Invisible files were not considered when assigning JFNs

GTRPI  JSYS 172

FUNCTION
Returns paging trap information for the specified process.

CALLING SEQUENCE
AC1:  Process handle

RETURNS  +1:  Always, with
AC1:  # of pager traps since process started
AC2:  # of page faults since process started
AC3:  Time spent (in mss) in page routines since process started

GTNCP%  JSYS 272

FUNCTION
Obtains information about the NCP.

RESTRICTIONS
For ARPANET systems only

CALLING SEQUENCE
AC1:  Function code
AC2:  Function-specific argument
AC3:  Function-specific argument
AC4:  Function-specific argument

RETURNS  +1:  Failure, error code in AC1
+2:  Success, data returned in AC's

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.GTNSZ</td>
<td>Returns negative number of NCP connections</td>
<td>AC2:  -&lt;# of NCP connections&gt;,0&lt;br&gt;AC3:  -&lt;# of NVTs&gt;,&lt;line # of 1st NVT&gt;</td>
</tr>
<tr>
<td>1</td>
<td>.GTNIX</td>
<td>Returns status of connection number</td>
<td>Arguments&lt;br&gt;AC2:  Connection number&lt;br&gt;AC3:  30-bit address of data block&lt;br&gt;AC4:  -&lt;block length&gt;,&lt;index of 1st item&gt;</td>
</tr>
</tbody>
</table>

60
2 .GTNNI  Return status of NVT line number (input connection)
   Arguments
   AC2: NVT line number (input)
   AC3: 30-bit address of data block
   AC4: -<block length>,<index of 1st item>
   Returns
   Data in data block
3 .GTNNO  Return status of NVT connection (output connection)
   Arguments
   AC2: NVT line number (output)
   AC3: 30-bit address of data block
   AC4: -<block length>,<index of 1st item>
   Returns
   Data in data block
4 .GTNJF  Return status of network-connection JFN
   Arguments
   AC2: JFN
   AC3: 30-bit address of data block
   AC4: -<block length>,<index of 1st item>
   Returns
   Data in data block

FORMAT OF RETURNED DATA BLOCK
Word Symbol Contents
0 .NCIDX NCP connection index
1 .NCFHS Foreign host
2 .NCLSK Local socket
3 .NCFSK Foreign socket
4 .NCFSM State of connection
5 .NCLNK Link
6 .NCNVT NVT, or -1 if none
7 .NCSIZ Byte size of connection
10 .NCMSG Message allocation
11 .NCBAL Bit allocation
12 .NCDAL Desired allocation
13 .NCBTC Bits transferred
14 .NGBPB Bytes per buffer
15 .NCCLK Time-out countdown
16 .NCSTS Connection status

GTRPW JSYS 171

FUNCTION
Returns the trap words.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
   AC1: Status word from last memory trap or 0
      if no traps
   AC2: Last monitor call that had an error

61
STATUS WORD

Bit          Meaning
BO(PF%USR)  Page failure - user mode reference
BS(PF%WR)   Page failure - write reference
B4(TSW%RD)  Trap status - read (always on)
B5(TSW%WT)  Trap status - write (same setting as BS)
B6(TSW%EX)  Trap status - execute (always on)
B7(TSW%MN)  Trap status - monitor mode reference
B18-B35     Address of reference that caused trap

FUNCTION

GTSTS  JSYS 24

Returns the status of a file associated with a JFN.

CALLING SEQUENCE

AC1:  0, JFN

RETURNS  +1: Always, with
          AC2:  status: 0B10, if JFN illegal

UFN STATUS WORD
Bit          Meaning
BO(GS%DIR)   File is open
B1(GS%RD)    File is open for read access
B2(GS%WR)    File is open for write access
B3(GS%XCF)   File is open for execute access
B4(GS%RN)    File is open for non-append access
B7(GS%LEN)   File is longer than 512 pages
B8(GS%EOP)   Last read was past end of file
B9(GS%ERR)   File may be in error
B10(GS%SAM)  Filespec is associated with this JFN
B11(GS%AS)   JFN is parse-only
B12(GS%ASG)  JFN is currently being assigned
B13(GS%HLT)  I/O errors are considered terminating conditions
B17(GS%FRK)  JFN is restricted
B18(GS%PLN)  If on, file line numbers are passed during input; if 0, line numbers are stripped before input
B32-B35(GS%MOD)  Date mode of the file

FUNCTION

GTTYP  JSYS 903

Returns the terminal type number for the specified terminal line.

CALLING SEQUENCE

AC1:  Terminal designator

RETURNS  +1: Always, with
TOPS-20 Monitor Calls Quick Reference Guide

AC2: Terminal type number
AC3: BO-17 # of input buffers to allocate
      B18-25 # of output buffers to allocate

HALTJ SYS 170

FUNCTION
Halts the current process and any inferior processes of the
current process. Sets B1-17(RF%STS) in the Process Status
Word to 2(.RFHLT).

HORKJ SYS 162

FUNCTION
Halts one or more inferior processes.

CALLING SEQUENCE
AC1: Process handle (inferior processes only)

RETURNS +1: Always

HPTIMJ SYS 501

FUNCTION
Returns the value of one of the high precision system
clocks.

CALLING SEQUENCE
AC1: Number of the clock to read

RETURNS +1: Failure, error code in AC1
          +2: Success, with
              AC1: Value of specified clock

CLOCKS
Code Symbol Meaning
0 .HELP Elapsed time since system startup
1 .HPRNT CPU runtime for this process

HSYSJ SYS 307

FUNCTION
Initiates an orderly shutdown of timesharing.

RESTRICIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.
CALLING SEQUENCE
AC1: Shutdown time with date and time in internal format
AC2: Date/time (internal format) when system will resume; 0 if unknown

RETURNS +1: Failure, error code in AC1
+2: Success, shutdown procedure initiated

IDCNV  JSYS 223

FUNCTION
Converts separate numbers for the local year, month, day, and time into internal date and time format.

CALLING SEQUENCE
AC2: Year, month
AC3: <day of month>, 0
AC4: B0(IC%DSA) Apply daylight savings according to setting of B1(IC%ADS)
     B1(IC%ADS) Apply daylight savings if B0(IC%DSA) is on
     B2(IC%UTZ) Use time zone in B12-17; if off, use local zone
     B3(IC%JD) Number in B18-35 of AC2 is in Julian day format
     B12-17(IC%TMZ) Time zone if B2(IC%UTZ) is on
     B18-35(IC%TIM) Local time in seconds since midnight

RETURNS +1: Failure, error code in AC1
+2: Success, with
     AC2: Internal date and time
     AC3: B0 and B2 On for compatibility with ODCNV
          B1(IC%ADS) Daylight savings was applied
          B12-17(IC%TMZ) Time zone used

IDTIM  JSYS 221

FUNCTION
Inputs the date and time and converts them to internal date and time format.

CALLING SEQUENCE
AC1: Source designator
AC2: Format option flags

RETURNS +1: Failure, with
     AC1: Updated byte pointer
     AC2: Error code
+2: Success, with
     AC1: Updated byte pointer
AC2: Date and time in internal format

**IDTIM Option Flags**

- B1(IT%NM) Month may not be numeric; ignore B2-3
- B2(IT%NNM) 2nd number in date is month
- B3(IT%ERR) Return error if order of day and month does not agree with setting of B2(IT%NNM)
- B7(IT%NIS) Seconds cannot be included in time specification
- B8(IT%AIS) Seconds (preceded by colon) must be included in time specification
- B9(IT%NAC) Colon cannot be used to separate hours and minutes
- B10(IT%AAC) Colon must be used to separate hours and minutes
- B11(IT%AMS) If B7-10 off, interpret time specification containing one colon as hmmm:ss
- B12(IT%AHM) If B7-10 off, interpret time specification containing one colon as hh:mm; return error if first field too large
- B14(IT%N24) Do not allow time specification in 24-hour format; require AM/PM specification
- B15(IT%NTM) Do not allow time specification to include AM, PM, NOON, or MIDNIGHT
- B16(IT%NTZ) Do not allow time zone specification

**IDTNC JYS 231**

FUNCTION

Inputs the date and/or time and converts it into separate numbers for the local year, month, day, or time.

**CALLING SEQUENCE**

AC1: Source designator
AC2: Format option flags

**RETURNS**

+1: Failure, with
   AC1: Updated byte pointer
   AC2: error code
+2: Success, with
   AC1: updated byte pointer
   If date was input
   AC2: Year., month
   AC3: <day of month>,<day of week>
   If time was input
   AC4: BO(IC%DAS) On if IT%NTI was set, or if IT%ADA was set and a time zone was input
   B1(IC%ADS) On if daylight savings time zone was input, or if IT%NTI was set
   BO(IC%UTZ) On if IT%NTI was set, or if IT%ADA was set

65
TOPS-20 Monitor Calls Quick Reference Guide

IDTNC

and a time zone was
input

B3(IC%JUD) On if a number in
Julian day format was
input

B12-17(IC%TMZ) Time zone supplied, or
local time zone

B18-35(IC%TIM) Time as seconds since
midnight

IDTNC Option Flags

B0(IT%NDT) Do not input date and ignore B1-3; if off, date required

B1(IT%NNM) Month may not be numeric; ignore B2-3

B2(IT%SNM) 2nd number in date is month

B3(IT%ERR) Return error if order of day/month does not match setting of B2(IT%SNM)

B6(IT%NTI) Do not input time and ignore B7-16; if off, time required

B7(IT%NIS) Seconds cannot be included in time specification

B8(IT%AIS) Seconds (preceded by colon) must be included

B9(IT%NAC) Colon cannot be used to separate hours and minutes

B10(IT%AAC) Colon must be used to separate hours and minutes

B11(IT%AMS) If B7-10 off, interpret time specification containing one colon as hh:mm:ss

B12(IT%AHM) If B7-10 off, interpret time specification containing one colon as hh:mm; return error if 1st field too large

B14(IT%N24) Do not allow time specification in 24-hour format; require AM/PM specification

B15(IT%NTM) Do not allow time specification to include AM, PM, NOON, or MIDNIGHT

B16(IT%NTZ) Do not allow time zone specification

FUNCTION

Initiates software interrupts on the specified channels in a process.

CALLING SEQUENCE

AC1: Process handle
AC2: 36-bit word (1Bn initiates interrupt on channel n)

RETURNS +1: Always
FUNCTION

Returns a logical name that is defined either for this job or for the system.

CALLING SEQUENCE

AC1: BO=17  Function code
     B18=35  Index into table of defined logical names
AC2: Byte pointer to string for logical name

RETURNS  +1: Failure, error code in AC1
          +2: Success, updated byte pointer in AC2

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.INLJB</td>
<td>List logical names defined for this job</td>
</tr>
<tr>
<td>1</td>
<td>.INLSY</td>
<td>List logical names defined for system</td>
</tr>
</tbody>
</table>

JFNS  JSYS 30

FUNCTION

Returns the filespec currently associated with the JFN.

CALLING SEQUENCE

AC1: Destination designator for ASCIIZ filename string
AC2: Indexable file handle; or pointer to filename string
AC3: Format control bits for string; or 0

RETURNS  +1: Always, with updated byte pointer in AC1

FORMAT CONTROL VALUES

<table>
<thead>
<tr>
<th>Value</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.JSNDF</td>
<td>Do not output this field</td>
</tr>
<tr>
<td>1</td>
<td>.JSAOF</td>
<td>Always output this field</td>
</tr>
<tr>
<td>2</td>
<td>.JSSSD</td>
<td>Suppress this field if system default</td>
</tr>
</tbody>
</table>

FORMAT CONTROL FIELDS

Field  Meaning
BO-2(JS%DEV)  Output for device field
B3-5(JS%DIR)  Output for directory field
B6-8(JS%NAME) Output for filename field (2 is illegal)
B9-11(JS%TYPE) Output for file type field (2 is illegal)
B12-14(JS%GEN) Output for generation number field
BO-14(JS%SPEC) Output for all filespec fields named above
B15-17(JS%PRG) Output for protection field
B18-20(JS%ACT) Output for account field
B21(JS%TMP)  Return ;T if appropriate
B22(JS%SIZE) Return size of file in pages
B23(JS%HDR)  Return creation date
B24(JS%DWR)  Return date of last write
B25(JS%LDR)  Return date of last read
B26(JS%PTR)  AC2 contains pointer to the string being returned
TOPS-20 Monitor Calls Quick Reference Guide

JFNS

B27 (JS%ATR)  Return filespec attributes if appropriate
B28 (JS%AT1)  Return specification attribute referenced in AC4
B29 (JS%OFL)  Return the "OFF-LINE" attribute
B32 (JS%PSD)  Punctuate the size and date fields
B33 (JS%TBR)  Tab before all fields returned, except for 1st field
B34 (JS%TBP)  Tab before all fields (except 1st) with value 1 or 2
B35 (JS%PAF)  Punctuate all fields from device through :T

KFORK USYS 153

FUNCTION
Kills one or more processes, releasing memory, PSB, and JFNs.

CALLING SEQUENCE
AC1:  Process handle

RETURNS
+1:  Always, unless current process attempts to kill itself

LOGOUT USYS 3

FUNCTION
Kills the specified job and appends an accounting entry to the accounting data file, unless job did not login.

RESTRICTIONS
WHEEL or OPERATOR required to logout job other than current job, job logged in under same username, or PTY job controlled by current job.

CALLING SEQUENCE
AC1:  Number of job to be logged out, or -1 for current job

RETURNS
+1:  Failure, error code in AC1
+2:  Success

LNMST USYS 504

FUNCTION
Translates a logical name to its original definition string.

CALLING SEQUENCE
AC1:  Function code
AC2:  Pointer to logical name string (without colon)
AC3:  Pointer to string for original logical name definition

68
TOPS-20 Monitor Calls Quick Reference Guide

RETURNS
+1: Failure, error code in AC1
+2: Success, updated byte pointer in AC3

FUNCTION CODES
Code  Symbol  Meaning
0    .LNSUB  Obtain job-wide definition of logical name
1    .LNSSY  Obtain system-wide definition of logical name

LOGIN   JSYS 1

FUNCTION
Logs a job into the system.

RESTRICIONS
In non-zero sections, OWGBP's must specify 7-bit bytes.

CALLING SEQUENCE
AC1: 36-bit user number for login
AC2: Pointer to beginning of password string
AC3: 5B21<account number>B35 or pointer to account string
(maximum of 39 characters read)

RETURNS
+1: Failure, error code in AC1
+2: Success, with
     AC1: Date and time of last login
     AC2: Updated byte pointer
     AC3: Updated byte pointer

LPINI   JSYS 547

FUNCTION
Loads the direct access Vertical Formatting Unit (VFU) or
translation Random Access Memory (RAM) for the line printer.

RESTRICIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: JFN of file containing VFU or RAM
AC2: <status bits>,<function code>
AC3: Unit number of line printer

RETURNS
+1: Always

STATUS BITS
Bit  Symbol  Meaning
B0  MD%LCP  Line printer is lowercase

FUNCTION CODES
Code  Symbol  Meaning
32  .MOLVF  Load VFU from file indicated by JFN
34  .MOLTR  Load translation RAM from file indicated by
TOPS-20 Monitor Calls Quick Reference Guide
LPINI

JFN

MDDT% JSYS 777

FUNCTION
Transfers control to the MDDT program while preserving the
context of the process that issued the MDDT% JSYS.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

MEET% JSYS 766

FUNCTION
Returns the value of the execution accounting meter or the
memory reference accounting meter.

RESTRICTIONS
Not available on KS-10 hardware.

CALLING SEQUENCE
AC1: Function code

RETURNS +1: Always, with 59-bit value in AC2 and AC3

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.MERA</td>
<td>Read process execution accounting meter doubleword; returns EBOX busy time (number of EBOX ticks)</td>
</tr>
<tr>
<td>2</td>
<td>.MERA</td>
<td>Read process memory-reference accounting meter doubleword; returns count of MBOX references (number of MBOX ticks)</td>
</tr>
</tbody>
</table>

DOUBLE-WORD FORMAT

<table>
<thead>
<tr>
<th>AC2</th>
<th>AC3</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Order Part</td>
<td>Low Order Part</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>24</td>
</tr>
</tbody>
</table>

MRECV JSYS 511

FUNCTION
Retrieves an IPCF (Inter-Process Communication Facility)
message from the process’s input queue.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR or IPCF capability.
CALLING SEQUENCE
AC1: Length of packet descriptor block
AC2: Address of packet descriptor block

RETURNS
-1: Failure, error code in AC1
+2: Success, with
   AC1: BO-17 Length of next entry in queue
   B18-35 Flags from next packet or 0 if queue empty

FORMAT OF PACKET DESCRIPTOR BLOCK
Word Symbol Meaning
0 .IPCFL Flags
1 .IPCFS PID of sender (RET)
2 .IPCFR PID of receiver; -1 for any PID in process;
   -2 for any PID in job
3 .IPCFF <length of message>.<destination address>
4 .IPCFC User number of sender (RET)
5 .IPCFE Enabled capabilities of sender (RET)
6 .IPCSD Number of sender’s connected directory (RET)
7 .IPCAS Account string of sender (RET)
10 .IPCLL Byte pointer for destination of sender’s node
   (optional)

FLAGS FOR WORD .IPCFL OF PACKET DESCRIPTOR BLOCK
Bit Symbol Meaning
B0 IP%CFB Do not block process if no messages in
   queue; if set, error return if no messages
B1 IP%CFS Use PID referenced in word .IPCFS as
   sender’s PID
B2 IP%CFR Use PID referenced in word .IPCFR as
   receiver’s PID
B3 IP%CFO Allow one send request above quota
B4 IP%TTL Truncate message if larger than space
   reserved
B5 IP%CPD Create PID for sender and return in word
   .IPCFS
B6 IP%JWP Make created PID job wide (ignored unless
   IP%CPD set)
B7 IP%NDA Do not allow other processes to use created
   PID (ignored unless IP%CPD set)
B18 IP%CPF Packet is privileged (requires IPCF)
B19 IP%CFV Packet is page of data
B21 IP%INT Reserved
B22 IP%EPN 18-bit page number in word .IPCFP

MSEND USYS 510

FUNCTION
Sends an IPCF (Inter-Process Communication Facility) message
to a specific PID or to <SYSTEM>INFO.

RESTRICTIONS
Some functions require WHEEL, OPERATOR, or IPCF capability
enabled.

CALLING SEQUENCE
AC1: Length of packet descriptor block
AC2: Address of packet descriptor block

RETURNS  +1: Failure, error code in AC1
         +2: Success

FORMAT OF PACKET DESCRIPTOR BLOCK
Word Symbol Meaning
0  .IPCFI Flags
1  .IPCFS PID of sender; address of PID if IP%CF or
   IP%CFR is set in word .IPCFI; or 0 if no PID
   exists for sender (RET if creating a PID)
2  .IPCFR PID of receiver: 0 if receiver is
   <SYSTEM>INFO
3  .IPCFP <message length>, <message starting address>

FLAGS FOR WORD .IPCFI OF PACKET DESCRIPTOR BLOCK
Bit Symbol Meaning
B0  IP%CFB Do not block process if no messages in
     queue; if set, error return if no messages
B1  IP%CFI Use PID referenced in word .IPCFS as
     sender's PID
B2  IP%CFR Use PID referenced in word .IPCFR as
     receiver's PID
B3  IP%CFD Allow one send request above quota
B4  IP%CTT Truncate message if larger than space
     reserved
B5  IP%CFP Create PID for sender and return in word
     .IPCFS
B6  IP%JWP Make created PID job wide (ignored unless
     IP%CFD set)
B7  IP%NDA Do not allow other processes to use created
     PID (ignored unless IP%CFD set)
B18 IP%CFP Packet is privileged (requires IPCF)
B19 IP%CFV Packet is page of data
B21 IP%INT Reserved
B22 IP%EPN 18-bit page number in word .IPCFP

FLAGS RETURNED IN WORD .IPCFI
Bit Symbol Meaning
B20 IP%CFZ Zero-length message was sent; packet
     consists of only packet descriptor block
B24-29 IP%CFE Error code field for <SYSTEM>INFO errors
      15  .IPCPI Insufficient privileges
      16  .IPCUF Invalid function
      67  .IPCSI <SYSTEM>INFO needs name
      72  .IPCFF <SYSTEM>INFO free space
           exhausted
      74  .IPCNP PID has no name or is invalid
      75  .IPCDD Duplicate name has been
           specified
      76  .IPCNN Unknown name has been
           specified

72
TOPS-20 Monitor Calls Quick Reference Guide
MSEND

77 .IPCEN Invalid name has been specified

B30-32 IPuracy System and sender code (enabled IPCF to set)
1 .IPCCC Sent by <SYSTEM>IPCF
2 .IPCCF Sent by system-wide
   <SYSTEM>INFO
3 .IPCCP Sent by receiver's
   <SYSTEM>INFO

B33-35 IP%CFM Special messages field (enabled WHL)
1 .IPCFCF Process's input queue contains undeliverable packet

FORMAT OF REQUEST PACKET TO <SYSTEM>INFO
Word Symbol Meaning
0 .IPCIO <user-defined code>,,<SYSTEM>INFO function>
1 .IPCII PID to receive copy of <SYSTEM>INFO's response
2 .IPCII Function-specific argument

<SYSTEM>INFO FUNCTION CODES
Function Argument Meaning
.IPCIW Name Return PID associated with specified name in word .IPCII
.IPCIQ PID Return name associated with specified PID in word .IPCII
.IPCII ASCIZ name Assign specified name to PID of process making request
.IPCIJ ASCIZ name Same as .IPCII function
.IPCIIK PID Inform PID when PID in word .IPCII is deleted (WHL/GPR)
.IPCIS PID Disassociate all PIDs with names (not available to user programs)

MSFRK JSYS 312

FUNCTION
Starts a process in monitor mode.

REstrictions
Requires enabled WHEEL or OPERATOR capability, or execution from monitor mode.

AC1: Process handle
AC2: PC word: <user mode flags>,,<virtual address>

RETURNS +1: Always
TOPS-20 Monitor Calls Quick Reference Guide

MSTR

MSTR uSYS 558

FUNCTION
Performs various structure-dependent functions.

RESTRICTIONS
Some functions require enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: <length of argblk>,<function code>
AC2: Address of argblk

RETURNS +1: Always, with some functions returning data in argblk

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Privileges</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.MSRNU</td>
<td>WH/DPR/MNT</td>
<td>Return status of next disk unit</td>
</tr>
<tr>
<td>1</td>
<td>.MSRUS</td>
<td>WH/DPR/MNT</td>
<td>Return status of given disk unit</td>
</tr>
<tr>
<td>2</td>
<td>.MSMNT</td>
<td>WH/DPR</td>
<td>Mount structure</td>
</tr>
<tr>
<td>3</td>
<td>.MSDIS</td>
<td>WH/DPR</td>
<td>Dismount structure</td>
</tr>
<tr>
<td>4</td>
<td>.MSGSS</td>
<td>--</td>
<td>Return status of structure</td>
</tr>
<tr>
<td>5</td>
<td>.MSRSS</td>
<td>WH/DPR</td>
<td>Change status of structure</td>
</tr>
<tr>
<td>6</td>
<td>.MSINI</td>
<td>WH/DPR</td>
<td>Initialize structure</td>
</tr>
<tr>
<td>7</td>
<td>.MSIMC</td>
<td>--</td>
<td>Increment job’s mount count for structure</td>
</tr>
<tr>
<td>10</td>
<td>.MSDMC</td>
<td>--</td>
<td>Decrement job’s mount count for structure</td>
</tr>
<tr>
<td>11</td>
<td>.MSGSU</td>
<td>--</td>
<td>Return job #s of structure users</td>
</tr>
<tr>
<td>12</td>
<td>.MSHOM</td>
<td>WH/DPR</td>
<td>Modify home block of structure</td>
</tr>
<tr>
<td>13</td>
<td>.MSICF</td>
<td>--</td>
<td>Increment fork’s mount count for structure</td>
</tr>
<tr>
<td>14</td>
<td>.MSDCF</td>
<td>--</td>
<td>Decrement fork’s mount count for structure</td>
</tr>
<tr>
<td>15</td>
<td>.MSEFL</td>
<td>WH/DPR</td>
<td>Receive interrupt when disk comes on-line</td>
</tr>
<tr>
<td>16</td>
<td>.MSIIC</td>
<td>WH/DPR</td>
<td>Ignore increment check for structure use</td>
</tr>
</tbody>
</table>

ARGUMENT BLOCKS

<table>
<thead>
<tr>
<th>Function</th>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>.MSRNU</td>
<td>0</td>
<td>.MSRCH</td>
<td>Channel number (0-7)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>.MSRCT</td>
<td>Controller number</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.MSRUN</td>
<td>Unit number (0-7)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.MSRST</td>
<td>Returned software status of unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BO(MS%MNT) Unit part of mounted structure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B2(MS%DIA) Unit in on-line diagnostics</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B3(MS%OFI) Unit is off-line</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B4(MS%ERR) Unit has read error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B5(MS%BBB) Unit has bad BAT block</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B6(MS%HBB) Unit has bad HOME block</td>
</tr>
</tbody>
</table>

74
B7(MS%WLK) Unit is write locked
B9-17(MS%TYP) Type of disk unit
  1 .MSRP4 RP04
  5 .MSRP5 RP05
  6 .MSRP6 RP06
  7 .MSRP7 RP07
 11 .MSRM3 RM03
 24 .MSR20 RP20
4 .MSRSN Byte pointer to ASCIZ structure name string
5 .MSRSA Byte pointer to ASCIZ structure alias string
6 .MSRNS <unit #>,<# units in structure>
7 .MSRSW Number of pages for swapping on the structure
10-12 .MSRUI Unit ID (3 words of 11-formatted ASCII)
13-15 .MSROI Owner ID (3 words of 11-formatted ASCII)
16-20 .MSRFI File system ID (3 words of 11-formatted ASCII)
21 .MSRSP Number of sectors per page
22 .MSRSC Number of sectors per cylinder
23 .MSRPC Number of pages per cylinder
24 .MSRCU Number of cylinders per unit
25 .MSRSC Number of sectors per unit
26 .MSRBT Number of bit words in bit table per cylinder
27 .MSRSE Serial number of CPU for which structure is used in booting system
O-27 Same as .MSRNU
0 .MSTNM Pointer to ASCIZ string for structure name
1 .MSTAL Pointer to ASCIZ string for structure alias
2 .MSTFL Flags, <# units in structure>
   B0(MS%FH) Do not fix bad HOME blocks
   B1(MS%FB) Do not fix bad BAT blocks
   B2(MS%CL) Mount structure for exclusive use by job
   B3(MS%GN) Ignore correctable errors in bit table and root directory
3 .MSTUI 3 words of data for each unit in structure
 0 .MSTCH Channel # of unit
 1 .MSTCT Controller # of unit
 2 .MSTUN Unit # of unit
0 .MSDIS Device designator, or pointer to ASCIZ structure alias string
0 .MSGSS Device designator, or pointer to ASCIZ structure alias string
1 .MSGST Returned status word
   B0(MS%PS) Structure is public

TOPS-20 Monitor Calls Quick Reference Guide

MSTR

B1(MS%DIS) Structure is being dismounted
B2(MS%DOM) Structure is domestic
B3(MS%PPS) Structure is PS
B4(MS%INI) Structure is being initialized
B5(MS%LIM) Directory size on structure limited to 30 pages
B6(MS%NRS) Structure is non-regulated

2 .MSGNU Number of units in structure
3 .MSGMC Mount count for this structure
4 .MSGFC Open file count for this structure
5 .MSGSI Pointer to ASCII string for structure's physical ID

0 .MSSSN Device designator, or pointer to ASCII structure alias string
1 .MSSST Word containing new values for bits being changed
2 .MSMWM Mask containing bits being changed

B1(MS%DIS) Structure is being dismounted
B2(MS%DOM) Structure is domestic
B6(MS%NRS) Structure is non-regulated
B7(MS%RWS) Read-after-write checking in swapping area
B8(MS%RWD) Read-after-write checking in data area

0 .MSINM Byte pointer to ASCII structure name string
1 .MSIAL Byte pointer to ASCII string containing alias of structure
2 .MSIFL B0(MS%NH) Do not fix bad HOME block

B1(MS%NB) Do not fix bad BAT block
B2(MS%XCL) Mount structure for exclusive use by job
B3(MS%IGN) Ignore errors in bit table and root directory

B12-17(MS%FCN) Function
1 .MSCRE Create new file system
2 .MSRRD Reconstruct root directory
3 .MSSHW Write new HOME blocks
4 .MSRIX Rebuild index table

B18-35 (.MSINU) # of units in structure
3-5 .MSISU 3 words of data for each unit in
<table>
<thead>
<tr>
<th>Structure Field</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>.MSICH</td>
<td>Channel # of unit</td>
</tr>
<tr>
<td>.MSICT</td>
<td>Controller # of unit</td>
</tr>
<tr>
<td>.MSIUN</td>
<td>Unit # of unit</td>
</tr>
<tr>
<td>.MSIST</td>
<td>Status word (reserved)</td>
</tr>
<tr>
<td>.MSISW</td>
<td>Number of pages for swapping on structure</td>
</tr>
<tr>
<td>.MSIFE</td>
<td>Number of pages for front-end file system</td>
</tr>
<tr>
<td>.MSUII</td>
<td>Unit ID (3 words of ASCII)</td>
</tr>
<tr>
<td>.MSODI</td>
<td>Owner ID (3 words of ASCII)</td>
</tr>
<tr>
<td>.MSIFI</td>
<td>File system ID (3 words of ASCII; reserved)</td>
</tr>
<tr>
<td>.MSIFB</td>
<td>Number of pages for file BOOTSTRAP.BIN</td>
</tr>
<tr>
<td>.MSISN</td>
<td>Serial number of CPU for which structure is used in booting system</td>
</tr>
<tr>
<td>.MSDEC</td>
<td>Device designator, or pointer to ASCIZ structure alias string</td>
</tr>
<tr>
<td>.MSJOB</td>
<td>Number of job (if not current job) whose mount count is to be incremented; (optional; enabled WHL/OPR)</td>
</tr>
<tr>
<td>.MSODEV</td>
<td>Device designator, or pointer to ASCIZ structure alias string</td>
</tr>
<tr>
<td>.MSUFL</td>
<td>Number of job (if not current job) whose mount count is to be decremented; (optional; enabled WHL/OPR)</td>
</tr>
<tr>
<td>.MSUAL</td>
<td>Device designator, or pointer to ASCIZ structure alias string</td>
</tr>
<tr>
<td>.MSUGW</td>
<td>&lt;flag bits&gt;, 0</td>
</tr>
<tr>
<td>.MSUGW</td>
<td>B0(MSUGW) Return users who have accessed structure</td>
</tr>
<tr>
<td>.MSUGW</td>
<td>B1(MSUGW) Return users who have incremented mount count</td>
</tr>
<tr>
<td>.MSUGW</td>
<td>B2(MSUGW) Return users who are connected to structure</td>
</tr>
<tr>
<td>.MSHDF</td>
<td>Offset specifying which word should be changed</td>
</tr>
<tr>
<td>.MSHVL</td>
<td>Value for new bits</td>
</tr>
<tr>
<td>.MSHMK</td>
<td>Mask showing which bits should be changed</td>
</tr>
<tr>
<td>.MSIEN</td>
<td>Device designator, or pointer to ASCIZ structure alias string</td>
</tr>
<tr>
<td>.MSIDEN</td>
<td>Device designator, or pointer to ASCIZ structure alias string</td>
</tr>
<tr>
<td>.MSIFM</td>
<td>Place process on software interrupt channel; if -1, deassign previously assigned channel</td>
</tr>
</tbody>
</table>

DATA RETURNED BY FUNCTION .MSG5U

<table>
<thead>
<tr>
<th>Word Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 .MSUFL</td>
<td>&lt;flag bits from call&gt;, &lt;# of items returned&gt;</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

MSTR

2 .MSUJ1 <flag bits for job>,<job #>
   <flag bits for job>,<job #>
   BO(MS%GTA) Job accessed structure
   B1(MS%GTM) Job incremented mount count for
   structure
   B2(MS%GTC) Job connected to structure

11-Formatted ASCII

<table>
<thead>
<tr>
<th>CHAR 1</th>
<th>CHAR 0</th>
<th>CHAR 3</th>
<th>CHAR 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR 4</td>
<td>CHAR 7</td>
<td>CHAR 6</td>
<td></td>
</tr>
<tr>
<td>CHAR 8</td>
<td>CHAR 11</td>
<td>CHAR 10</td>
<td></td>
</tr>
</tbody>
</table>

MTALN JSYS 774

FUNCTION
Associates a given magtape drive with the specified logical unit number.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: <drive type>,<logical unit # of magtape>
AC2: Decimal serial number of magtape drive

RETURNS +1: Always

MTDPR JSYS 77

FUNCTION
Performs various device-dependent control functions.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability;
or ARPA net or DECnet software.

CALLING SEQUENCE
AC1: JFN of device
AC2: Function code
AC3: Function arguments or address of argblk
AC4: Function arguments (if required)

RETURNS +1: Always, with
   AC2: Requested data
   AC3: Requested data or updated byte pointer

ARPANET FUNCTION CODES

78
TOPS-20 Monitor Calls Quick Reference Guide

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>.MGACP</td>
<td>If connection in RFCR state, send RFC to accept</td>
</tr>
<tr>
<td>21</td>
<td>.MOSND</td>
<td>If connection in buffered send mode, send all currently buffered bytes</td>
</tr>
<tr>
<td>22</td>
<td>.MOSIN</td>
<td>Send INS/INR command</td>
</tr>
<tr>
<td>23</td>
<td>.MOAIN</td>
<td>Simulate CLS F.S.M. action</td>
</tr>
<tr>
<td>24</td>
<td>.MOAIN</td>
<td>Assign interrupt channels for change of state or INS/INR message receipt</td>
</tr>
<tr>
<td>AC3</td>
<td>B0-5</td>
<td>INS/INR PSI channel</td>
</tr>
<tr>
<td></td>
<td>B12-17</td>
<td>State change PSI channel</td>
</tr>
<tr>
<td>25</td>
<td>--</td>
<td>If input, send allocate message; if output, wait for allocate message</td>
</tr>
<tr>
<td>26</td>
<td>--</td>
<td>Setup 1st I/O buffer and send allocate without requiring user I/O</td>
</tr>
</tbody>
</table>

**DECnet FUNCTION CODES**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>.MQACN</td>
<td>Allow network task to enable interrupt channels for some tasks</td>
</tr>
<tr>
<td>AC3</td>
<td>B0-8(MO%CDN)</td>
<td>Connect event pending</td>
</tr>
<tr>
<td></td>
<td>B9-17(MO%INA)</td>
<td>Interrupt message available</td>
</tr>
<tr>
<td></td>
<td>B18-26(MO%DAV)</td>
<td>Data available</td>
</tr>
<tr>
<td></td>
<td>nh</td>
<td># of channel to be enabled: 0 to 5, 23 to 35</td>
</tr>
<tr>
<td></td>
<td>.MCIA</td>
<td>Clear interrupt</td>
</tr>
<tr>
<td></td>
<td>.MDNCI</td>
<td>No change</td>
</tr>
<tr>
<td>25</td>
<td>.MORLS</td>
<td>Return logical link status</td>
</tr>
<tr>
<td>AC3</td>
<td>&lt;flag bits&gt;...&lt;disconnect code&gt; (RET)</td>
<td>(RET)</td>
</tr>
<tr>
<td></td>
<td>B0(MO%CDN)</td>
<td>Link is connected</td>
</tr>
<tr>
<td></td>
<td>B1(MO%SRR)</td>
<td>Link is a server</td>
</tr>
<tr>
<td></td>
<td>B2(MO%RFC)</td>
<td>Link waiting for connection</td>
</tr>
<tr>
<td></td>
<td>B3(MO%RCC)</td>
<td>Link waiting for connect confirmation</td>
</tr>
<tr>
<td></td>
<td>B4(MO%EOM)</td>
<td>Link has entire message to be read</td>
</tr>
<tr>
<td></td>
<td>B5(MO%ABT)</td>
<td>Link has been aborted</td>
</tr>
<tr>
<td></td>
<td>B6(MO%SBN)</td>
<td>Link has been closed normally</td>
</tr>
<tr>
<td></td>
<td>B7(MO%INT)</td>
<td>Link has interrupt message available</td>
</tr>
<tr>
<td></td>
<td>B8(MO%LWC)</td>
<td>Link has been previously connected</td>
</tr>
</tbody>
</table>

**Disconnect codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.DCX0</td>
</tr>
<tr>
<td>1</td>
<td>.DCX1</td>
</tr>
<tr>
<td>2</td>
<td>.DCX2</td>
</tr>
<tr>
<td>3</td>
<td>.DCX3</td>
</tr>
<tr>
<td>4</td>
<td>.DCX4</td>
</tr>
<tr>
<td>5</td>
<td>.DCX5</td>
</tr>
<tr>
<td>6</td>
<td>.DCX6</td>
</tr>
</tbody>
</table>

79
overflow
7. DCX7  Unspecified error
8. DCX8  Third party aborted link
9. DCX9  User abort (asynchronous disconnect)
11. DCX11 Undefined error code
21. DCX21 Connect initiate with illegal destination address
22. DCX22 Connect confirm with illegal destination address
23. DCX23 Connect initiate or confirm with zero source address
24. DCX24 Flow control violation
32. DCX32 Too many connections to node
33. DCX33 Too many connections to destination process
34. DCX34 Access not permitted
35. DCX35 Logical link services mismatch
36. DCX36 Invalid account
37. DCX37 Segment size too small
38. DCX38 Process aborted
39. DCX39 No path to destination node
40. DCX40 Link aborted due to data loss
41. DCX41 Destination process does not exist
42. DCX42 Confirmation of disconnect initiate
43. DCX43 Image data field too long

26. MDRHN Return ASCII host node name at other end of logical link
27. MDRTN Return unique task name associated with this end of logical link
30. MDRUS Return source task user identification supplied in connect initiate message
31. MDRPW Return source task's password supplied in connect initiate message
32. MORAC Return account string supplied by source task in connect initiate message
33. MORDA Return optional data supplied in connect/disconnect messages
34. MORCN Return object type used by source task to address connection
35. MDRIM Read interrupt message
AC3: Byte pointer to receiving buffer (8-bit)

36 .MOSIM Send interrupt message
AC3: Byte pointer to message (8-bit)
AC4: Count of bytes in message (16 maximum)

37 .MOROD Return unique identification of source task
AC3: Pointer to string for source task
object-descriptor (8-bit)

40 .MOCLZ Reject connection either implicitly or explicitly
AC2: <reject code>, .MOCLZ
AC3: Pointer to string for returned data
(8-bit)
AC4: Count of bytes in data string (16 maximum)

41 .MOCC Accept connection either implicitly or explicitly
AC3: Pointer to string for returned data
AC4: Count of bytes in data string (16 maximum)

42 .MORSS Return maximum segment size that can be sent
over this link; (illegal unless link in run
state)

43 .MOANT Attach network terminal
AC3: Address of argblk
0 Count including this word
1 .SHTY ID of TTY controlling local
job
2 .SHESC Flags, <ASCII escape char>
SHLPM Local page mode

FRONT-END FUNCTION CODES

Code Symbol Meaning/Arguments
3 .MOEDF Flush TOPS-20 buffers and send all data to
front end
AC3: 0 Flush buffers and send EOF to FE
#0 Flush buffers only

4 .MODTE Assign specified device to DTE controller on
front end (enabled WHL/OPR)
AC3: Device type (WHL/OPR)

MTA/MT FUNCTION CODES

Code Symbol Meaning/Arguments
0 .MOCLE Clear any error flags from previous MTDPR
1 .MOREW Rewind tape; if labeled, mount 1st volume in
set and position at BOT
2 .MOSDR Set direction of tape motion for reading
(unlabeled only)
AC3: 0 Read forwards
1 Read backwards
3 .MOEDF Write tape mark
4 .MOSDM Set hardware data mode for tape data transfer
AC3: Hardware data mode
5 .MOSRS Set record size
AC3: Record size in bytes
### TOPS-20 Monitor Calls Quick Reference Guide

**MDFWR**
- Advance one record in read direction

**MDBKRR**
- Back up one record from read direction

**MDEOT**
- Advance to EOT (unlabeled) or EDV (labeled)

**MGRUL**
- Rewind and unload tape (illegal for MOUNTed tapes)

**MDORDN**
- Return density

**MDERS**
- Erase tape gap (unlabeled only)

**MORDM**
- Return hardware data mode

**MORRS**
- Return record size

**MDFWF**
- Advance to next tape mark

**MOBKFR**
- Backup to last tape mark or BOT

**MDSPR**
- Set parity
  - AC3: Desired parity
    - 0: .SUPT0 Odd parity
    - 1: .SUPT0 Even parity

**MDPRR**
- Return parity

**MONRB**
- Return number of bytes remaining in current record

**MOFOU**
- Force output of partial records during sequential write

**MOSDN**
- Set density (unlabeled only)
  - AC3: Desired density

**MOINF**
- Return tape information
  - AC3: Address of argbik
    - 0: .MOICT Word count not including this word
    - 1: .MOITP MTA type code
    - 2: .MOIDT MTA reel ID
    - 3: .MOISN LH Channel/controller/unit RH Serial #
    - 4: .MOIRD Number of reads done
    - 5: .MOIWT Number of writes done
    - 6: .MOIRC Record number from BOT
    - 7: .MOIFC Number of files on tape
    - 8: .MOISR Number of soft read errors
    - 9: .MOISW Number of soft write errors
    - 10: .MOIHR Number of hard read errors
    - 11: .MOIHFR Number of hard write errors
    - 12: .MOIFW Number of frames read
    - 13: .MOIFWF Number of frames written

**MORDR**
- Return read direction
  - AC3: 0 Forwards
  - 1 Backwards

**MOSID**
- Set reel ID of mounted tape (enabled WHL/DPR)
  - AC3: 36-bit reel ID

**MOIEL**
- Set error logging for tape
  - AC3: 0 Log errors
  - #0 Do not log errors

**MONDP**
- Wait for all activity to stop

**MOLOC**
- Identify 1st volume in MOUNT request or next volume for volume switch (WHL/DPR)
  - AC3: Pointer to argbik
    - 0: .MOCNT Word count
    - 1: .MOCTN MT unit # to associate with MTA
    - 2: .MOLBT Label type

---

82
3 .MODNS Density
4 .MDAVL Address of volume labels
5 .MDVNL # of volume labels at
   .MOAVL
6 .MDCVNL Volume number in volume set
7 .MDVSN SIXBIT file set identifier
37 .MOSTA Return current magtape status
   AC3: Address of argblk
   0 .MDCNT Word count including this
      word
1 .MODDN Density flags (RET)
   B1(SJ%CP2) 200 BPI
   B2(SJ%CP8) 556 BPI
   B3(SJ%CP8) 800 BPI
   B4(SJ%CP8) 1600 BPI
   B5(SJ%CP8) 6250 BPI
2 .MODDM Data mode flags (RET)
   B1(SJ%CMC) Core dump
   B2(SJ%CMG) SIXBIT
   B3(SJ%CMC) ANSI ASCII
   B4(SJ%CM) Industry
      compatible
   B5(SJ%CMH) High density
      mode
3 .MOTRK Recording track flags (RET)
   B1(SJ%CTR) 7-track drive
   B2(SJ%CTR) 9-track drive
4 .MOCSF Tape status flags (RET)
   B0(SJ%OFS) Off-line
   B1(SJ%MA1) Maintenance
      mode enabled
   B2(SJ%MRQ) Maintenance
      mode requested
   B3(SJ%RST) Beginning of
      tape
   B4(SJ%REW) Rewinding
   B5(SJ%WLK) Write locked
40 .MODFL Enable interrupts for on-line/off-line
     transition (WHL/DPR)
42 .MOPST Set interrupt channel to indicate
     availability of UHL(BOV)/UTL(EDV) labels
     AC3: PSI channel; -1 to clear
43 .MORVL Rewind current labeled tape volume
44 .MOVLS Switch volumes for unlabeled multi-volume set
   AC3: Address of argblk
   0 Word count including this word
   1 Flags: <function code>
      1 .VSMNV Mount absolute volume #
      2 .VFST Mount 1st volume in set
      3 .VSLST Mount last volume in set
      4 .VSMRV Mount relative volume #
      5 .VSFLS Force volume switch
         (labeled only)
   2 Volume number (if required)
45 .MONTR Set translate flag (EBCDIC ==> ASCII; labeled
TOPS-20 Monitor Calls Quick Reference Guide

MTOPR

only)
AC3: 0 Clear translate flag
   #0 Set translate flag

46 .MORDL Read user header labels
AC3: Pointer to string for label

47 .MOWUL Write user header or trailer labels (labeled only)
AC3: Byte pointer to label contents (must be 76 bytes)
AC4: Label identifier code (any ASCII char)

50 .MOLAI Read available fields from volume and header labels
AC3: Pointer to argblk
   0 Word count
   1 Label type (RET)
      1 .LTUNL unlabeled
      2 .LTANS ANSI
      3 .LTEBC EBCDIC
      4 .LTT20 TOPS-20
   2 Byte pointer to string for volume name
   3 Byte pointer to string for owner name
   4 Tape format (RET)
   5 Record length (RET)
   6 Block length (RET)
   7 Creation date (RET)
   10 Expiration date (RET)
   11 Byte pointer to string for file name
   12 Generation number (RET)
   13 Version number (RET)
   14 Form-control value (RET)
      SP No line format characters
      A FORTRAN format control characters
      M All necessary line format characters
      X Data in stream mode

51 .MOSMV Value for form-control field in HDR2 label
AC3: Mode
   0 .TPFST X
   1 .TPFCP M
   2 .TPFFC A
   3 .TPFNCE Space

52 .MOSDS Set deferred volume switch (labeled only)

PLPT FUNCTION CODES

Code Symbol Meaning/Arguments
27 .MOPSI Enable software interrupt on nonfatal device conditions
   AC3: Address of argblk
      0 Word count including this word
      1 Interrupt channel number
      2 Flags
      BO(MO%MSG) Suppress CTY device
31 .MONDP  Wait for all activity to stop
32 .MOLVF  Load line printer's VFU from file referenced in argblk
            AC3: Address of argblk
                0 Word count including this word
                1 JFN of file containing VFU
33 .MORVF  Read name of current VFU file in monitor's
            data base
            AC3: Address of argblk
                0 Word count including this word
                1 JFN of file containing VFU
                2 Number of bytes in string
34 .MOLTR  Load line printer's translation RAM from file
            referenced in argblk
            AC3: Address of argblk
                0 Word count including this word
                1 JFN of file containing translation
                    RAM
35 .MORTR  Read name of current translation RAM file in
            monitor's data base
            AC3: Address of argblk
                0 Word count including this word
                1 Pointer to string for ASCIZ name
                2 Number of bytes in string
36 .MOSTS  Set status of line printer
            AC3: Address of argblk
                0 Word count including this word
                1 Software status word
                    BO(MO\%LCP)  Printer is lowercase
                    B12(MO\%EOF) Set MO\%EOF when all
                    data printed
                    B14(MO\%SER) Clear software error
                    condition
                2 Value for page counter register
37 .MORST  Read line printer status
            AC3: Address of argblk
                0 Word count including this word
                1 Status word
                    BO(MO\%LCP)  Printer is lowercase
                    B1(MO\%RLD)  FE has been reloaded
                    B10(MO\%FER) Fatal hardware error
                    occurred
                    B12(MO\%EOF) All data sent has been
                    printed
                    B13(MO\%IDP) Output in progress
                    B14(MO\%SER) Software error
                    occurred
                    B15(MO\%HE) Hardware error
                    occurred
                    B16(MO\%OL) Printer is off-line
                    B17(MO\%FAX) Printer does not exist
                    B30(MO\%RPE) RAM parity error
                    occurred
                    B31(MO\%LVU) Printer has optical
                    VFU

85
TOPS-20 Monitor Calls Quick Reference Guide
MTOPR

B33(MO%LFV) VFU error occurred
B34(MO%LCI) Character interrupt occurred
B35(MO%LPC) Page counter register overflowed

2 Value of page counter register

.MDFLD Flush any output not yet printed

PCDP FUNCTION CODES
Code Symbol Meaning/Arguments
27 .MPSI Enable software interrupt on nonfatal device conditions
0 AC3: Address of arbblk
1 0 Word count including this word
2 1 Interrupt channel number
3 Flags
4 BO(MO%MSG) Suppress CTY device messages

37 .MORST Read card punch status
AC3: Address of arbblk
0 0 Word count including this word
1 1 Status word
2 B10(MO%FER) Fatal error condition
3 B12(MO%EOF) All pending output processed
4 B13(MO%INF) Output in progress
5 B14(MO%SER) Software error occurred
6 B15(MO%HE) Hardware error occurred
7 B16(MO%OL) Card punch is off-line
8 B17(MO%FX) Punch doesn’t exist
9 B32(MO%EM) Hopper empty or stacker full
10 B33(MO%SC) Stack check
11 B34(MO%PK) Pick check
12 B35(MO%CK) Read check

PCDR FUNCTION CODES
Code Symbol Meaning/Arguments
27 .MPSI Enable software interrupt on nonfatal device conditions
0 AC3: Address of arbblk
1 0 Word count including this word
2 1 Interrupt channel number
3 Flags
4 BO(MO%MSG) Suppress CTY device messages

37 .MORST Read card reader status
AC3: Address of arbblk
0 0 Word count including this word
1 1 Status word
2 BO(MO%COL) Card reader is online
3 B1(MO%RLD) FE has been reloaded
4 B10(MO%FER) Fatal hardware error occurred
TOPS-20 Monitor Calls Quick Reference Guide

B12(MD%EDF) Card reader at EOF
B13(MD%IDP) Input in progress
B14(MD%SER) Software error occurred
B15(MD%HE) Hardware error occurred
B16(MD%OL) Reader is off-line
B17(MD%FX) Reader does not exist
B31(MD%FL) Output stacker full
B32(MD%EM) Input hopper empty
B33(MD%CK) Stack check
B34(MD%PCK) Pick check
B35(MD%RCK) Read check

PTY FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>.MOAPI</td>
<td>Assign PTY interrupt channels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: B0(MD%WFI) Enable waiting-for-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>input interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(MD%DIR) Enable output-is-ready</td>
</tr>
<tr>
<td></td>
<td></td>
<td>interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B12-17(MD%SIC) Interrupt channel for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTY output</td>
</tr>
<tr>
<td>25</td>
<td>.MDPIH</td>
<td>Determine if PTY job needs input</td>
</tr>
<tr>
<td>26</td>
<td>.MOBAT</td>
<td>Set batch control bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: 0 Job not under batch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Job under batch</td>
</tr>
</tbody>
</table>

TTY FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>.MDPIH</td>
<td>Determine if TTY job needs input</td>
</tr>
<tr>
<td>26</td>
<td>.MOSPD</td>
<td>Set terminal line speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: B0(MD%RMT) Remote line (WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(MD%AUT) Remote autobaud line (WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B35 Function code</td>
</tr>
<tr>
<td>27</td>
<td>.MORSP</td>
<td>Return terminal line speed</td>
</tr>
<tr>
<td>30</td>
<td>.MORLW</td>
<td>Return terminal page width</td>
</tr>
<tr>
<td>31</td>
<td>.MOSLW</td>
<td>Set terminal page width</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Page width</td>
</tr>
<tr>
<td>32</td>
<td>.MORLL</td>
<td>Return terminal page length</td>
</tr>
<tr>
<td>33</td>
<td>.MOSLL</td>
<td>Set terminal page length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Page length</td>
</tr>
<tr>
<td>34</td>
<td>.MOSNT</td>
<td>Set terminal receive-system-messages code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: 0(.MOSMY) Allow messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1(.MOSMN) Refuse messages</td>
</tr>
<tr>
<td>35</td>
<td>.MORNT</td>
<td>Return terminal receive-system-messages code</td>
</tr>
<tr>
<td>36</td>
<td>.MOSIG</td>
<td>Set terminal input on inactive line code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: 0 Do not ignore input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Ignore input</td>
</tr>
<tr>
<td>37</td>
<td>.MORBH</td>
<td>Read 128-character break mask</td>
</tr>
<tr>
<td>40</td>
<td>.MOSBM</td>
<td>Set 128-character break mask</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of arg11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 Word count not including this word</td>
</tr>
</tbody>
</table>

87
TOPS-20 Monitor Calls Quick Reference Guide
MTOPR

1-4 Break character mask
41   .MORFW Return current value of field width
42   .MOSFW Set field width
   AC3: Field width
43   .MOSDF Set pause-at-end-of-page mode
   AC3: 0(.MOSFF) Disable pause-at-end-of-page mode
   1(.MOSNF) Enable pause-at-end-of-page mode
44   .MORXO Read end-of-page mode
45   .MOSLC Set terminal's line counter
   AC3: Line counter value
46   .MORLX Read terminal's line counter
47   .MOSLM Set line maximum
   AC3: Line maximum value
50   .MORLM Read line maximum
48   .MOTPS Assign terminal interrupt channels
   AC3: Address of argblk
      0 Word count including this word
      1 B0-17 Output PSI channel
      818-35 Input PSI channel
52   .MDPCS Set terminal pause/unpause characters
   AC3: <pause character>,<unpause character>
53   .MDPCR Read terminal pause/unpause characters

MTUS % JUSYS 600

FUNCTION
Allows privileged programs to perform various utility functions for magnetic-tape MT: devices.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: MT unit number
AC3: Address of argblk

RETURNS  +1: Always

FUNCTION CODES
Code Symbol    Meaning/Arguments
  1 .MTNNV Declare volume switch error
      0 .MTCNT Word count
      1 .MTCOD Error code to return to user
  2 .MTRLA Read labels
      0 .MTCNT Word count
      1 .MTVL1 Byte pointer to area for VDL1 label
      2 .MTVL2 Byte pointer to area for VDL2 label
      3 .MTHD1 Byte pointer to area for HDR1 label
      4 .MTHD2 Byte pointer to area for HDR2 label
  3 .MTASI Return assignment information
TOPS-20 Monitor Calls Quick Reference Guide

0  .MTCNT Word count
1  .MTPHU Returned MTA # associated with MT
4  .MTCVV Clear volume ID for specified MT

**MULTI**  Jsys 512

FUNCTION
Performs various IPCF (Inter-Process Communication Facility) functions.

RESTRICTIONS
Some functions require WHEEL, OPERATOR, or IPCF capability enabled.

CALLING SEQUENCE
AC1: Length of argbk
AC2: Address of argbk

RETURNS
+1: Failure, error code in AC1
+2: Success, with requested data in argbk

ARGUMENT BLOCK
Word Contents
0  Function code
1-n Function-specific arguments

**FUNCTION CODES**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MUNEB</td>
<td>Enable specified PID to receive packets</td>
</tr>
<tr>
<td>2</td>
<td>MUDIS</td>
<td>Disable specified PID from receiving packets</td>
</tr>
<tr>
<td>3</td>
<td>MUGTI</td>
<td>Return PID associated with &lt;SYSTEM&gt;INFO</td>
</tr>
<tr>
<td>4</td>
<td>MUCPI</td>
<td>Create private copy of &lt;SYSTEM&gt;INFO for job (enabled IPCF)</td>
</tr>
<tr>
<td>5</td>
<td>MUDES</td>
<td>Delete specified PID</td>
</tr>
<tr>
<td>6</td>
<td>MUCRE</td>
<td>Create PID for specified process or job</td>
</tr>
<tr>
<td>7</td>
<td>MUSSQ</td>
<td>Set send/receive quotas for specified PID (enabled IPCF)</td>
</tr>
<tr>
<td>10</td>
<td>MUCHD</td>
<td>Change job # associated with specified PID (enabled WHL)</td>
</tr>
</tbody>
</table>

Flags, <process handle or job #>
B6(IP%JWP)  PID is job-wide
B7(IP%JGA)  PID is not available to other processes

1  PID
2  B18-26  New send quota
2  B27-35  New receive quota
1  PID
2  New job # or PID belonging to new job

89
TOPS-20 Monitor Calls Quick Reference Guide
MUTIL

11 .MUFDJ Return job # associated with specified PID
   1 PID

12 .MUFPJ Return all PIDs associated with specified job
   1 Job # or PID belonging to job

13 .MUFSQ Return send/receive quotas for specified PID
   1 PID

15 .MUFPQ Return all PIDs associated with same process
   as given PID
   1 PID

16 .MUSPQ Set maximum number of PIDs allowed for job
   (enabled IPCF)
   1 Job # or PID
   2 PID quota

17 .MUFPQ Return maximum number of PIDs allowed for job
   1 Job # or PID

20 .MUQRY Return Packet Descriptor Block for next
   packet in queue associated with specified PID
   1 PID
   2 -1 Next descriptor block for process
   -2 Next descriptor block for job

21 .MUAPF Associate PID with specified process
   1 PID
   2 Process handle

22 .MUPIC Place specified PID on software interrupt
   channel
   1 PID
   2 Channel number; -1 to remove PID

23 .MUQDI Set PID of <SYSTEM>INFO (enabled IPCF)
   1 PID of <SYSTEM>INFO

24 .MUSSP Place specified PID into system PID table at
   offset (enabled WHL/DPR/IPCF)
   1 Index into system PID table
   2 PID

25 .MUQSP Return PID from system PID table 1 Index into
   system PID table

26 .MUMPS Return system-wide maximum packet size

27 .MUSKP Set PID to receive deleted PID messages
   1 Source (subordinate) PID
   2 Object (controller) PID

30 .MURKP Return controlling PID for this subordinate
   PID
   1 Source (subordinate) PID
   2 Object (controller) PID (RET)

NIN JSYS 225

FUNCTION
Inputs an integer, with leading spaces ignored.

CALLING SEQUENCE
AC1: Source designator
AC3: Radix (2-10) of number being input

RETURNS +1: Failure, with
AC1: Updated byte pointer
AC3: Error code
+2: Success, with
AC1: Updated byte pointer
AC2: Number input

NODE JSYS 567

FUNCTION
Performs network utility functions.

RESTRICTIONS
Some functions require WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS +1: Always

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.NDSLN</td>
<td>Set local node name (WHL/OPR)</td>
</tr>
<tr>
<td>1</td>
<td>.NDGLN</td>
<td>Get local node name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDNOD Byte pointer to ASCIZ node name</td>
</tr>
<tr>
<td>2</td>
<td>.NDSNM</td>
<td>Set local node number (WHL/OPR)</td>
</tr>
<tr>
<td>3</td>
<td>.NDGNM</td>
<td>Get local node number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDNOD Local node number</td>
</tr>
<tr>
<td>4</td>
<td>.NDLP</td>
<td>Set loopback port (KS-10 only: WHL/OPR/MNT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDPRT NSP port number</td>
</tr>
<tr>
<td>5</td>
<td>.NDLP</td>
<td>Clear loopback port (KS-10 only: WHL/OPR/MNT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDPRT NSP port number</td>
</tr>
<tr>
<td>6</td>
<td>.NDLP</td>
<td>Find loopback port (KS-10 only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDPRT Flags, &lt;NSP port number&gt; (RET)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO(NDLPR) Loopback running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(NDLPA) Loopback port assigned</td>
</tr>
<tr>
<td>7</td>
<td>.NDSNT</td>
<td>Set node table (WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDNND # of nodes in topology message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .NDMSK Address of topology message</td>
</tr>
<tr>
<td>10</td>
<td>.NDGNT</td>
<td>Get node table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDNND 0 , &lt;word count&gt; (1/node)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On return</td>
</tr>
<tr>
<td></td>
<td></td>
<td># returned nodes , &lt;word count&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .NDCNT # of words in node block (RET)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 .NDBK1 Addresses of n node blocks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1/returned node; RET)</td>
</tr>
<tr>
<td>11</td>
<td>.NDSIC</td>
<td>Set topology interrupt channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .NDCHN Channel # for topology interrupts</td>
</tr>
<tr>
<td>12</td>
<td>.NDCIC</td>
<td>Clear topology interrupt channel</td>
</tr>
<tr>
<td>13</td>
<td>.NDGVR</td>
<td>Get NSP version number</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

NODE

0 .NDVRR Number of versions (RET)
1 .NDSVR Address of block for NSP communications version
2 .NDSVR Address of block for NSP routing version

14 .NDGI Get line information
0 .NDNLN 0,<word count> (1/line);
On return
<# returned lines>,<word count>
1 .NDBK1 Addresses of n line blocks
(1/returned line; RET)

15 .NDFVY Verify node name
0 .NDNOD Byte pointer to ASCIZ node name
1 .NDFLG Flags returned by monitor
BO(ND%EXM) Node exactly matches name in monitor's database

16 .NDRN Return node name
0 .NDNOD Node number
1 .NDCVR Byte pointer to string for ASCIZ node name

NODE BLOCK

Word Symbol Contents
0 .NDNAM Byte pointer to ASCIZ node name
1 .NDSTA Node state:
   .NDSON On, add to table of reachable nodes
   if not there
   .NDSOF Off, remove from table if there
2 .NDNXT Obsolete
3-4 -- ASCIZ node name (word 4 not returned if name
   .IE. 4 characters)

TOPOLOGY MESSAGE

| 4 3 2 1 10 7 6 5 14 13 12 11 20 17 16 15 |
| + + + + + + + + + + + + + + + + + + + |
| ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ |
| Byte 1 Byte 2 Byte 3 Byte 4 |
| Nodes 4-1 Nodes 10-5 Nodes 14-11 Nodes 20-15 |

Value Meaning
00 Node not reachable
01 Reserved
10 Reachable Phase II node
11 Reachable Phase III node

NSP VERSION BLOCK

Word Symbol Contents
0 .NDVRR Version number
1 .NDECO ECO number
2 .NDCST Customer change order
LINE BLOCK
Word Symbol Contents
0 .NDLNM Line number
1 .NDLST State of Line
   .NDLON On
   .NDLOF Off
   .NDLCN Controller loopback
   .NDLCB Cable loopback
   .NDLND Byte pointer to ASCIZ remote node name

NOUT JSYS 224

FUNCTION
Outputs an integer number.

CALLING SEQUENCE
AC1: Destination designator
AC2: Number to be output
AC3: B0(NO%MAG) Output magnitude only
     B1(NO%SGN) Output + before positive number
     B2(NO%LFL) Output leading filler
     B3(NO%ZRO) Output 0's as leading filler
     B4(NO%ODV) Output on column overflow and return an error
     B5(NO%AST) Output asterisks on column overflow
     B11-17(NO%COL) Number of columns to output
     B18-35(NO%RDX) Radix (2-36) of number being output

RETURNS  +1: Failure, error code in AC3
         +2: Success, updated byte pointer in AC1, if pertinent

NTMANX JSYS 604

FUNCTION
Provides an interface between the DECnet-20 Network Management layer and lower layers of the Digital Network Architecture.

RESTRICTIONS
Requires WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Address of argblk

RETURNS  +1 Always, with error code in AC1

ARGUMENT BLOCK
Word Symbol Contents
0 .NTCNT Word count including this word
1 .NTENT Entity on which to perform function
0 .NTMOD Node
1 .NTLIN Line
2 .NTLOG Logging
3 .NTCKT Circuit
4 .NTMOD Module

2 .NTEID Byte pointer to entity ID
3 .NTFNC Function code
   -2 .NTMAP Map node number/node name
   -1 .NTREX Return local node ID
   0 .NTSET Set Parameter
   1 .NTCLR Clear Parameter
   2 .NTZRO Zero all counters
   3 .NTSHO Show selected items
   4 .NTSZC Show and zero all counters
   5 .NTRET Return list of items

4 .NTSEL Selection criterion for function
   Selectors for .NTSHO
   0 .NTSUM Summary
   1 .NTSTA Status
   2 .NTCHA Characteristics
   3 .NTCOU Counters
   4 .NT_EVT Event
   Selectors for .NTRET
   -1 .NTKND Known items
   -2 .NTACT Active items
   -3 .NTLOOP Loop

5 .NTQUA Byte pointer to function qualifier
6 .NTBPT Byte pointer to parameter data buffer
7 .NTBYT Parameter data buffer length in bytes
   (functions .NTMAP, .NTRET, .NTREX, .NTSHO,
   and .NTSZC)
10 .NTERR Network management return code

ODOCNET SYS 222

FUNCTION
Converts internal date and time format into separate numbers
for local weekday, day, month, year, and time and does not
convert the numbers to text.

CALLING SEQUENCE
AC2: Internal date/time, or -1 for current date/time
AC4: B0(IC%DSA) Apply daylight savings according to
     B1(IC%ADS)
     B1(IC%ADS) Apply daylight savings if 1B0(IC%DSA)
     B2(IC%JTZ) Use time zone in B12-B17(IC%TMZ)
     B3(IC%JDJ) Apply Julian day format
     B12-17(IC%TMZ) Time zone to use if 1B2(IC%JTZ)

RETURNS +1: Always, with
     AC2: Year.,<numerical month> or
          Year,.<Julian day> if IC%JDJ
     AC3: <day of month>,<day of week> or
          0,.<day of week> if IC%JDJ

94
AC4: BO, B2  On for compatibility with IDCNV
       B1(IC%ADS)  If daylight savings was applied
       B3(IC%JUD)  If Julian day format was applied
       B12-17(IC%TMZ) Time zone used
       B18-35(IC%TIM) Local time in seconds since midnight

ODTIM   JSYS 220

FUNCTION
Converts the internal date and/or time to text.

CALLING SEQUENCE
AC1: Destination designator
AC2: Internal date/time, or -1 for current date/time
AC3: Format option flags; or
     0 for format: dd-mmm-yy hh:mm:ss; or
     -1 for format: weekday, month day, year hh:mm:ss

RETURNS  +1: Always, with updated byte pointer in AC1

FORMAT OPTION FLAGS
BO(OT%NDA)  Do not output date and ignore B1-8
B1(OT%DAY)  Output day of week according to B2(OT%FDY)
B2(OT%FDY)  Output full text for day of week
B3(OT%MMN)  Output month as numeric and ignore B4(OT%FMN)
B4(OT%FMN)  Output full text for month
B5(OT%4YR)  Output year as a 4-digit number
B6(OT%DAM)  Output day of month after month
B7(OT%SPA)  Output day month year with space delimiter;
             if 186(OT%DAM), output month day, year
B8(OT%SLA)  Output numeric date with slash delimiter; if 0B7 and 0B8, output day-month-year with dash delimiter
B9(OT%NTM)  Do not output time and ignore B10-13
B10(OT%NCSC) Do not output seconds
B11(OT%24H) Output time in 12-hour format with AM or PM
B12(OT%MCN) Output time without colon between hours and minutes
B13(OT%TMZ) Output time with "-" and time zone
B17(OT%SCL) Suppress columnization of date and time (omit leading spaces and zeros)
TPPS-20 Monitor Calls Quick Reference Guide

ODTNC

ODTNC USYS 230

FUNCTION
Outputs the date and/or the time as separate numbers for
local year, month, day, or time.

CALLING SEQUENCE
AC1: Destination designator
AC2: Year,<numerical month>
AC3: <day of month>,<day of week>
AC4: B1(ICSADS) Apply daylight savings on output
B12-17(ICSMTZ) Time zone desired
B18-35(ICSIM) Local time in seconds since midnight
AC5: Format option flags

RETURNS +1: Always, with updated byte pointer in AC1

FORMAT OPTION FLAGS
B0(OT%ND) Do not output date and ignore B1-8
B1(OT%DY) Output day of week according to B2(OT%FDY)
B2(OT%FDY) Output full text for day of week
B3(OT%MN) Output month as numeric and ignore B4(OT%FMN)
B4(OT%FMN) Output full text for month
B5(OT%YR) Output year as a 4-digit number
B6(OT%DM) Output day of month after month
B7(OT%SP) Output day month year with space delimiter;
   if 1B6(OT%DM), output month day, year
B8(OT%SL) Output numeric date with slash delimiter; if
   0B7 and 0B8, output day-month-year with dash
delimiter
B9(OT%TM) Do not output time and ignore B10-13
B10(OT%SC) Do not output seconds
B11(OT%TH) Output time in 12-hour format with AM or PM
B12(OT%NC) Output time without colon between hours and
   minutes
B13(OT%MZ) Output time with "-" and time zone (US zones
   and Greenwich Mean only)
B17(OT%SC) Suppress columnization of date and time (omit
   leading spaces and zeros)

OPENF USYS 21

FUNCTION
Opens the given file.

CALLING SEQUENCE
AC1: 0, UFN
AC2: B0-S(OF%BSZ) Byte size (maximum of 36.; 36. default)
B6-9(OF%MOD) Data mode in which to open file
B10(OF%HER) Halt on I/O, device, or data error
B19(OF%RD) Allow read access
B20(OF%WR) Allow write access
B21(OF%EX) Allow execute access
B22(OF%APP) Allow append access
B23(OF%RDU)  Allow unrestricted read access (illegal with OF%THW or OF%WR)
B25(OF%THW)  Allow thawed access
B26(OF%AWT)  Block and wait for access to be granted
B27(OF%PDT)  Do not update access dates of file
B28(OF%AWT)  Do not wait if access disallowed; return error
B29(OF%RTD)  Enforce restricted access
B30(OF%PLN)  Disable line number checking
B31(OF%ODU)  Suppress system updating of modified pages in memory to thawed files on disk unless CLOSF or UFPGS issued
B32(OF%OLI)  Open device even if off-line
B33(OF%FDT)  Force update of .FBREF (last read) in FDS and increment RH of .FBCNT (number of references)
B34(OF%RAR)  Wait if file off-line

RETURNS  +1: Failure, error code in AC1
          +2: Success

PBIN  JSYS 73

FUNCTION
Inputs the next sequential byte from the primary input designator.

RETURNS  +1: Always, with the byte R-U in AC1

PBOUT  JSYS 74

FUNCTION
Outputs a byte sequentially to the primary output designator.

CALLING SEQUENCE
AC1: Byte to be output, right-justified

RETURNS  +1: Always

PDVOPK  JSYS 603

FUNCTION
Manipulates program data vectors (PDVs), using program data vector addresses (PDVAs).

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk
AC3: Byte pointer to string in memory
TOPS-20 Monitor Calls Quick Reference Guide
PDV0P%

RETURNS +1: Always, with data returned in the data block, and updated count in .PDCT2 if needed

FUNCTION CODES
Code  Symbol    Meaning
0    .PDGET   Return all PDVAs within range specified in argbk
1    .PDADD   Add PDVAs specified in data block to system's database for process
2    .PDEREM   Remove PDVAs within range specified in argbk from system's data base for process
3    .PDNAM   Return ASCIZ name of program referenced in word .PVNAM of PDV
4    .PDVER   Return program version number from word .PVVER of PDV
5    .POLOC   Return all PDVAs of PDVs for program referenced in AC3

ARGUMENT BLOCK
Word  Symbol   Meaning
0    .PDCT1   Number of words in argbk
1    .PDPHD   Handle of desired process
2    .PDCT2   # of words in data block; on return
3    .PDAT    Starting address of data block for returned data
4    .POADDR   Starting address of memory range
5    .POADE   Ending address of memory range

FORMAT OF PROGRAM DATA VECTOR
Word  Symbol   Meaning
0    .PVCNT   Length of PDV including this word
1    .PVNAM   Pointer to ASCIZ program name string for this PDV
2    .PVSTR   Program starting address
3    .PVREE   Program reenter address
4    .PVVER   Program version number
5    .PVMEM   Address of block of memory containing length in Word 0 and memory map in remaining words
6    .PVSYM   Address of program symbol table
7    .PVCTM   Time at which program was compiled
10   .PVCR   Version number of compiler
11   .PVLTM   Time at which program was loaded
12   .PVLR   Version number of LINK
13   .PVMON   Address of monitor data block (not used currently)
14   .PVPRG   Address of program data block (not used currently)
15   .PV CST  Address of customer-defined data block

98
PEEK  JSYS 311

FUNCTION
Transfers a block of words from the monitor to the user space.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: <word count>,<1st virtual address of monitor>
AC2: 1st user address

RETURNS
+1: Failure, error code in AC1
+2: Success

PLOCK  JSYS 561

FUNCTION
Locks physical memory and places a designated section of the process' address space in memory.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capabilities.

CALLING SEQUENCE
AC1: Address of 1st page if locking; -1 if unlocking
AC2: <process handle>,<# of 1st page>
AC3: <control flags>,<repeat count>
   B0(LK%CNT)  B18-35 of AC3 contain # of pages to lock
   B1(LK%PHY)  AC1 contains 1st page desired
   B2(LK%NC)  Pages will not be cached
   B3(LK%OL)  Off-line pages are to be locked

RETURNS
+1: Always

PMAP  JSYS 56

FUNCTION
Maps one or more complete pages from a file to a process (Case I), from a process to a file (Case II), or from one process to another process (Case III); or unmaps pages from a process (Case IV) and deletes pages from a file (Case V).

CALLING SEQUENCE
AC1: uFN,<file page #> (Case I)
     <source process handle>,<process page #> (Cases II & III)
-1 (Cases IV & V)
AC2: <destination process handle>,<process page #> (Cases I & III)
TOPS-20 Monitor Calls Quick Reference Guide

PMAP

<destination JFN>,<file page #> (Case II)  
JFN,<file page #> (Case IV)

AC3:  
B0(PM%CNT)  B18-35 contain repeat count
B2(PM%RD)  Permit read access (Cases I - III only)
B3(PM%WR)  Permit write access (Cases I - III only)
B4(PM%EX)  Reserved
B5(PM%PLD)  Preload page being mapped (Cases I - III only)
B9(PM%CPY)  Create private copy of page (Cases I - III only)
B10(PM%EPN)  B18-35 of AC2 contain extended (18-bit) process page number (Cases I - III only)
B11(PM%ABT)  Unmap page and discard changed contents (Cases I - III only)
B18-35(PM%RPT) # of pages to map if 1B0(PM%CNT)

RETURNS +1: Always

PMCTL JSYS 860

FUNCTION
Controls physical memory, allowing a privileged program to add or remove most pages of physical memory and to control use of cache memory.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Length of argblk
AC3: Address of argblk

RETURNS +1: Always

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.MCRCE</td>
<td>Return status of cache memory</td>
</tr>
<tr>
<td>1</td>
<td>.MCSCCE</td>
<td>Set status of cache memory</td>
</tr>
<tr>
<td>2</td>
<td>.MCRPS</td>
<td>Return status of specified page</td>
</tr>
<tr>
<td>3</td>
<td>.MCSPS</td>
<td>Set status of specified page</td>
</tr>
</tbody>
</table>

 Meaning:
 - .MCCE: Return status of cache memory
 - .MCSCCE: Set status of cache memory
 - .MCRPS: Return status of specified page
 - .MCSPS: Set status of specified page

Arguments:
 - 0: MCPNN - count,.<physical page #>
 - 1: MCPST Returned page status
 - 0: MCPSA Page available
 - 1: MCPSS Page in transition
 - 2: MCPSO Page off-line (nonexistent)
 - 3: MCPSE Page off-line due to error
TOPS-20 Monitor Calls Quick Reference Guide
PMCTL

0 .MCPPN Physical page number
1 .MCPST Status for page
   0 .MCPSA Mark page available
   1 .MCPSI Mark page in transition
   2 .MCPSO Mark page off line
      (nonexistent)
   3 .MCPSL Mark page off line due
to error

4 .MCRM Return information about MDS memory errors
   0 .PMTMTP <BB1!<count>B17>,<controller #>
   1 .PMRSE Error register at error
   2 .PMMSY Syndrome of error
   3 .PMMBN Block number of error
   4 .PMMSB Spare bit number
   5 .PMMEA Error address
   6 .PMMSN 4 words of 32-bit PROM serial
      numbers

PPNST .SYS 557

FUNCTION
Translates a project-programmer number (PPN, a TOPS-10
36-bit directory designator) to its corresponding TOPS-20
string.

CALLING SEQUENCE
AC1: Destination designator
AC2: Project-programmer number (36-bit)
AC3: Byte pointer to structure name string for which given
PPN applies

RETURNS +1: Always, with updated byte pointer in AC1

PRARG .SYS 545

FUNCTION
Returns or sets up an argument block for the specified
process.

CALLING SEQUENCE
AC1: <function code>,<process handle>
AC2: Address of argblk
AC3: Length of argblk

RETURNS +1: Always, with number of returned words in AC3

FUNCTION CODES
Code Symbol Meaning
 1 .PRAD Return arguments in argblk
 2 .PRAST Set arguments from argblk

ARGUMENT BLOCK

101
TDPS-20 Monitor Calls Quick Reference Guide
PRARG

<table>
<thead>
<tr>
<th>Word</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Number of argblks</td>
</tr>
<tr>
<td>1 - n</td>
<td>Argument pointers</td>
</tr>
<tr>
<td>n+1</td>
<td>Data</td>
</tr>
</tbody>
</table>

**ARGUMENT POINTER**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>1</td>
</tr>
<tr>
<td>B1-3</td>
<td>0</td>
</tr>
<tr>
<td>B4-6</td>
<td>Data structure type</td>
</tr>
<tr>
<td>0</td>
<td>Scalar or array without dope vector</td>
</tr>
<tr>
<td>1</td>
<td>Array with dope vector</td>
</tr>
<tr>
<td>3</td>
<td>Immediate (data in B18-35)</td>
</tr>
<tr>
<td>B7-12</td>
<td>Type code</td>
</tr>
<tr>
<td>00</td>
<td>Unspecified</td>
</tr>
<tr>
<td>02</td>
<td>Integer</td>
</tr>
<tr>
<td>04</td>
<td>Real</td>
</tr>
<tr>
<td>17</td>
<td>ASCIZ string</td>
</tr>
<tr>
<td>B13-17</td>
<td>0</td>
</tr>
<tr>
<td>B18-35</td>
<td>Data offset in block or data; -1 for last LOAD-class command</td>
</tr>
</tbody>
</table>

**PSOUT** JSYS 76

FUNCTION
Outputs a string sequentially to the primary output designator.

**CALLING SEQUENCE**
AC1: Byte pointer to ASCIZ string

**RETURNS** +1: Always, with updated byte pointer in AC1

**RCDIR** JSYS 553

FUNCTION
Translates the given directory string to its corresponding 36-bit directory number.

**RESTRICTIONS**
In non-zero sections, OWGBPs must specify 7-bit bytes.

**CALLING SEQUENCE**
AC1: <flag bits>, 0
AC2: Byte pointer to ASCIZ string (to obtain 36-bit directory number)  
JFN (to obtain directory number associated with file)  
36-bit user number (to obtain logged-in directory)  
36-bit directory number (to check validity)
AC3: 36-bit directory number (to use RCDIR to step through directory string with wildcards)
RETURNS +1: Always, with
AC1:  <flag bits>.,0
AC2:  Updated byte pointer (if pointer was supplied)
AC3:  36-bit directory number

FLAGS SUPPLIED IN RCDIR CALL
Bit  Symbol  Meaning
B14  RC%PAR  Allow partial recognition on directory name
B15  RC%STP  Step to next directory in group and return number
B16  RC%AWL  Allow directory name to contain wildcards
B17  RC%EMO  Match given string exactly

FLAGS RETURNED FROM RCDIR CALL
Bit  Symbol  Meaning
B0   RC%DIR  Directory is files-only
B1   RC%ANA  Obsolete
B2   RC%RLM  User sees all messages from <SYSTEM>MAIL.TXT on login
B3   RC%NOM  No match was found for string
B4   RC%AMB  String given was ambiguous
B5   RC%NDI  No more directories in group
B6   RC%WLD  Directory name contained wildcards

RCM  JSYS 134

FUNCTION
Returns the word mask of the activated interrupt channels
for the specified process.

CALLING SEQUENCE
AC1:  Process handle

RETURNS +1: Always, with
AC1:  36-bit word (1Bn indicates channel n activated)

RCUSR  JSYS 554

FUNCTION
Translates the given user name string to its corresponding
36-bit user number.

RESTRICTIONS
Directory may not be files-only.

CALLING SEQUENCE
AC1:  <flag bits>.,0
AC2:  Byte pointer to ASCII username string
AC3:  36-bit user number (if stepping to next username in group)
RETURNS +1: Always, with
AC1: <flag bits>, 0
AC2: Updated byte pointer
AC3: 36-bit user number

FLAGS SUPPLIED ON CALL
Bit Symbol Meaning
B14 RC%PAR Allow partial recognition on username string
B15 RC%XTP Step to next username in group
B16 RC%AWL Allow username to contain wildcards
B17 RC%EMO Match given string exactly

FLAGS RETURNED FROM CALL
Bit Symbol Meaning
B1 RC%ANA Obsolete
B2 RC%RLM User sees all messages from <SYSTEM>MAIL.TXT on login
B3 RC%NOM No match was found for string
B4 RC%AMB String given was ambiguous
B5 RC%NMD No more usernames in group
B6 RC%WLD Username given contained wildcards

RCVIM JSYS 751

FUNCTION
Retrieves a message from the ARPANET special message queue.

RESTRICTIONS
For ARPANET systems only.

CALLING SEQUENCE
AC1: B0 If on, leader is 96-bit; if off, leader is 32-bit
B1 If on, 32-bit data in each word of message (high-order); if off, 36-bit data in each word
B18-35 Special queue header
AC2: Address for storing extended message

RETURNS +1: Failure, error code in AC1
+2: Success

RCVOK% JSYS 575

FUNCTION
Allows installation-supplied access-control program to service an approval request in the GETOK% request queue.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.
TOPS-20 Monitor Calls Quick Reference Guide

CALLING SEQUENCE
AC1: Address of argblk
AC2: Length of argblk

RETURNS
+1: Always

ARGUMENT BLOCK (RET)

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.RFCFJ</td>
<td>&lt;GETOK% function code&gt;,&lt;job # of requestor&gt;</td>
</tr>
<tr>
<td>1</td>
<td>.RCUNO</td>
<td>User number</td>
</tr>
<tr>
<td>2</td>
<td>.RCCDR</td>
<td>Connected directory</td>
</tr>
<tr>
<td>3</td>
<td>.RCQDN</td>
<td>Request number</td>
</tr>
<tr>
<td>4</td>
<td>.RCNUA</td>
<td>B0-17 # arguments passed to RCVOK% block</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 # user arguments in user block</td>
</tr>
<tr>
<td>5</td>
<td>.RCARA</td>
<td>Address of user arguments</td>
</tr>
<tr>
<td>6</td>
<td>.RCCAP</td>
<td>Capabilities enabled</td>
</tr>
<tr>
<td>7</td>
<td>.RCTER</td>
<td>Controlling terminal number; or -1 for detached job</td>
</tr>
<tr>
<td>10</td>
<td>.RCRJB</td>
<td>Requested job #</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>User arguments</td>
</tr>
</tbody>
</table>

RDTY JSYS 523

FUNCTION

Reads input from the primary input designator into the caller's address space.

CALLING SEQUENCE
AC1: Byte pointer to string for input
AC2: <flag bits>,<# of bytes in string>

0,0,# of bytes in string> to break on EDL only
B0(RD%BRK) Break on CTRL/Z or ESC
B1(RD%TOP) Break on CTRL/G, CTRL/L, CTRL/Z, ESC, CR, LF
B2(RD%PUN) Break on punctuation:
CTRL/A-CTRL/F CTRL/H-CTRL/I CTRL/K
CTRL/N-CTRL/Q CTRL/S-CTRL/T
CTRL/X-CTRL/Y
ASCII codes 34-36, 40-57, 72-100, 133-140, 173-176
B3(RD%BEL) Break on EDL (CRLF or LF only)
B4(RD%CRF) Suppress CR and return LF only
B5(RD%RND) Return if attempt to delete past beginning of input buffer
B7(RD%RIE) Return if input buffer empty
B8(RD%SEG) Return if attempt to edit past beginning of input buffer
B10(RD%RAI) Convert lowercase input to uppercase
B11(RD%SUI) Suppress CTRL/U indication
AC3: Byte pointer to CTRL/R buffer; 0 if no reprompt text

RETURNS
+1: Failure, error code in AC1
+2: Success, with
AC1: Updated byte pointer
AC2: <flag bits>, <updated byte count>
B12(RD%BTM) Break character terminated input
B13(RD%BFE) Input buffer empty
B14(RD%BLR) Backup limit for editing reached

RELD JSYS 71

FUNCTION
Releases one or all devices assigned to the job.

CALLING SEQUENCE
AC1: Device designator; -1 to release all assigned devices
     devices assigned to this job

RETURNS
+1: Failure, error code in AC1
   +2: Success

RELSQ JSYS 753

FUNCTION
Deassigns the ARPANET special message queue, and discards
all pending messages.

RESTRICTIONS
For ARPANET systems only.

CALLING SEQUENCE
AC1: Special queue handle (RET by ASNSQ); -1 to deassign
     all special queues

RETURNS
+1: Always

RESET JSYS 147

FUNCTION
Closes all files at or below the current process and
releases all JFNs; kills all inferior processes; clears the
PSI for the current process; sets TT%WAK, TT%ECO, and .TTASI
of the controlling terminal's JFN mode word; releases all
PIDs of the current process; dequeues all ENQ requests for
the current process, clears PA1050's entry vector; and,
releases all process handles inferior to the current process
or killed with KFORK.

RETURNS
+1: Always
RFACS  JSYS 161

FUNCTION
Returns the ACs of the specified process.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of 20-word block to store AC values of specified process

RETURNS  +1: Always

RFBSZ  JSYS 45

FUNCTION
Returns the byte size for a specific opening of a file.

CALLING SEQUENCE
AC1:  JFN

RETURNS  +1: Failure, error code in AC1
         +2: Success, byte size R-J in AC2

RFCDC  JSYS 112

FUNCTION
Returns the control character output control (CCDC) words for the specified terminal.

CALLING SEQUENCE
AC1:  File designator

RETURNS  +1: Always, with CCDC words in AC2 and AC3

RFMOD  JSYS 107

FUNCTION
Returns the JFN mode word associated with the specified file.

CALLING SEQUENCE
AC1:  Source designator

RETURNS  +1: Always, with mode word in AC2
RFORK  JSYS 155

FUNCTION
Resumes one or more processes that have been directly frozen.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always

RFPOS  JSYS 111

FUNCTION
Returns the current position of the specified terminal's cursor.

CALLING SEQUENCE
AC1: Device designator

RETURNS +1: Always, with
AC2: <line number>,<column number>
0 if designator not terminal

RFPTR  JSYS 43

FUNCTION
Returns the current position of the specified file's pointer.

CALLING SEQUENCE
AC1: JFN

RETURNS +1: Failure, error code in AC1
+2: Success, byte number in AC2

RFRKH  JSYS 165

FUNCTION
 Releases the specified process handle if the process is inferior to at least one other process in the job or has been killed with KFORK.

CALLING SEQUENCE
AC1: Process handle; -1 for all

RETURNS +1: Failure, error code in AC1
+2: Success
FUNCTION
Returns the status of the specified process.

CALLING SEQUENCE
AC1: 0.,<process handle> (short form)
      flags.,<process handle> (long form)
B0  RF%LNG Long form call
   B1-17 Unused, must be zero
AC2: Address of status return block (long form only)

RETURNS +1: Always, with
   AC1: Status word (short form only)
   AC2: Process PC flags (short form only)
   AC3: -1 if process deleted (short form only)

PROCESS STATUS WORD
Bit Symbol Meaning
0  RF%FRZ Process is frozen
B1-17 RF%STS Status code for process
   0  .RFRUN Process is runnable
   1  .RFIO Process is dismissed for I/O
   2  .RFHLT Process dismissed by HFORK or
      HALTF or never started
   3  .RFFPT Process dismissed by forced
      process termination
   4  .RFWAT Process dismissed waiting for
      another process to terminate
   5  .RFSLP Process dismissed for specified
      amount of time
   6  .RFTRP Process dismissed because
      intercepted by superior
   7  .RFABK Process dismissed because address
      break encountered

B18-35 RF%SIC Number of software interrupt channel causing
forced process termination

STATUS-RETURN BLOCK (Long Form Only)
Word Symbol Meaning
0  .RFCNT <returned word count>,<max. words desired>
      (RH user specified)
1  .RFPBW Process status word; -1 if unassigned process
       handle in AC1
2  .RFPFL Process PC flags
3  .RFPFC Process PC
4  .RFSFL Status flag word
   BO  RF%EXO Process is execute-only
TOPS-20 Monitor Calls Quick Reference Guide
RFTAD

RFTAD  JSYS 533

FUNCTION
Returns the dates and times associated with the specified file.

CALLING SEQUENCE
AC1: Source designator
AC2: Address of argblk
AC3: Length of argblk

RETURNS  +1: Always, with dates returned in argblk

ARGUMENT BLOCK
Word  Symbol  Meaning
0  .RSWRT  Internal date and time file was last written
1  .RSCRV  Internal date and time file was created
2  .RSREF  Internal date and time file was last referenced
3  .RSCRE  System date and time of last write by monitor
4  .RSTD  Tape-write date and time for archived or migrated files
5  .RSNET  Online expiration date and time
6  .RSFET  Offline expiration date and time

RIN  JSYS 54

FUNCTION
Inputs a non-sequential (random) byte from the specified file.

RESTRICTIONS
Disk file only.

CALLING SEQUENCE
AC1: JFN
AC3: Byte number within file

RETURNS  +1: Always, with byte R-U in AC2; 0 if EDF

RIR  JSYS 144

FUNCTION
Returns the channel and priority level table addresses for the specified process.

RESTRICTIONS
Process must run in section zero; for multiple-section processes use XRIR%.
CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC2: <LEVTAB address>,<CHNTAB address>
     0 if no SIR issued for process

RIRCM  JSYS 143

FUNCTION
Returns the mask for reserved software interrupt channels
for the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with channel mask in AC2

RLJFN  JSYS 23

FUNCTION
Releases the specified closed JFNs belonging to the current
process or its inferiors.

CALLING SEQUENCE
AC1: JFN; -1 for all JFNs

RETURNS +1: Failure, error code in AC1
     +2: Success

RMAP  JSYS 61

FUNCTION
Acquires a handle on a page in a process to determine the
access allowed for that page.

CALLING SEQUENCE
AC1: <process handle>,<page # within process>

RETURNS +1: Always, with
AC1: <process/file designator>,<page #>
     -1 if page does not exist
AC2: Access bits; 0 if page does not exist
     B2(RM%RD) Read access allowed
     B3(RM%WR) Write access allowed
     B4(RM%EX) Execute access allowed
     B5(RM%PEX) Page exists
     B9(RM%CPY) Copy-on-write access allowed
RNAMF  JSYS 35

FUNCTION
Renames an existing file.

CALLING SEQUENCE
AC1:  Source file JFN
AC2:  Destination file JFN

RETURNS  +1:  Failure, error code in AC1
          +2:  Success, JFN in AC1 is released, and JFN in
               AC2 is associated with file under its new
               filespec

ROUT  JSYS 55

FUNCTION
Outputs a byte nonsequentially (randomly) to the specified
file.

RESTRICTIONS
For disk files only.

CALLING SEQUENCE
AC1:  JFN
AC2:  Byte to be output, right-justified
AC3:  Destination byte number within file

RETURNS  +1:  Always

RPACS  JSYS 57

FUNCTION
Returns the accessibility of a page.

CALLING SEQUENCE
AC1:  <process/file designator>,<process/file page number>

RETURNS  +1:  Always, with
          AC2:  Flags
              B2(PA%RD)  Read access allowed
              B3(PA%WT)  Write access allowed
              B4(PA%EX)  Execute access allowed
              B5(PA%PEX) Page exists
              B6(PA%IND) Indirect pointer
              B9(PA%CPY) Copy-on-write
              B10(PA%PRV) Private page
              B20(P1%RD) Read access allowed in 1st
                        pointer
              B21(P1%WT) Write access allowed in
                        1st pointer
              B22(P1%EX) Execute access allowed in

RPCAP JSYS 150

FUNCTION
Returns the capabilities for the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
  AC2: Capabilities possible for this process
  AC3: Capabilities enabled for this process

RSCAN JSYS 500

FUNCTION
Places a text string in, or reads a text string from, the job's rescan buffer (an area of storage in the Job Storage Block).

CALLING SEQUENCE
AC1: Byte pointer to new text string (1st call, to store string)
  0,,<function code> (2nd call, to read string)

RETURNS +1: Failure, error code in AC1
  +2: Success, with
  AC1: Updated pointer if one supplied, or
  Count of characters in rescan buffer, or O if rescan buffer empty

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>RSINI</td>
<td>Make rescan buffer available for input</td>
</tr>
<tr>
<td>1</td>
<td>RSCNT</td>
<td>Return count of characters remaining in rescan buffer</td>
</tr>
</tbody>
</table>

RSMAP% JSYS 610

FUNCTION
Reads a section map, and provides information about the mapping of one section of a fork's memory.

CALLING SEQUENCE
AC1: <fork handle>,,<section number>
RETURNS +1: Always, with
AC1: -1 if no current mapping:
   0 if mapping in private section;
   <fork handle>..<section #> if indirect
   or shared mapping to another fork's
   section; or JFN..<section #> if
   shared mapping to file section
AC2: Access bits
   B2(SMXRD) Read access allowed
   B3(SMXWR) Write access allowed
   B4(SMXEX) Execute access allowed
   B5(PAXPEx) Section exists
   B6(SMXIND) Section created using
               indirect pointer

RTFRK J SYS 322

FUNCTION
Returns the handle of a process that was suspended because
of a monitor call intercept and the monitor call that the
process was attempting to execute.

RETURNS +1: Always, with
AC1: Handle of process that generated
     interrupt
AC2: J SYS instruction that caused process
     suspension

RTIW J SYS 173

FUNCTION
Reads the terminal interrupt word for the specified process
or the entire job, and returns the terminal interrupt word
mask.

AC1: B0(RT%DIM) Return mask for deferred terminal
     interrupts
   B18-35(RT%PRH) Process handle, or -5 for entire job

RETURNS +1: Always, with
AC2: Terminal interrupt mask
AC3: Deferred terminal interrupt mask
FUNCTION
Returns the run time of the specified process or of the entire job.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: Runtime (in mss) right-justified
AC2: Divisor to convert mss to sec (1000)
AC3: Console time (in mss)

FUNCTION
Returns the word mask for the interrupts waiting on software channels for the specified process.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC1: 36-bit word (1Bn indicates pending interrupt on channel n)
AC2: Status of interrupts in progress (1Bn in LH indicates priority level n interrupt occurred in user code; 1B(18+n) in RH indicates priority level n interrupt occurred in monitor code)

FUNCTION
Releases the working set by removing all of the current process’s pages from its working set.

RETURNS +1: Always

FUNCTION
Sets the account to which the specified file is to be charged.

RESTRICTIONS
In non-zero sections, DWGBP’s must specify 7-bit bytes.
TOPS-20 Monitor Calls Quick Reference Guide
SACTF

CALLING SEQUENCE
AC1: JFN
AC2: <5821<account number>B35>; or byte pointer to account string (maximum 39 characters)

RETURNS +1: Failure, error code in AC1
+2: Success, updated byte pointer in AC2

SAVE  JSYS 202

FUNCTION
Saves, in nonsharable format, pages of a process in the specified file.

RESTRICTIONS
Legal for single-section processes only.

CALLING SEQUENCE
AC1: <process handle>,JFN
AC2: table entry; or 0,.<table pointer>

RETURNS +1: Always

TABLE FORMAT
Word  Contents
0 to n <length of save area>,<address of 1st word to save>
 n+1  0

SCTTY  JSYS 324

FUNCTION
Redefines the controlling terminal for the specified process and all of its inferiors.

RESTRICTIONS
Requires SC%SCT capability enabled in the process capability word for some functions; cannot be used to change the job’s controlling terminal or the controlling terminal of the current process or its superiors.

CALLING SEQUENCE
AC1: <function code>,<process handle>
AC2: Terminal designator

RETURNS +1: Always

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SCRT</td>
<td>Return designator of given process’s controlling terminal in AC2</td>
</tr>
<tr>
<td>1</td>
<td>.SCSET</td>
<td>Change given process’s (and inferiors) controlling terminal to terminal designated</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

SCTTY

in AC2 (SC%SCT)

<table>
<thead>
<tr>
<th>.SCRST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset given process's (and inferiors) controlling terminal to job's controlling terminal (SC%SCT)</td>
</tr>
</tbody>
</table>

**SCVEC, JSYS 301**

FUNCTION
Sets the entry vector and the UUO locations for the compatibility package.

CALLING SEQUENCE
AC1: Process handle
AC2: <entry vector length>,<entry vector address>; 0 to merge compatibility package into caller's address space; or -1 to disable UUO simulation
AC3: <UUO location>,<PC location>

RETURNS +1: Always

**COMPATIBILITY PACKAGE'S ENTRY VECTOR**

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SVEAD</td>
<td>Entry address for interpreting UUOs</td>
</tr>
<tr>
<td>1</td>
<td>.SVINE</td>
<td>Initial entry for setup and first UUO</td>
</tr>
<tr>
<td>2</td>
<td>.SVGET</td>
<td>Entry for GET share file routine (obsolete)</td>
</tr>
<tr>
<td>3</td>
<td>.SV40</td>
<td>Address to receive contents of location 40 on UUO call</td>
</tr>
<tr>
<td>4</td>
<td>.SVRPC</td>
<td>Address to receive return PC word on UUO call</td>
</tr>
<tr>
<td>5</td>
<td>.SVMAK</td>
<td>Entry for MAKE share file routine (obsolete)</td>
</tr>
<tr>
<td>6-7</td>
<td>.SVCST</td>
<td>2 word block for handling CTRL/C, START sequences between compatibility package and TOPS-20 Command Processor</td>
</tr>
</tbody>
</table>

**SDSTS, JSYS 146**

FUNCTION
Sets the status of a device.

RESTRICTIONS
No-op for devices that do not have device-dependent status bits.

CALLING SEQUENCE
AC1: JFN
AC2: New status bits

RETURNS +1: Always
TOPS-20 Monitor Calls Quick Reference Guide
SDVEC

SDVEC  JSYS 543
FUNCTION
Sets the entry vector for the Record Management System (RMS).

RESTRICTIONS
Requires RMS software (currently available only with BASIC and COBOL).

CALLING SEQUENCE
AC1: process handle
AC2: <entry vector length>,<entry vector address>

RETURNS  +1: Always

RECORD MANAGEMENT SYSTEM'S ENTRY VECTOR
Word Symbol  Meaning
0 .SDEAD  Entry address for RMS calls
1 .SDINE  Initial entry for first RMS call
2 .SDVER  Pointer to RMS version block
3 .SDMS   Address in which to store RMS call
4 .SDRPC  Address in which to store return PC word

SETER  JSYS 336
FUNCTION
Sets the most recent error condition encountered by a process, stores it in the Process Storage Block.

CALLING SEQUENCE
AC1: Process handle
AC2: Error code to set

RETURNS  +1: Always

SETJB  JSYS 541
FUNCTION
Sets job parameters for the specified job.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability to set parameters for other than current job.

CALLING SEQUENCE
AC1: Jobno, or -1 for current job
AC2: Function code
AC3: Function value

RETURNS  +1: Always
FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function/Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SU DEN</td>
<td>Set default magtape density</td>
</tr>
<tr>
<td>1</td>
<td>.SUP AR</td>
<td>Set default for magtape parity</td>
</tr>
<tr>
<td>0</td>
<td>.SUPRO</td>
<td>Odd parity</td>
</tr>
<tr>
<td>1</td>
<td>.SUPRE</td>
<td>Even parity</td>
</tr>
<tr>
<td>2</td>
<td>.SJ DM</td>
<td>Set default for magtape data mode</td>
</tr>
<tr>
<td>3</td>
<td>.SUR S</td>
<td>Set default for magtape record size (in bytes)</td>
</tr>
<tr>
<td>4</td>
<td>.SJ DFS</td>
<td>Set spooling mode</td>
</tr>
<tr>
<td>0</td>
<td>.SU SPI</td>
<td>Immediate mode spooling</td>
</tr>
<tr>
<td>1</td>
<td>.SUSPD</td>
<td>Deferred mode spooling</td>
</tr>
<tr>
<td>5</td>
<td>.SUSRM</td>
<td>Set remark for current job session; pointer to remark in AC3</td>
</tr>
<tr>
<td>6</td>
<td>.SJT2O</td>
<td>Indicate if job is at EXEC or program level</td>
</tr>
<tr>
<td>-1</td>
<td>Job is at EXEC level</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Job is at program level</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.SU DFR</td>
<td>Set job default retrieval</td>
</tr>
<tr>
<td>0</td>
<td>.SURFA</td>
<td>OPENF of off-line disk file should fail (default)</td>
</tr>
<tr>
<td>1</td>
<td>.SURWA</td>
<td>OPENF of off-line disk file should wait for restoral</td>
</tr>
<tr>
<td>10</td>
<td>.SUBAT</td>
<td>Set batch flags and batch stream number</td>
</tr>
<tr>
<td>B0-1(0B%WTO)</td>
<td>Write to operator capability</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>.OBALL</td>
<td>WTO &amp; WTOR allowed</td>
</tr>
<tr>
<td>1</td>
<td>.OBNRW</td>
<td>No WTR allowed</td>
</tr>
<tr>
<td>2</td>
<td>.OBNDM</td>
<td>No message allowed</td>
</tr>
<tr>
<td>B10(0B%BSN)</td>
<td>0B%BSN contains batch stream #</td>
<td></td>
</tr>
<tr>
<td>B11-17(0B%BSN)</td>
<td>Batch stream #</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>.SULLO</td>
<td>Set job logical location (node name)</td>
</tr>
</tbody>
</table>

SETNM JSYS 210

FUNCTION
Sets the private name of the program being used by the current job.

CALLING SEQUENCE
AC1: SIXBIT name used to identify program

RETURNS +1: Always

SETSN JSYS 506

FUNCTION
Sets either the system name or the private name of the program being used by the current job.

CALLING SEQUENCE
AC1: SIXBIT name to be used as system name
AC2: SIXBIT name to be used as private name
TDPS-20 Monitor Calls Quick Reference Guide
\[\text{SETSN}\]

RETURNS  
\begin{enumerate}
\item +1: Failure
\item +2: Success
\end{enumerate}

\[\text{SEVEC} \quad \text{JSYS} \quad 204\]

FUNCTION  
Sets the entry vector of the specified process.

RESTRICTIONS  
The process must run in only one section of memory.

CALLING SEQUENCE  
\begin{enumerate}
\item AC1: Process handle
\item AC2: <entry vector length>,<entry vector address>; or 0 to remove entry vector
\end{enumerate}

RETURNS  
\begin{enumerate}
\item +1: Always
\end{enumerate}

\[\text{SFACS} \quad \text{JSYS} \quad 160\]

FUNCTION  
Sets the ACs of the specified process.

CALLING SEQUENCE  
\begin{enumerate}
\item AC1: Process handle
\item AC2: Address of 20 word block containing new AC values for process
\end{enumerate}

RETURNS  
\begin{enumerate}
\item +1: Always
\end{enumerate}

\[\text{SFBSZ} \quad \text{JSYS} \quad 46\]

FUNCTION  
Resets the byte size for a specific opening of a file.

CALLING SEQUENCE  
\begin{enumerate}
\item AC1: .JFN
\item AC2: Byte size, right-justified
\end{enumerate}

RETURNS  
\begin{enumerate}
\item +1: Failure, error code in AC1
\item +2: Success
\end{enumerate}
SFCDC  **J$YS 113**

**FUNCTION**
Sets the control character output control (CCOC) for the specified terminal.

**CALLING SEQUENCE**
AC1: TTY designator
AC2: CCOC word
AC3: CCOC word

**RETURNS**  +1: Always

SFMOD  **J$YS 110**

**FUNCTION**
Sets the program-related modes (in the JFN mode word) for the specified terminal.

**CALLING SEQUENCE**
AC1: TTY designator
AC2: JFN mode word

**RETURNS**  +1: Always

SFORK  **J$YS 157**

**FUNCTION**
Starts the specified process; if the process is frozen, SFORK changes the PC but does not resume the process. On extended machines, the PC section number is obtained from the process entry vector.

**RESTRICTIONS**
Requires TOPS-20 Version 5 or later for extended addressing.

**CALLING SEQUENCE**
AC1: <flags>,<process handle>
    1BO(SF%CON)  Ignore address in AC2 and start process where halted
AC2: <flags>,<process starting address> (PC of process being started)

**RETURNS**  +1: Always
SFPOS _JSYS 526

FUNCTION
Sets the position of the specified terminal's pointer.

CALLING SEQUENCE
AC1: TTY designator
AC2: <line number>,<column number>

RETURNS +1: Always

SFPTR _JSYS 27

FUNCTION
Sets the position of the specified file's pointer for subsequent I/O to the file.

CALLING SEQUENCE
AC1: JFN
AC2: Byte number to which pointer is to be set; -1 for current EOF

RETURNS +1: Failure, error code in AC1
       +2: Success

SFRKV _JSYS 201

FUNCTION
Starts the specified process using the position given in its entry vector.

CALLING SEQUENCE
AC1: Process handle
AC2: Offset in entry vector of start address to use

RETURNS +1: Always

SFTAD _JSYS 534

FUNCTION
Sets the dates and times associated with the specified file.

REstrictions
Some functions require enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Source designator
AC2: Address of argblk
AC3: Length of argblk

122
RETURNS +1: Always

ARGUMENT BLOCK

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.RSWRT</td>
<td>Internal date/time file was last written (enabled WHL/OPR for &gt;current)</td>
</tr>
<tr>
<td>1</td>
<td>.RSCRV</td>
<td>Internal date and time file was created (enabled WHL/OPR for &gt;current)</td>
</tr>
<tr>
<td>2</td>
<td>.RSREF</td>
<td>Internal date and time file was last read (enabled WHL/OPR for &gt;current)</td>
</tr>
<tr>
<td>3</td>
<td>.RSRE</td>
<td>System date and time of last write by monitor (enabled WHL/OPR)</td>
</tr>
<tr>
<td>4</td>
<td>.RSTDT</td>
<td>Tape-write date and time of archived or migrated files (enabled WHL/OPR)</td>
</tr>
<tr>
<td>5</td>
<td>.RSNET</td>
<td>On-line expiration date and time (date/time or interval)</td>
</tr>
<tr>
<td>6</td>
<td>.RSFET</td>
<td>Offline expiration date and time (date/time or interval)</td>
</tr>
</tbody>
</table>

SFUST JSYS B51

FUNCTION
Sets the name of either the author of the file or the user who last wrote to the file.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability, or caller must have write or owner access to specified file.

CALLING SEQUENCE
AC1: <function code>,..JFN
AC2: Byte pointer to ASCIZ author/user name string

RETURNS +1: Always, with updated byte pointer in AC2

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SFAUT</td>
<td>Set name of author of file</td>
</tr>
<tr>
<td>1</td>
<td>.SFLWR</td>
<td>Set name of user who last wrote file (enabled WHL/OPR)</td>
</tr>
</tbody>
</table>

SIBE JSYS 102

FUNCTION
Tests to see if the designated file input buffer is empty.

CALLING SEQUENCE
AC1: Source designator

RETURNS +1: if device is active terminal and input buffer not empty; or if device is not terminal, is open for read, and input buffer not empty
AC2: Byte count remaining in input buffer
+2: if device is non-active terminal
AC2: Error code
   if device is active terminal and input buffer
   empty; if device not terminal and not open
   for read; or if device not terminal, is open
   for read, and input buffer empty
AC2: 0

SINU JSYS S2

FUNCTION
Reads a string from the specified source.

CALLING SEQUENCE
AC1: Source designator
AC2: Byte pointer address to store string
AC3: 0 to read string that terminates with null byte
   +n to read string of n characters, or terminate on
   byte that matches contents of AC4
   -n to read string of n bytes
AC4: Byte (R-U) on which to terminate input (if +n in AC3)

RETURNS +1: Always, with
   AC1: Updated byte pointer
   AC2: Updated byte pointer
   AC3: Updated count of bytes transferred

SINR JSYS 531

FUNCTION
Reads a record from the specified device; the calling
program must specify the record size (SET TAPE RECORD-LENGTH
of .MOSRS function of MTOPR); default record size is 1000
bytes.

RESTRICTIONS
Will not read across record boundaries.

CALLING SEQUENCE
AC1: Source (device) designator
AC2: Byte pointer to address to store record
AC3: 0 to read record that terminates with null byte
   +n to read record of n characters, or terminate on
   byte that matches contents of AC4
   -n to read record of n bytes
AC4: Byte (R-U) on which to terminate input (if +n in AC3)

RETURNS +1: Always, with
   AC1: Updated byte pointer
   AC2: Updated byte pointer

124
AC3: O if specified record size = actual record size (all bytes read)
# of bytes read if specified record size > actual record size
# of bytes requested if specified record size < actual record size;
10X10 returned and unread bytes discarded

SIR  USYS 125

FUNCTION
Sets the addresses of the channel and priority level tables
for the specified process.

RESTRICTIONS
The process must run in section 0 of memory, with channel
and priority level tables in that section. (Use XSIR% to set
table addresses for multiple-section processes.)

CALLING SEQUENCE
AC1: Process handle
AC2: LEVTAB,,CHNTAB

RETURNS +1: Always

SIRCW  USYS 142

FUNCTION
Sets the mask for reserved software interrupt channels for
the specified inferior process, causing conditions occurring
on software channels that have the corresponding mask bit
set to terminate or freeze the process, rather than generate
an interrupt.

CALLING SEQUENCE
AC1: Inferior process handle
AC2: Channel mask with bits set for reserved channels
AC3: Deferred terminal interrupt word

RETURNS +1: Always

SIZEF  USYS 36

FUNCTION
Returns the length of an existing file.

CALLING SEQUENCE
AC1: JFN
RETURNS +1: Failure, error code in AC1
TOPS-20 Monitor Calls Quick Reference Guide

SIZEF

+2: Success, with
   AC2: File byte count (byte size from FDB)
   AC3: File page count

SUPRI __JSYS 245

FUNCTION
Sets the scheduler priority control word.

RESTRICTIONS
This JSYS is reserved for DIGITAL. Requires enabled WHEEL or
OPERATOR capability.

CALLING SEQUENCE
AC1: Job #
AC2: Priority word

RETURNS  +1: Always

PRIORITY WORD
Bits  Contents
B0-17  Percentage of CPU resources (1 - 99%) guaranteed to
      job; 0 for no request
B18  System job flag (JPSYS): higher priority than user
     jobs with guaranteed runtime
B24-29  Highest priority queue job may run in; 0 for no
        queue assignment request
B30-35  Lowest priority queue job may run in, specified as
        desired queue+1; 0 for no queue assignment request

SKEDX __JSYS 577

FUNCTION
Reads or modifies the monitor's scheduler data base.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Address of argblk

RETURNS  +1: Always

FUNCTION CODES
Code Symbol  Function
  1 .SKRBC Read bias control knob setting
      0.SACNT Word count including this word
      1.SAKNB Bias control knob setting (RET)
  2 .SKSBC Set bias control setting (WHL/OPR)
      0.SACNT Word count including this word
      1.SAKNB Bias control setting (1-20)
<table>
<thead>
<tr>
<th>Number</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.SKRCS</td>
<td>Read class parameters</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SACLS</td>
<td>Class of job (RET)</td>
</tr>
<tr>
<td>2</td>
<td>.SASHR</td>
<td>Share of CPU allocated to class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(RET: 0.0&lt;=n.n&lt;=1.0)</td>
</tr>
<tr>
<td>3</td>
<td>.SAUSE</td>
<td>Amount of CPU used by class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(RET: 0.0&lt;=n.n&lt;=1.0)</td>
</tr>
<tr>
<td>4</td>
<td>.SA1ML</td>
<td>1 min load avg for class (RET)</td>
</tr>
<tr>
<td>5</td>
<td>.SA5ML</td>
<td>5 min load avg for class (RET)</td>
</tr>
<tr>
<td>6</td>
<td>.SA15L</td>
<td>15 min load avg for class (RET)</td>
</tr>
<tr>
<td>4</td>
<td>.SKSCS</td>
<td>Set class parameters (WHL/DPR)</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SACLS</td>
<td>Class of job</td>
</tr>
<tr>
<td>2</td>
<td>.SASHR</td>
<td>Share of CPU allocated to class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0.0&lt;=n.n&lt;=1.0)</td>
</tr>
<tr>
<td>5</td>
<td>.SKICS</td>
<td>Start or stop class scheduler (WHL/DPR)</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SACTL</td>
<td>Control flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0(SK%SACT) Class by accounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(SK%WDF) Withhold windfall</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(SK%STP) Class scheduler off</td>
</tr>
<tr>
<td>6</td>
<td>.SKSCJ</td>
<td>Set job class (WHL/DPR for other than calling job)</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SAJOB</td>
<td>Job #; -1 for calling job</td>
</tr>
<tr>
<td>2</td>
<td>.SAJCL</td>
<td>Class of job</td>
</tr>
<tr>
<td>3</td>
<td>.SAWA</td>
<td>Windfall allocation</td>
</tr>
<tr>
<td>7</td>
<td>.SKRJP</td>
<td>Read class parameters for a job</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SAJSH</td>
<td>Job's share of CPU (RET; 0.0&lt;=n.n&lt;=1.0)</td>
</tr>
<tr>
<td>2</td>
<td>.SAJUS</td>
<td>Job's current CPU use (RET; 0.0&lt;=n.n&lt;=1.0)</td>
</tr>
<tr>
<td>10</td>
<td>.SKBCR</td>
<td>Read class setting for batch jobs</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SABCL</td>
<td>Batch class; -1 if none (RET)</td>
</tr>
<tr>
<td>11</td>
<td>.SKBCS</td>
<td>Set batch class (WHL/DPR)</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SABCL</td>
<td>Batch class; -1 for none</td>
</tr>
<tr>
<td>12</td>
<td>.SKBBG</td>
<td>Run all batch jobs in &quot;dregs&quot; queue; illegal if class scheduling in use (WHL/DPR)</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SADRS</td>
<td>0 don't run in dregs queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#0 run in dregs queue</td>
</tr>
<tr>
<td>13</td>
<td>.SKDDC</td>
<td>Reserved</td>
</tr>
<tr>
<td>14</td>
<td>.SKRCV</td>
<td>Read status</td>
</tr>
<tr>
<td>0</td>
<td>.SACNT</td>
<td>Word count including this word</td>
</tr>
<tr>
<td>1</td>
<td>.SACTL</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0(SK%SACT) Class by accounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(SK%WDF) Withhold windfall</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(SK%STP) Class scheduler off</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B3(SK%DRG) Batch jobs being run in dregs queue</td>
</tr>
</tbody>
</table>
SKPIR _JSYS 127

FUNCTION
Tests to see if the software interrupt system is enabled for the specified process, and performs a skip return if PSI enabled.

CALLING SEQUENCE
AC1: Process handle

RETURNS
+1: Software interrupt system is off
+2: Software interrupt system is on

SMAP% _JSYS 767

FUNCTION
Maps one or more contiguous sections of memory.
Maps one or more complete sections from a file to a process (Case I) or from a process to another process (Case II), creates new sections (Case III), or deletes sections from a process (Case IV).

CALLING SEQUENCE
AC1: JFN,,<file section number> (Case I)
     <fork handle>,<section number> (Case II)
     0 (Case III)
     -1 (Case IV)
AC2: <fork handle>,<process section number> (Cases I-IV)
AC3: 0,,<# (1-37) of contiguous sections to map> (Case IV)
      flags,,<# (1-37) of contiguous sections to map> (Cases I-III)
      B2(SMXRD) Allow read access (Cases I-III)
      B3(SMXWR) Allow write access (Cases I-III)
      B4(SMXEX) Allow execute access (Cases I-III)
      B6(SMXIND) Map using indirect section pointer (Case II-III)

RETURNS
+1: Always

SMON _JSYS 6

FUNCTION
Sets various flags and parameters in the monitor's data base.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability; some functions are for ARPANET systems only.

CALLING SEQUENCE
AC1: Function code
AC2: New value for function
## FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SFFAC</td>
<td>Allow FACT file entries</td>
</tr>
<tr>
<td>1</td>
<td>.SFCDE</td>
<td>CHECKD found errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'CDE) to set; 0 to clear</td>
</tr>
<tr>
<td>2</td>
<td>.SFCDR</td>
<td>CHECKD is running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'CDR) to set; 0 to clear</td>
</tr>
<tr>
<td>3</td>
<td>.SFMTS</td>
<td>Manual start in progress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'MTS) to set; 0 to clear</td>
</tr>
<tr>
<td>4</td>
<td>.SFRTS</td>
<td>Allow remote LOGINS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'RTS) to set; 0 to clear</td>
</tr>
<tr>
<td>5</td>
<td>.SFPTY</td>
<td>Allow PTY LOGINS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'PTY) to set; 0 to clear</td>
</tr>
<tr>
<td>6</td>
<td>.SFCTR</td>
<td>Allow CTY LOGINS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'CTR) to set; 0 to clear</td>
</tr>
<tr>
<td>7</td>
<td>.SFOPR</td>
<td>Operator in attendance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'OPR) to set; 0 to clear</td>
</tr>
<tr>
<td>10</td>
<td>.SFLCL</td>
<td>Allow local LOGINS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'LCL) to set; 0 to clear</td>
</tr>
<tr>
<td>11</td>
<td>.SFBE</td>
<td>Bit table errors found on startup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'BTE) to set; 0 to clear</td>
</tr>
<tr>
<td>12</td>
<td>.SFCDR</td>
<td>Users can change directory parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'CDR) to set; 0 to clear</td>
</tr>
<tr>
<td>13</td>
<td>.SFNVT</td>
<td>Allow ARPANET terminal LOGINS (ARPA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'NVT) to set; 0 to clear</td>
</tr>
<tr>
<td>21</td>
<td>.SFUSG</td>
<td>Allow USAGE file entries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'USG) to set; 0 to clear</td>
</tr>
<tr>
<td>22</td>
<td>.SFFLO</td>
<td>Set full disk latency optimization (requires KL10-E revision level 2 and R120 board MB555 revision level 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'FLO) to set; 0 to clear</td>
</tr>
<tr>
<td>23</td>
<td>.SFMTA</td>
<td>Enable MOUNT magtape allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'MTA) to set; 0 to clear</td>
</tr>
<tr>
<td>24</td>
<td>.SFMSO</td>
<td>Set system message level 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'MSO) to set; 0 to clear</td>
</tr>
<tr>
<td>25</td>
<td>.SFMS1</td>
<td>Set system message level 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1(SF'MS1) to set; 0 to clear</td>
</tr>
<tr>
<td>44</td>
<td>.SFNTN</td>
<td>Turn ARPANET on (ARPA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1 to set; 0 to clear</td>
</tr>
<tr>
<td>45</td>
<td>.SFNDC</td>
<td>Reinitialize ARPANET if down (ARPA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1 to set; 0 to clear</td>
</tr>
<tr>
<td>46</td>
<td>.SFRHI</td>
<td>Initialize ARPANET host table (ARPA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1 to set; 0 to clear</td>
</tr>
<tr>
<td>47</td>
<td>.SFTRZ</td>
<td>Set local time zone</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: time zone</td>
</tr>
<tr>
<td>50</td>
<td>.SFLHN</td>
<td>Set local ARPANET host number (ARPA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: ARPANET host number</td>
</tr>
<tr>
<td>51</td>
<td>.SFVIR</td>
<td>Enable account validation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1 to set; 0 to clear</td>
</tr>
<tr>
<td>52</td>
<td>.SFSTS</td>
<td>Enable status reporting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 1 to set; 0 to clear</td>
</tr>
<tr>
<td>53</td>
<td>.SFSDK</td>
<td>Set GETOK% defaults</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: flags,&lt;GETOK% function code&gt;</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
SMON

BO(SF%EDK) 0 to disable access checking
1 to enable access checking
B1(SF%DDK) 0 to deny access if checking disabled
1 to allow access if checking disabled

54 .SFMCY Set maximum offline expiration period
AC2: expiration period in days

55 .SFRDU Update last access read time for directories
AC2: 1 to set; 0 to clear

56 .SFACY Set maximum offline expiration period for archive files
AC2: expiration period in days

57 .SFRTW Set no-retrieval-waits flag
AC2: 1 to set; 0 to clear

60 .SFTDF Set tape mount controls
AC2: BO(MT%WUT) 1 to unload unrecognizable tapes
0 to treat unrecognizable tapes as unlabeled

61 .SFWSP Enable working set preloading
AC2: 1 to set; 0 to clear

SNDIM JSYS 750

FUNCTION
Places a message in a previously assigned ARPANET special message queue.

RESTRICTIONS
For ARPANET systems only.

CALLING SEQUENCE
AC1: B0 If on, message contains 96-bit leader; if off, message contains 32-bit leader
B1 If on, data in high-order 32 bits of each word of message; if off, data in all 36 bits of each word of message
B18-35 Special queue header
AC2: Address of extended message

RETURNS +1: Failure, error code in AC1
+2: Success, message queued

See BBN Report #1822 for the format of the extended message.
FUNCTION
Performs system performance analysis.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Function-specific argument
AC3: Function-specific argument
AC4: Function-specific argument

RETURNS
+1: Failure, error code in AC1
+2: Success

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SNPLC</td>
<td>Declare and lock code into monitor's address space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: number of pages desired</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: user page number of start of breakpoint routines to be locked</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: monitor page # corresponding to user page #</td>
</tr>
<tr>
<td>1</td>
<td>.SNPLS</td>
<td>Lock swappable monitor</td>
</tr>
<tr>
<td>2</td>
<td>.SNPDB</td>
<td>Define a breakpoint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Number of breakpoint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address in monitor space to be patched</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC4: Instruction to be executed before patched instruction</td>
</tr>
<tr>
<td>3</td>
<td>.SNPBB</td>
<td>Insert all breakpoints and start analyzing</td>
</tr>
<tr>
<td>4</td>
<td>.SNPRB</td>
<td>Remove all breakpoints and stop analyzing</td>
</tr>
<tr>
<td>5</td>
<td>.SNPPU</td>
<td>Unlock and release all storage; remove all breakpoints</td>
</tr>
<tr>
<td>6</td>
<td>.SNPSY</td>
<td>Obtain address of monitor symbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Radix-50 symbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Radix-50 program name if local address desired; 0 to search entire symbol table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Monitor address or value of symbol</td>
</tr>
<tr>
<td>7</td>
<td>.SNPAD</td>
<td>Obtain monitor symbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: 36-bit value of symbol to be looked up in monitor's symbol table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Radix-50 program name if local value desired; 0 to search entire symbol table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Radix-50 symbol closest to and less than given value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Difference between returned value and given value</td>
</tr>
</tbody>
</table>
FUNCTION
Tests to see if the designated file output buffer is empty.

CALLING SEQUENCE
AC1: Destination designator

RETURNS  +1: Output buffer is not empty
          AC2: number of bytes remaining in output buffer
       +2: Output buffer is empty
          AC2: 0
          Error return
          AC2: Error code

FUNCTION
Tests to see if the designated file's output buffer is full.

CALLING SEQUENCE
AC1: File designator

RETURNS  +1: Output buffer is not full
          AC2: Count of bytes in buffer
          Error return
          AC2: 0
       +2: Output buffer is full or error
          AC2: Count of bytes in buffer if no error

FUNCTION
Writes a string to the specified destination.

CALLING SEQUENCE
AC1: Destination designator
AC2: Byte pointer to string to be written
AC3: 0 to write string that terminates with null byte
     +n to write string of n characters, or terminate on byte that matches contents of AC4
     -n to write string of n bytes
AC4: Byte (R-U) on which to terminate output (if +n in AC3)

RETURNS  +1: Always, with
          AC1: Updated byte pointer
          AC2: Updated byte pointer
          AC3: Updated count of bytes transferred
TOPS-20 Monitor Calls Quick Reference Guide
SOUTR

SOUTR  JSYS 532

FUNCTION
Writes a variable-length record to the specified device; the
calling program must specify the record size (SET TAPE
RECORD-LENGTH of .MOSR5 function of WTOPR); default record
size is 1000 bytes.

CALLING SEQUENCE
AC1: Destination designator
AC2: Byte pointer to string to be written
AC3: 0 to write record that terminates with null byte
+\( \| \) to write record of \( n \) characters, or terminate on
byte that matches contents of AC4
-\( \| \) to write record of \( n \) bytes
AC4: Byte (R-U) on which to terminate input (if +\( \| \) in AC3)

RETURNS +1: Always, with
AC2: Last non-zero byte written
AC3: (# bytes written) - (# bytes requested)

SPACS  JSYS 60

FUNCTION
Sets the accessibility of a page.

CALLING SEQUENCE
AC1: <process/file designator>,<process/file page number>
AC2: Access flags
B2(PA%RD)  Permit read access
B3(PA%WT)  Permit write access
B4(PA%EX)  Permit execute access
B9(PA%CPY)  Permit copy-on-write

SPJFN  JSYS 207

FUNCTION
Sets the primary JFNs (.PRIIN and .PRIOU) for the specified
process.

CALLING SEQUENCE
AC1: Process handle
AC2: <primary input JFN>,<primary output JFN>; or -1 in
appropriate half to set to process's controlling
terminal

RETURNS +1: Always

133
SPLFK

FUNCTION
Splices a process structure.

RESTRICTIONS
The new superior must be either the calling process or an inferior of it; the new inferior process must be an inferior of the calling process. The new superior and new inferior processes must not be the same process.

CALLING SEQUENCE
AC1: Process handle of new superior process
AC2: Process handle of new inferior process

RETURNS
+1: Failure, error code in AC1
+2: Success, with
   AC1: process handle of new superior
   AC2: process handle of new inferior

SPOOL  JSYS 517

FUNCTION
Defines and initializes a device to be used for input spooling, or sets and reads the directory for a spooled device.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: <length of argblk>,<function code>
AC2: Address of argblk

RETURNS
+1: Failure, error code in AC1
+2: Success

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SPLDI</td>
<td>Define an input spooling device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SPLDV Device designator of input device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SPLNA Pointer to input file string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 .SPLGN Generation number of first file</td>
</tr>
<tr>
<td>1</td>
<td>.SPLSD</td>
<td>Set directory of spooled device (enabled WHL/DPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SPLDV Device designator of spooled device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SPLDR Directory number of user who opened spooled device</td>
</tr>
<tr>
<td>2</td>
<td>.SPLRD</td>
<td>Read directory of spooled device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SPLDV Designator of spooled device</td>
</tr>
</tbody>
</table>
SPRIW  JSYS  243

FUNCTION
Sets the priority word for the specified process.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Process handle
AC2: Priority word

RETURNS  +1: Always

PRIORITY WORD
Bits  Contents
B0-17 Percentage of CPU resources (1 - 99%) guaranteed to job; 0 for no request
B18 System job flag (JPSYS); higher priority than user jobs with guaranteed runtime
B24-29 Highest priority queue job may run in; 0 for no queue assignment request
B30-35 Lowest priority queue job may run in, specified as desired queue+1; 0 for no queue assignment request

SSAVE  JSYS  203

FUNCTION
Creates a sharable, save-format file for the given JFN by copying (not sharing) pages from the given process.

CALLING SEQUENCE
AC1: <process handle>,JFN
AC2: One table entry; or 0.,<table address>
AC3: 2nd word of 2-word table entry (if SS%EPN set); or 0

RETURNS  +1: Always

TABLE ENTRY
Word  Contents
0  Flags:
   B0-17(SS%NNP)  # of pages in each group
   B18(SS%CPY)  Allow copy-on-write access
   B19(SS%CA)  Limit access according to user's current page access ANDed with table word access
   B20(SS%RD)  Allow read access
   B21(SS%WR)  Allow write access
   B22(SS%EXE)  Allow execute access
   B23(SS%EPN)  Table entry is 2 words long; 2nd word contains page # of 1st page of group
   B27-35(SS%FPN)  If OB23(SS%EPN), page # of 1st page in group; if 1B23(SS%EPN), 0
1  Page number of 1st page in group (for pages in
TOPS-20 Monitor Calls Quick Reference Guide

**SSAVE**

non-zero section)

2 0

**STAD JSYS 226**

**FUNCTION**
Sets the system's date.

**RESTRICTIONS**
Requires enabled WHEEL or OPERATOR capability if the system's date is already set.

**CALLING SEQUENCE**
AC1: day,,<fraction of day>

**RETURNS**
+1: Failure, error code in AC1
+2: success

**STCMP JSYS 540**

**FUNCTION**
Compares two ASCIZ strings.

**RESTRICTIONS**
Alphabets are compared in upper case, regardless of case in string.

**CALLING SEQUENCE**
AC1: Byte pointer to test string
AC2: Byte pointer to base string

**RETURNS**
+1: always, with

AC1:
0 if strings are equal; or flags
BO(SC%LSS) Test string is less than base string
B1(SC%SUB) Test string is subset of base string
B2(SC%GTR) Test string is greater than base string
AC2: Base byte pointer, pointing before 1st non-matching byte

**STDEV JSYS 120**

**FUNCTION**
Translates the given device name string to its corresponding device designator.
CALLING SEQUENCE
AC1: Byte pointer to device name string

RETURNS  +1: Failure, error code in AC2
          +2: Success, device designator in AC2

STI  J SYS 114

FUNCTION
Simulates terminal input.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability if specified
terminal not assigned or opened by calling process, or is
not accepting advice.

CALLING SEQUENCE
AC1: TTY designator
AC2: Character to be input, right-justified

RETURNS  +1: Always

STIW  J SYS 174

FUNCTION
Sets the terminal interrupt word for the entire job or a
specific process.

RESTRICTIONS
Requires enabled SC%CTY capability in process capability
word.

CALLING SEQUENCE
AC1: BO(STXDIM) Set deferred terminal interrupt mask
given in AC3
       B18-35(STXPRH) process handle
AC2: Terminal interrupt word mask (1Bn enables terminal
code n)
AC3: Deferred terminal interrupt word mask (1Bn defers
terminal code n)

RETURNS  +1: Always
TOPS-20 Monitor Calls Quick Reference Guide

ST0  JSYS 246

FUNCTION
Simulates terminal output.

CALLING SEQUENCE
AC1: TTY designator

RETURNS  +1: Always, with character right-justified in AC2

STPAR  JSYS 217

FUNCTION
Sets the device-related modes for the specified terminal.

CALLING SEQUENCE
AC1: TTY designator
AC2: JFN mode word

RETURNS  +1: Always

STPPN  JSYS 556

FUNCTION
Translates the given directory name string to its corresponding project-programmer number (a TOPS-10 36-bit directory designator).

RESTRICTIONS
In non-zero sections, DWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: JFN; 36-bit directory number; or byte pointer to ASCIZ directory name string

RETURNS  +1: Always, with project-programmer number in AC2

STSTS  JSYS 25

FUNCTION
Clears the status of a file.

CALLING SEQUENCE
AC1: O,JFN
AC2: Flags
B9(GS%ERR)  File may be in error
B13(GS%HLT)  I/O errors are terminating conditions
B17(GS%FRK)  JFN is restricted

RETURNS  +1: Failure, error code in AC1
+2: success

**STYP JSYS 302**

FUNCTION
Sets the terminal type number for the specified terminal line.

CALLING SEQUENCE
AC1: TTY designator
AC2: TTY type

RETURNS +1: Always

**SWJFN JSYS 47**

FUNCTION
Swaps the association of two JFNs by exchanging all information cells of each JFN.

CALLING SEQUENCE
AC1: JFN
AC2: Another JFN

RETURNS +1: Always

**SWTRP% JSYS 573**

FUNCTION
Provides a process with the ability to intercept arithmetic overflow or underflow conditions.

CALLING SEQUENCE
AC1: Process handle
AC2: Function code
AC3: Function-dependent argument

RETURNS +1: Always

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SWRT</td>
<td>Set arithmetic trap location</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of arithmetic trap block; 0 to clear</td>
</tr>
<tr>
<td>1</td>
<td>.SWRT</td>
<td>Read arithmetic trap location</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Trap block address; 0 if none set (RET)</td>
</tr>
<tr>
<td>2</td>
<td>.SWLUT</td>
<td>Set LUUD block address for non-zero sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: LUUD address; 0 to clear</td>
</tr>
<tr>
<td>3</td>
<td>.SWLUT</td>
<td>Read LUUD block address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: LUUD address; 0 if none set (RET)</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
SWTRP%

LUU0 BLOCK FORMAT
Offset 0
12 13 17 18  26 27 30 31 35

.ARPFL(0) | PC flags | 0 | opcode | AC | 0 |

.ARDPC(1) | 0 | Location of LUU0 +1 |

.AREFA(2) | 0 | E of the LUU0 |

.ARNPC(3) | 0 | New PC |
0 5 6 35

SYERR JSYS 527

FUNCTION
Places information in the system error file.

RESTRICTIONS
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability.

CALLING SEQUENCE
AC1: Address of argblk
AC2: Length of argblk

RETURNS +1: Always

SYSGT JSYS 16

FUNCTION
Returns the table number, table length, and word 0 of the specified system table.

CALLING SEQUENCE
AC1: SIXBIT table name

RETURNS +1: Always, with
AC1: Word 0 of table
AC2: - # of words in table, table #; 0 if table not found

TBADD JSYS 536

FUNCTION
Adds an entry to a standard-formatted command table used for user program command recognition.

CALLING SEQUENCE
AC1: Address of table
AC2: Entry to be added to table

140
RETURNS +1: Always, with address of new entry in AC1

**TBDEL JSYS 535**

FUNCTION
Deletes an entry from a standard-formatted command table used for user program command recognition.

CALLING SEQUENCE
AC1: Address of table
AC2: Address of entry to be deleted

RETURNS +1: Always

**TBLUK JSYS 537**

FUNCTION
Compares the specified string with strings indicated by a command table.

CALLING SEQUENCE
AC1: Address of table
AC2: Byte pointer to string to be compared with string in table

RETURNS +1: Always, with
AC1: Address of entry that matches input string, or address where entry would be if in table
AC2: Recognition flags
   BO(TL%NOM) Input string has no match in table
   B1(TL%AM8) Input string has more than one match in table
   B2(TL%ABR) Input string is valid abbreviation
   B3(TL%EXM) Input string has exact match in table
AC3: Pointer to remainder of string in table if B2

COMMAND TABLE FORMAT
Word Contents
O <=# of remaining words>,<# of remaining words>
1 - n <address of argbik>,<available to user>

ARGUMENT BLOCK
Word Contents
O If O(BO-6) and 1B7(CM%FW), <flags>B18-35 and string begins in next word; if ~O(BO-6) or 0B7,string starts in this word
B34(CM%NDR) Do not recognize this string
Start of string if O(BO-6) and 1B7(CM%FW) in word 0

FUNCTION
Reads input from a terminal or a file.

CALLING SEQUENCE
AC1: Address of argb1k

RETURNS
+1: Failure, error code in AC1
+2: Success, updated pointer in word .RDDBP, appropriate bits set in word .RDFLG, and updated count in word .RDDBc of argb1k

ARGUMENT BLOCK
Word  Symbol   Contents

0  .RDCWB  Word count not including this word
1  .RDFLG  Flags
   B0(RD%BRK)  Break on CTRL/Z or ESC
   B1(RD%TOP)  Break on CTRL/G, CTRL/K, CTRL/L, CTRL/Z, ESC, CR, LF
   B2(RD%PUN)  Break on punctuation
   B3(RD%BEL)  Break on EDL (CRLF or LF only)
   B4(RD%CRF)  Suppress CR and return LF only
   B5(RD%RND)  Return if user tries to delete past beginning of buffer
   B6(RD%JFN)  JFNs in word .RDIDU
   B7(RD%RIE)  Return if input buffer empty
   B8(RD%BBG)  Not used
   B9(RD%BEG)  Return when .RDBKL pointer is reached
   B10(RD%RAI) Convert lowercase input to UPPERCASE
   B11(RD%SUI) Suppress CTRL/U indication
   B12(RD%BTM) Break character terminated input (RET)
   B13(RD%BFE) Returned because user tried to delete past beginning of buffer (RET)
   B14(RD%BLR) Backup limit for editing reached (RET)

2  .RDIDU  Byte pointer to string; or <input JFN>,<output JFN>
3  .RDDBP  Byte pointer to destination string buffer
4  .RDDBC  Number of bytes in destination string
5  .RDBFP  Byte pointer to beginning of destination buffer
6  .RDRTY  Byte pointer to beginning of CTRL/R buffer
7  .RDBRK  Address of 4-word break character mask block
10  .RDBKL  Byte pointer to backup limit in destination buffer

142
TFORK  JSYS 321

FUNCTION
Sets and removes monitor call intercepts (JSYS traps) for the given inferior processes.

REstrictions
Requires enabled WHEEL, OPERATOR, or MAINTENANCE capability for use on execute-only processes.

Calling Sequence
AC1: <function code>,<process handle>
AC2: <interrupt channel>,<size of monitor call bit table> (in bits)
AC3: Address of monitor call bit table

returns +1: Always

Function Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.TFSET</td>
<td>Set JSYS traps for given process (illegal for execute-only processes)</td>
</tr>
<tr>
<td>1</td>
<td>.TFRAL</td>
<td>Remove all JSYS traps for given process (illegal for execute-only processes)</td>
</tr>
<tr>
<td>2</td>
<td>.TFRTG</td>
<td>Remove JSYS traps indicated in monitor call bit table for given process (illegal for execute-only processes)</td>
</tr>
<tr>
<td>3</td>
<td>.TFSPS</td>
<td>Set interrupts on given software channel</td>
</tr>
<tr>
<td>4</td>
<td>.TFRPS</td>
<td>Return interrupt channel in left half of AC2</td>
</tr>
<tr>
<td>5</td>
<td>.TFTST</td>
<td>Test if caller is to be intercepted when it attempts to execute monitor calls; On return AC2: -1 intercept; 0 no intercept</td>
</tr>
<tr>
<td>6</td>
<td>.TFRES</td>
<td>Remove intercepts for all inferiors and clear assigned software channels</td>
</tr>
<tr>
<td>7</td>
<td>.TFUOO</td>
<td>Set JSYS traps for TOPS-10 UUDs for given process (illegal for execute-only processes)</td>
</tr>
<tr>
<td>10</td>
<td>.TFSUU</td>
<td>Set JSYS traps for both TOPS-10 UUDs indicated in monitor call bit table (illegal for execute-only processes)</td>
</tr>
<tr>
<td>11</td>
<td>.TFRUU</td>
<td>Remove JSYS traps for TOPS-10 UUDs</td>
</tr>
</tbody>
</table>

THIBR  JSYS 770

FUNCTION
Blocks the current process for the specified elapsed time or until awakened by a TWAKE monitor call.

REstrictions
This call is temporary and may not be defined in future releases.
CALLING SEQUENCE
AC1: 0.,<maximum number of seconds to block>

RETURNS +1: Never
+2: Always

TIME JSYS 14

FUNCTION
Returns the amount of time since the system was last restarted.

RETURNS +1: Always, with
AC1: Time in milliseconds, right-justified
AC2: 1000 (divisor for conversion to seconds)

TIMER JSYS 522

FUNCTION
Controls the amount of time either a process within a job or the entire job can run.

CALLING SEQUENCE
AC1: <process handle>,<function code>
AC2: Time at which to generate interrupt
AC3: Software channel number on which to generate interrupt

RETURNS +1: Failure, error code in AC1
+2: Success

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.TIMRT</td>
<td>Set total runtime of entire job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Total runtime in mss</td>
</tr>
<tr>
<td>1</td>
<td>.TIMEL</td>
<td>Set elapsed time for process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Elapsed time in mss before interrupt</td>
</tr>
<tr>
<td>2</td>
<td>.TIMDT</td>
<td>Set exact time to generate interrupt for process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Time of interrupt in internal format</td>
</tr>
<tr>
<td>3</td>
<td>.TIMDD</td>
<td>Remove any pending interrupts at given time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Time of interrupt in internal format</td>
</tr>
<tr>
<td>4</td>
<td>.TIMBF</td>
<td>Remove any pending interrupts before given time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Time of interrupt in internal format</td>
</tr>
<tr>
<td>5</td>
<td>.TIMAL</td>
<td>Remove all pending requests for given process</td>
</tr>
</tbody>
</table>
FUNCTION
Controls terminal linking.

RESTRICTIONS
Some functions require enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: B0(TL%CRD)
     Clear link from remote to object
designator
B1(TL%COR)
     Clear link from object to remote
designator
B2(TL%EDR)
     Establish link from object to remote
designator
B3(TL%EDR)
     Establish link from remote to object
designator
B4(TL%SAB)
     Examine B5(TL%ABS) to determine setting
     of object designator's accept link bit
B5(TL%ABS)
     Set object designator's accept link bit
B6(TL%STA)
     Examine B7(TL%AAD) to determine setting
     of object designator's accept advice
     bit
B7(TL%AAD)
     Set object designator's accept advice
     bit
B18-35(TL%OBu)
Object designator

AC2: 0,,<remote designator>

RETURNS
+1: Failure, error code in AC1
    +2: Success

FUNCTION
Returns various flags and parameters in the monitor's data
base.

CALLING SEQUENCE
AC1: Function code
AC2: Function-specific arguments

RETURNS
+1: Always, with
    AC2: Value of function
    normally, 1 if set; 0 if clear

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SFFAC</td>
<td>FACT file entries are allowed</td>
</tr>
<tr>
<td>1</td>
<td>.SFCDE</td>
<td>CHECKD found errors</td>
</tr>
<tr>
<td>2</td>
<td>.SFCDR</td>
<td>CHECKD is running</td>
</tr>
<tr>
<td>3</td>
<td>.SFMST</td>
<td>Manual start is in progress</td>
</tr>
<tr>
<td>4</td>
<td>.SFRMT</td>
<td>Remote LOGINS are allowed</td>
</tr>
<tr>
<td>5</td>
<td>.SFPTY</td>
<td>PTY LOGINS are allowed</td>
</tr>
<tr>
<td>6</td>
<td>.SFCTY</td>
<td>CTY LOGINS are allowed</td>
</tr>
</tbody>
</table>

145
7 .SFOPR Operator is in attendance
10 .SFCL Local LOGINS are allowed
11 .SFATE Bit table errors found on startup
12 .SFCRD Users can change nonprivileged directory
   parameters
13 .SFVNT ARPANET terminal LOGINS are allowed
21 .SFUSG USAGE file entries are allowed
22 .SFLO Disk latency optimization using RH20 backup
   register is enabled
23 .SFMTA MOUNT transient tape allocation is enabled
24 .SFMS0 System message level 0 is enabled
25 .SFMS1 System message level 1 is enabled
44 .SFNTN ARPANET is on
45 .SFNDU ARPANET will be reinitialized if it is down
46 .SFNHI ARPANET host table will be initialized
47 .SFMTZ Local time zone
50 .SFLOF ARPANET local host number
51 .SFATR Account validation is running
52 .SFSTS Status reporting is enabled
53 .SFSDK GETOK% defaults
AC2: flags, <GETOK% function code>
   0 Access checking is disabled
   1 Access checking is enabled
   0 Access is denied if checking disabled
   1 Access is allowed if checking disabled
54 .SFMCY Maximum offline expiration period in days for
   ordinary files
55 .SFDOU Update last access read time for directories
56 .SFDACY Maximum offline expiration period in days for
   archive files
57 .SFRTW File-retrieval requests should fail
60 .SFOTT Tape mount controls
   0 Treat unrecognizable tapes
   1 Unload unrecognizable tapes
   as unlabeled
61 .SFWSP Enable working set preloading

TTMSG USYS 775

FUNCTION
Sends a message to a specified terminal or to all terminals.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability to send to all
-terminals. Messages sent by privileged users may have a
-maximum of 581 characters; messages sent by non-privileged
-users may have a maximum of 526 characters. This call is
temporary and may not be defined in future releases.
CALLING SEQUENCE
AC1: 400000 + TTY number; or -1 for all terminals
AC2: Byte pointer to message string

RETURNS +1: Always

TWAKE JSYS 771

FUNCTION
Wakes the specified job that is blocked because of the execution of a THIBR call.

RESTRICTIONS
This call is temporary and may not be defined in future releases.

CALLING SEQUENCE
AC1: 0,,<number of job to be awakened>

RETURNS +1: Failure, error code in AC1
+2: Success

UFPGS JSYS 525

FUNCTION
Updates pages of the specified file.

CALLING SEQUENCE
AC1: JFN,,<file page # of 1st page to be updated>
AC2: Flags,,<# of sequential pages to update>
BO(UF%NOW) Perform UFPGS without blocking

RETURNS +1: Failure, error code in AC1
+2: Success

USAGE JSYS 564

FUNCTION
Controls accounting on the system by writing entries into the system’s data file.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: Function code
AC2: Function argument; or address of record descriptor block

RETURNS +1: Always
FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.USENT</td>
<td>Write entry into system’s data file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Address of record descriptor block</td>
</tr>
<tr>
<td>1</td>
<td>.USCLS</td>
<td>Close system’s data file</td>
</tr>
<tr>
<td>2</td>
<td>.USCKP</td>
<td>Perform checkpoint of all jobs</td>
</tr>
<tr>
<td>3</td>
<td>.USLGI</td>
<td>Initialize checkpoint entry for job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Address of record descriptor block</td>
</tr>
<tr>
<td>4</td>
<td>.USLGD</td>
<td>Terminate checkpoint entry for job and write entry to system’s data file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Address of record descriptor block</td>
</tr>
<tr>
<td>5</td>
<td>.USEEN</td>
<td>Terminate current session, write entry to system’s data file, and initialize new</td>
</tr>
<tr>
<td></td>
<td></td>
<td>checkpoint entry for job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Address of record descriptor block</td>
</tr>
<tr>
<td>6</td>
<td>.USCKI</td>
<td>Set checkpoint time interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Time interval in minutes</td>
</tr>
<tr>
<td>7</td>
<td>.USENA</td>
<td>Install accounting data base into running monitor from PS:&lt;SYSTEM&gt;ACCOUNTS-TABLE.BIN</td>
</tr>
<tr>
<td>10</td>
<td>.USCAS</td>
<td>Change accounting shift</td>
</tr>
<tr>
<td>11</td>
<td>.USSAS</td>
<td>Set accounting shifts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Pointer to argblk of format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 &lt;# table entries&gt;,&lt;max # entries&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1-n B0-6(USSDOW) Days-of-week entry in effect (0=Monday)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B7-17 Not used, must be zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Time in seconds since (USSSM) midnight for accounting shift change</td>
</tr>
<tr>
<td>12</td>
<td>.USRAS</td>
<td>Read accounting shifts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: Pointer to argblk (see .USSAS for format)</td>
</tr>
</tbody>
</table>

**USRIO** JSYS 310

FUNCTION

Places the user program into user I/O mode for executing various hardware I/O instructions.

RESTRICTIONS

Requires enabled WHEEL or OPERATOR capability.

RETURNS

+1: Failure, error code in AC1
+2: Success, user I/O flag set
UTEST  JSYS 563

FUNCTION
Provides a method for determining if every instruction in a section of monitor code actually gets executed.

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.

CALLING SEQUENCE
AC1: <function code>,<length of argblk>
AC2: Address of argblk

RETURNS +1: Always

FUNCTION CODES
Code  Symbol  Function
0     .UTSET  Start testing code
1     .UTCLR  Stop testing code and update bit map in argblk

ARGUMENT BLOCK
Word  Symbol  Contents
0     .UTADR  Address of beginning of code section to be tested
1     .UTLEN  Length of code section to be tested
2     .UTMAP  Start of bit map representing instructions to be tested in code section

UTFRK  JSYS 323

FUNCTION
Resumes the execution of a process that was suspended because of a monitor call intercept.

CALLING SEQUENCE
AC1: Flags,<process handle>
      B0(UT%TRP)  Cause failure return for suspended process

RETURNS +1: Always

VACCT  JSYS 566

FUNCTION
Verifies accounts by validating the supplied account for the given user.

CALLING SEQUENCE
AC1: user number; directory number; or -1 for current user
AC2: Byte pointer to account string
RETURNS +1: Always, with updated pointer in AC2

WAIT USYS 306

FUNCTION
Dismisses the current process indefinitely and does not return.

WFDRK USYS 163

FUNCTION
Causes the current process to wait for an inferior process to terminate.

CALLING SEQUENCE
AC1: Inferior process handle

RETURNS +1: Always, when specified processes terminates

WILD% USYS 565

FUNCTION
Compares a possibly wild string against a non-wild string to see if the latter matches the wild string.

CALLING SEQUENCE
AC1: Flags,.<function code>
AC2: Wild argument: JFN or byte pointer to string
AC3: Non-wild argument: JFN or byte pointer to string

RETURNS +1: Always

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.WLSTR</td>
<td>Compare non-wild string against wild string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC1: B0(WL%LCD) Lowercase characters are distinct from uppercase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC1: 0 Strings matched</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0(WL%NOM) If on, non-wild string did not match wild string</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(WL%ABR) If on, non-wild string is abbreviation of wild string</td>
</tr>
<tr>
<td>1</td>
<td>.WLJFN</td>
<td>Compare non-wild filespec against wild filespec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On return</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC1: 0 Filespecs matched</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(WL%DEV) Device field does not match</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(WL%DIR) Directory field does not match</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

B3(WL%NAM) Name field does not match
B4(WL%EXT) File type does not match
B5(WL%GEN) Generation number does not match

XGSEV% USYS 614

FUNCTION
Gets an extended special entry vector that has been set to allow use of TOPS-10 Compatibility and RMS entry vectors in non-zero sections.

CALLING SEQUENCE
AC1: <vector type code>,<fork handle>

RETURNS +1: Always, with
AC2: Length of entry vector
AC3: B0-5 Flags
B6-35 Address of entry vector

XGTPW% USYS 612

FUNCTION
Returns the page-fail words of a process that runs in more than one section of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK

Word  Contents
0    Length of block, including this word
1    0
2    0
3    0
4    0

On return
Flags
B0(PF%USR) Page failure on user-mode reference
B1(PF%WTF) Page failure on write reference

On return
Address that referenced page

On return
MUUD opcode and AC

On return
30-bit effective address of MUUD
FUNCTION
Returns the entry vector of a specified process which runs in more than one section of memory.

CALLING SEQUENCE
AC1: Process handle

RETURNS +1: Always, with
AC2: Length of entry vector
AC3: address of entry vector

XRIR% JSYS 601

FUNCTION
Reads the addresses of the channel and priority level tables for a process running in more than one section of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK
Word Contents
0 Length of argblk, including this word
1 Address of interrupt level table
2 Address of channel table

XRMAP% JSYS 611

FUNCTION
Acquires a handle on a page in an extended process to determine the access allowed for that page.

CALLING SEQUENCE
AC1: <process handle>, 0
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK
Word Contents
0 Length of argblk, including this word
1 Number of pages in this group on which to return data
   On return <process/file designator>, <page number> (page handle)
2 Number of first page in this group
   On return Access flags; or -1 if page non-existant
3 Address of block for returned data
n Number of pages in this group on which to return data
n+1 Number of first page in this group
n+2 Address of block for returned data

ACCESS FLAGS
Bit  Symbol  Meaning
B2  RM%RD  Read access allowed
B3  RM%WR  Write access allowed
B4  RM%EX  Execute access allowed
B5  RM%PEX Page exists
B9  RM%CPY Copy-on-write access allowed

XSFRK% JSYS 605

FUNCTION
Starts the specified process in a non-zero section of memory.

CALLING SEQUENCE
AC1: Flags,.<process handle>
    BO(SF%CDN)  Continue process that has halted
AC2: <PC flags>,.0
AC3: Address to set PC to (ignored if SC%CDN on)

RETURNS +1: Always

XSIR% JSYS 602

FUNCTION
Sets the addresses of the channel and priority level tables for a process running in one or more sections of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: Address of argblk

RETURNS +1: Always

ARGUMENT BLOCK
Word Contents
0  Length of argblk, including this word
1  Address of interrupt level table
2  Address of channel table

153
XSSEV% JSYS 613

FUNCTION
Allows setting of extended special entry vector for use with TOPS-10 Compatibility Package and RMS entry vectors in non-zero sections.

CALLING SEQUENCE
AC1: <vector type code>..<<fork handle>
    0 .XSEVC TOPS-10 Compatibility
    1 .XSEVD RMS
AC2: Length of entry vector
AC3: B1(XS%EEV) Extended entry vector; if on, entry vector points to 2-word extended PC and extended format UUD word
    B6-35 Address of entry vector

RETURNS +1: Always

XSVEC% JSYS 607

FUNCTION
Sets or clears the entry vector of a process that runs in one or more sections of memory.

CALLING SEQUENCE
AC1: Process handle
AC2: Length of entry vector; or 0 to clear
AC3: Address of entry vector

RETURNS +1: Always
<table>
<thead>
<tr>
<th>ASCII Code</th>
<th>Wake-up Code</th>
<th>CCDC Code</th>
<th>Character or Control Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>C</td>
<td>1B1</td>
<td>CTRL/\ null,break</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>1B3</td>
<td>CTRL/A</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>1B5</td>
<td>CTRL/B</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>1B7</td>
<td>CTRL/C</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>1B9</td>
<td>CTRL/D</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>1B11</td>
<td>CTRL/E</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>1B13</td>
<td>CTRL/F</td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>1B15</td>
<td>CTRL/G bell</td>
</tr>
<tr>
<td>10</td>
<td>F</td>
<td>1B17</td>
<td>CTRL/H backspace</td>
</tr>
<tr>
<td>11</td>
<td>P</td>
<td>1B19</td>
<td>CTRL/I horizontal tab</td>
</tr>
<tr>
<td>12</td>
<td>F</td>
<td>1B21</td>
<td>CTRL/J line feed</td>
</tr>
<tr>
<td>13</td>
<td>C</td>
<td>1B23</td>
<td>CTRL/K vertical tab</td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>1B25</td>
<td>CTRL/L form feed</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>1B27</td>
<td>CTRL/M carriage return</td>
</tr>
<tr>
<td>16</td>
<td>C</td>
<td>1B29</td>
<td>CTRL/N</td>
</tr>
<tr>
<td>17</td>
<td>C</td>
<td>1B31</td>
<td>CTRL/O</td>
</tr>
<tr>
<td>20</td>
<td>C</td>
<td>1B33</td>
<td>CTRL/P</td>
</tr>
<tr>
<td>21</td>
<td>C</td>
<td>1B35</td>
<td>CTRL/Q</td>
</tr>
<tr>
<td>22</td>
<td>C</td>
<td>2B1</td>
<td>CTRL/R</td>
</tr>
<tr>
<td>23</td>
<td>C</td>
<td>2B3</td>
<td>CTRL/S</td>
</tr>
<tr>
<td>24</td>
<td>C</td>
<td>2B5</td>
<td>CTRL/T</td>
</tr>
<tr>
<td>25</td>
<td>C</td>
<td>2B7</td>
<td>CTRL/U</td>
</tr>
<tr>
<td>26</td>
<td>C</td>
<td>2B9</td>
<td>CTRL/V</td>
</tr>
<tr>
<td>27</td>
<td>C</td>
<td>2B11</td>
<td>CTRL/W</td>
</tr>
<tr>
<td>30</td>
<td>C</td>
<td>2B13</td>
<td>CTRL/X</td>
</tr>
<tr>
<td>31</td>
<td>C</td>
<td>2B15</td>
<td>CTRL/Y</td>
</tr>
<tr>
<td>32</td>
<td>C</td>
<td>2B17</td>
<td>CTRL/Z</td>
</tr>
<tr>
<td>33</td>
<td>all</td>
<td>2B19</td>
<td>ESCAPE (altmode)</td>
</tr>
<tr>
<td>34</td>
<td>C</td>
<td>2B21</td>
<td>CTRL/backslash</td>
</tr>
<tr>
<td>35</td>
<td>C</td>
<td>2B23</td>
<td>CTRL/right square bracket</td>
</tr>
<tr>
<td>36</td>
<td>C</td>
<td>2B25</td>
<td>CTRL/uparrow</td>
</tr>
<tr>
<td>37</td>
<td>F</td>
<td>2B27</td>
<td>CTRL/backarrow</td>
</tr>
<tr>
<td>40</td>
<td>P</td>
<td></td>
<td>SPACE</td>
</tr>
<tr>
<td>41</td>
<td>P</td>
<td></td>
<td>!</td>
</tr>
<tr>
<td>42</td>
<td>P</td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>43</td>
<td>P</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>44</td>
<td>P</td>
<td></td>
<td>%</td>
</tr>
<tr>
<td>45</td>
<td>P</td>
<td></td>
<td>&amp;</td>
</tr>
<tr>
<td>46</td>
<td>P</td>
<td></td>
<td>(</td>
</tr>
<tr>
<td>47</td>
<td>P</td>
<td></td>
<td>)</td>
</tr>
<tr>
<td>48</td>
<td>P</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>49</td>
<td>P</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>50</td>
<td>P</td>
<td></td>
<td>:</td>
</tr>
<tr>
<td>51</td>
<td>P</td>
<td></td>
<td>&lt;</td>
</tr>
<tr>
<td>52</td>
<td>P</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>53</td>
<td>P</td>
<td></td>
<td>;</td>
</tr>
<tr>
<td>54</td>
<td>P</td>
<td></td>
<td>/</td>
</tr>
<tr>
<td>55</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60-71</td>
<td>A</td>
<td>0-9</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>P</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TOP5-20 Monitor Calls Quick Reference Guide
CCDC Word

75 p =
76 p >
77 p ?
100 p @
101-132 a UPPERCASE LETTERS A-Z
133 p [
134 p \,
135 p ]
136 p ^
137 p _
140 p accent grave
141-172 a lowercase letters a-z
173(1) p {
174(1) p |
175(1) p }
176(1) p ~
177 a11 DELETE (RUBOUT)

+-----------------------------------------+-------------------+
| A Alphanumeric character               | O(00) Ignore (send nothing) |
| C Non-formatting CTRL/char             | 1(01) Indicate by ^X |
| F Formatting CTRL/char                 | 2(10) Send character code |
| P Punctuation character                | 3(11) Simulate format action |
+-----------------------------------------+-------------------+

COMMUNICATIONS PROTOCOLS

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.VN2OF</td>
<td>RSX20F protocol</td>
</tr>
<tr>
<td>1</td>
<td>.VNMCB</td>
<td>MCB DECnet protocol</td>
</tr>
<tr>
<td>2</td>
<td>.VND60</td>
<td>DN60 (IBMCOM) protocol</td>
</tr>
<tr>
<td></td>
<td>.VDNDC</td>
<td>DDCMP (DECnet) protocol</td>
</tr>
<tr>
<td>3</td>
<td>.VNMGP</td>
<td>MDP (DDCM maintenance) protocol</td>
</tr>
<tr>
<td>4</td>
<td>.VNCNL</td>
<td>Controller loopback</td>
</tr>
<tr>
<td>5</td>
<td>.VNCBL</td>
<td>Cable loopback</td>
</tr>
</tbody>
</table>

156
DEVICE TYPES

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Symbol</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSK</td>
<td>disk structure</td>
<td>0</td>
<td>.DVDSK</td>
<td>no</td>
</tr>
<tr>
<td>MTA</td>
<td>magtape</td>
<td>2</td>
<td>.DVMTA</td>
<td>yes</td>
</tr>
<tr>
<td>MT</td>
<td>logical magtape</td>
<td>2</td>
<td>.DVMTA</td>
<td>yes</td>
</tr>
<tr>
<td>LPT</td>
<td>spoiled line printer</td>
<td>7</td>
<td>-</td>
<td>yes</td>
</tr>
<tr>
<td>PLPT</td>
<td>physical line printer</td>
<td>7</td>
<td>.DVLP</td>
<td>yes</td>
</tr>
<tr>
<td>CDR</td>
<td>spoiled card reader</td>
<td>10</td>
<td>-</td>
<td>yes</td>
</tr>
<tr>
<td>PCDR</td>
<td>physical card reader</td>
<td>10</td>
<td>.DVCR</td>
<td>yes</td>
</tr>
<tr>
<td>FE</td>
<td>front-end pseudo-device</td>
<td>11</td>
<td>.DVFE</td>
<td>no</td>
</tr>
<tr>
<td>TTY</td>
<td>terminal</td>
<td>12</td>
<td>.DVTTY</td>
<td>yes</td>
</tr>
<tr>
<td>PTY</td>
<td>pseudo-terminal</td>
<td>13</td>
<td>.DVPTY</td>
<td>yes</td>
</tr>
<tr>
<td>NUL</td>
<td>null device</td>
<td>15</td>
<td>.DVNU</td>
<td>no</td>
</tr>
<tr>
<td>NET</td>
<td>ARPA network</td>
<td>16</td>
<td>.DVNET</td>
<td>no</td>
</tr>
<tr>
<td>CDP</td>
<td>spoiled card punch</td>
<td>21</td>
<td>-</td>
<td>yes</td>
</tr>
<tr>
<td>PCDP</td>
<td>physical card punch</td>
<td>21</td>
<td>.DVCDP</td>
<td>yes</td>
</tr>
<tr>
<td>DCN</td>
<td>DECnet active component</td>
<td>22</td>
<td>.DVDCN</td>
<td>no</td>
</tr>
<tr>
<td>SRV</td>
<td>DECnet passive component</td>
<td>23</td>
<td>.DVSR</td>
<td>no</td>
</tr>
</tbody>
</table>

Device designator = <600000(.DVDES)+type>,<unit number>,<unit number> -1 if no units

Terminal designator = 0,<400000(.TTDES)+TTY number>

DIRECTORY PROTECTION FIELDS

<table>
<thead>
<tr>
<th>Value</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>DPXRD</td>
<td>Directory access controlled by individual file access</td>
</tr>
<tr>
<td>10</td>
<td>DPXCN</td>
<td>Connecting to directory and changing protection/account allowed</td>
</tr>
<tr>
<td>4</td>
<td>DPXCF</td>
<td>Creating files in directory allowed</td>
</tr>
</tbody>
</table>

FILE PROTECTION FIELDS

<table>
<thead>
<tr>
<th>Value</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>FPXRD</td>
<td>Read access</td>
</tr>
<tr>
<td>20</td>
<td>FPXWR</td>
<td>Write access</td>
</tr>
<tr>
<td>10</td>
<td>FPXEX</td>
<td>Execute access</td>
</tr>
<tr>
<td>4</td>
<td>FPXAPP</td>
<td>Append access</td>
</tr>
<tr>
<td>2</td>
<td>FPXDIR</td>
<td>Directory listing access</td>
</tr>
</tbody>
</table>
**FILE DESCRIPTOR BLOCK (FDB)**

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.FBHDR</td>
<td>FDB header word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO-28 Reserved for DIGITAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B29-35(FB%LEN) Length of this file's FDB</td>
</tr>
<tr>
<td>1</td>
<td>.FBCTL</td>
<td>BO(FB%TMP) File is temporary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(FB%PRM) File is permanent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(FB%EX) File does not exist (no file type)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B3(FB%DEL) File is deleted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B4(FB%NXF) File does not exist (not yet closed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B5(FB%LNG) File is longer than 512 pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B6(FB%SHT) Reserved for DIGITAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B7(FB%DIR) File is a directory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B8(FB%NDD) File is not to be backed-up</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B9(FB%BAT) File may have one or more bad pages</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B10(FB%DR) Directory has subdirectories</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B11(FB%ARG) File has archive status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B12(FB%INV) File is invisible</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B13(FB%OFF) File is offline</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B14-17(FB%CF) File class field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.(FBNRM) not RMS file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.(FBRMS) RMS file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18(FB%NDL) File cannot be deleted</td>
</tr>
<tr>
<td>2</td>
<td>.FBEXL</td>
<td>Link to FDB of next file with same name but different type</td>
</tr>
<tr>
<td>3</td>
<td>.FBADR</td>
<td>Disk address of file index block</td>
</tr>
<tr>
<td>4</td>
<td>.FBPRT</td>
<td>File access code: 500000.,&lt;access flags&gt;</td>
</tr>
<tr>
<td>5</td>
<td>.FBCRE</td>
<td>Date/time that file was closed after last write</td>
</tr>
<tr>
<td>6</td>
<td>.FBAUT</td>
<td>Pointer to file author string</td>
</tr>
<tr>
<td>7</td>
<td>.FBBGN</td>
<td>&lt;generation #&gt;,&lt;internal directory #&gt; if 1B7 of .FBCTL</td>
</tr>
<tr>
<td>10</td>
<td>.FBACT</td>
<td>Pointer to alphanumeric account designator string</td>
</tr>
<tr>
<td>11</td>
<td>.FBBV</td>
<td>File I/O flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO-5(FB%RET) Generation retention count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B6-11(FB%BSZ) File byte size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B14-17(FB%MCD) Data mode of last file open</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35(FB%PGC) File page count</td>
</tr>
<tr>
<td>12</td>
<td>.FBSIZ</td>
<td>Number of bytes in file</td>
</tr>
<tr>
<td>13</td>
<td>.FCRVR</td>
<td>File creation date/time</td>
</tr>
<tr>
<td>14</td>
<td>.FBWRT</td>
<td>Date/time of last user write</td>
</tr>
<tr>
<td>15</td>
<td>.FBWRT</td>
<td>Date/time of last non-write access</td>
</tr>
<tr>
<td>16</td>
<td>.FBCNT</td>
<td># of file writes,# of file references</td>
</tr>
<tr>
<td>17</td>
<td>.FBBKO</td>
<td>Used by DUMPER</td>
</tr>
<tr>
<td>20</td>
<td>.FBBK1</td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>21</td>
<td>.FBBK2</td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>22</td>
<td>.FBBBT</td>
<td>Flags,# file pages when deleted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(AR%RAR) User request for file archive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(AR%RIV) System request for involuntary file migration</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
File Descriptor Block

B3(AR%NDL)  Do not delete contents of file
when archiving
B4(AR%NAR)  Resist involuntary migration
B5(AR%EXM)  File exempt from involuntary
migration
B6(AR%1ST)  1st pass of archival-
collection run in progress
B7(AR%RFL)  Restore failed
B10(AR%WRN) Warn user of approaching
on-line expiration
B11-17(AR%RSN) Reason file was moved
offline
  .AREXP(1) File expired
  .ARRAR(2) Archiving was requested
  .ARRIR(3) Migration was requested
B18-35(AR%PSZ) 0,,<# file pages when
archived>

23  .FBNET On-line expiration date/time
24  .FBUSW User-settable word
25  .FBNL Address of FDB for next generation of file
26  .FBNAM Pointer to filename block
27  .FBEXT Pointer to file type block
30  .FBLWR Pointer to user-who-last-wrote string
31  .FBTDT Archive or collection tape-write date/time
32  .FBFET Offline expiration date/time
33  .FBTP1 Tape ID for first archive or collection run
34  .FBSS1 <1st tape save set #>,<1st tape file #>
35  .FBTP2 Tape ID for second archive or collection run
36  .FBSS2 <2nd tape save set #>,<2nd tape file #>

FORK (PROCESS) HANDLES

Value  Symbol  Meaning
400000  .FHSLF  Current process
400000+n  --  Process n, inferior to current process
             (relative fork handle)
-1  .FHSUP  Superior process
-2  .FHTOP  Top-level process
-3  .FHSAI  Current process and all inferiors
-4  .FHINF  All inferiors of current process
-5  .FHJOB  All processes in job

FLOATING-POINT FORMAT CONTROL

Bit  Symbol  Meaning
00-1  FL%SGN  Sign control for 1st field; 1st character
position used for minus for negative
numbers; for positive numbers, 1st character
position defined according to:
  0  .FLDIG  1st character is digit
TOPS-20 Monitor Calls Quick Reference Guide
Floating-Point Format Control

1. FLSPC 1st character is space
2. FLPLS 1st character is plus sign
3. FLSPA 1st character is space

B2-3 FL%JUS Justification control for 1st field
0. FLLLSP Right justify with leading spaces
1. FLLZR Right justify with leading 0's
2. FLLAS Right justify with leading asterisks
3. FLTSP Left justify with trailing spaces

B4 FL%ONE Output at least 1 digit in 1st field
B5 FL%DOL Prefix number with dollar sign ($)
B6 FL%PNT Output decimal point
B7-8 FL%EXP 3rd (exponent) field control
0. FLEXN No exponent field
1. FLEXE Output E as 1st character of exponent field
2. FLEXD Output D as 1st character of exponent field
3. FLEXM Output *10^ as 1st characters of exponent field

B9-10 FL%ESG Exponent sign control; 1st character position used for minus for negative exponents; for positive exponents, 1st character position defined according to:
0. FLDE 1st character after exponent prefix is digit
1. FLPL 1st character after prefix is plus sign
2. FLSP 1st character after prefix is space
3. FLDG 1st character after exponent prefix is digit

B11 FL%DVL Use free format on overflow of 1st field and expand exponent on overflow of 3rd field
B13-17 FL%RND Digit position at which rounding will occur; if 0, rounding occurs at 12th digit; if 37, no rounding occurs

B18-23 FL%FST Number of characters in 1st field, including $ if FL%DOL set
B24-29 FL%SNR Number of characters in 2nd field
B30-35 FL%THD Number of characters in 3rd field

I/O IDENTIFIERS

Symbol Meaning
LH,RH Job File Number (file handle)
.O,JFN Primary input designator
.PRIIN 0.100 Primary output designator
.PRIQU 0.101
.NULD 0.377777 Null designator
.TTDES 0.400000 Universal terminal designator
.CTRDM 0.777777 Process's controlling terminal
.DVDES 600000.xxx Universal device designator
.777777.xxx,777777 Universal default byte pointer

160
## JFN Mode Word

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>TT%OSP</td>
<td>Output suppress control (1 = ignore output; 0 = allow output)</td>
</tr>
<tr>
<td>B1</td>
<td>TT%FFF</td>
<td>Has mechanical form feed</td>
</tr>
<tr>
<td>B2</td>
<td>TT%TAB</td>
<td>Has mechanical tab</td>
</tr>
<tr>
<td>B3</td>
<td>TT%LCA</td>
<td>Has lower case</td>
</tr>
<tr>
<td>B4-10</td>
<td>TT%LEN</td>
<td>Page length</td>
</tr>
<tr>
<td>B11-17</td>
<td>TT%WID</td>
<td>Page width</td>
</tr>
<tr>
<td>B18-23</td>
<td>TT%WAK</td>
<td>Wakeup control on:</td>
</tr>
<tr>
<td>B18</td>
<td>not used</td>
<td></td>
</tr>
<tr>
<td>B19</td>
<td>TT%IGN</td>
<td>Ignore other TT%WAK bits</td>
</tr>
<tr>
<td>B20</td>
<td>TT%WIF</td>
<td>Formatting control character</td>
</tr>
<tr>
<td>B21</td>
<td>TT%WKN</td>
<td>Non-formatting control character</td>
</tr>
<tr>
<td>B22</td>
<td>TT%WKP</td>
<td>Punctuation character</td>
</tr>
<tr>
<td>B23</td>
<td>TT%WKA</td>
<td>Alphanumeric character</td>
</tr>
<tr>
<td>B24</td>
<td>TT%ECO</td>
<td>Echo on</td>
</tr>
<tr>
<td>B25</td>
<td>TT%ECM</td>
<td>Echo mode</td>
</tr>
<tr>
<td>B26</td>
<td>TT%ALK</td>
<td>Accept links</td>
</tr>
<tr>
<td>B27</td>
<td>TT%AAD</td>
<td>Accept advice</td>
</tr>
<tr>
<td>B28-29</td>
<td>TT%DAM</td>
<td>Terminal data mode</td>
</tr>
<tr>
<td>00</td>
<td>.TTBIN</td>
<td>No translation</td>
</tr>
<tr>
<td>01</td>
<td>.TTASC</td>
<td>Translate both echo and output</td>
</tr>
<tr>
<td>10</td>
<td>.TTAT0</td>
<td>Translate output only</td>
</tr>
<tr>
<td>11</td>
<td>.TTATE</td>
<td>Translate echo only</td>
</tr>
<tr>
<td>B30</td>
<td>TT%UCO</td>
<td>Upper case output control</td>
</tr>
<tr>
<td>0</td>
<td>Do not indicate</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Indicate by 'X</td>
<td></td>
</tr>
<tr>
<td>B31</td>
<td>TT%LIC</td>
<td>Lower case input control</td>
</tr>
<tr>
<td>0</td>
<td>No conversion</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Convert lower to upper</td>
<td></td>
</tr>
<tr>
<td>B32-33</td>
<td>TT%DOM</td>
<td>Duplex mode</td>
</tr>
<tr>
<td>00</td>
<td>.TTFDX</td>
<td>Full duplex</td>
</tr>
<tr>
<td>01</td>
<td>.THDXG</td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>10</td>
<td>.TLHDX</td>
<td>Character half duplex</td>
</tr>
<tr>
<td>11</td>
<td>.TLBDX</td>
<td>Line half duplex</td>
</tr>
<tr>
<td>B34</td>
<td>TT%PGM</td>
<td>Pause-on-command mode</td>
</tr>
<tr>
<td>0</td>
<td>Disable pause-on-command mode</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Enable pause-on-command mode</td>
<td></td>
</tr>
<tr>
<td>B35</td>
<td>TT%CAR</td>
<td>System carrier state; on if line is dataset and carrier is on</td>
</tr>
</tbody>
</table>

## Job Capability Word

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>SC%CTC</td>
<td>Process can enable for CTRL/C interrupts</td>
</tr>
<tr>
<td>B1</td>
<td>SC%GTA</td>
<td>Process can examine monitor tables with GETAB</td>
</tr>
<tr>
<td>B2</td>
<td>SC%LOG</td>
<td>Process can execute protected log functions</td>
</tr>
<tr>
<td>B6</td>
<td>SC%CTC</td>
<td>Process can change source of terminal interrupts for other processes</td>
</tr>
</tbody>
</table>

161
TOPS-20 Monitor Calls Quick Reference Guide
Job Capability Word

**B9-17 Inferior Process Capabilities**

**B9** SC\%SUP
Process can manipulate its superior process

**B17** SC\%FRZ
Unprocessed software interrupts can cause
process to be frozen instead of terminated

**B18-35 User Capabilities**

**B18** SC\%WHL
User has WHEEL capability

**B19** SC\%OPR
User has OPERATOR capability

**B20** SC\%CNF
User has CONFIDENTIAL INFORMATION ACCESS
capability

**B21** SC\%MNT
User has MAINTENANCE capability

**B22** SC\%IPC
User has IPCF capability

**B23** SC\%ENQ
User has ENQ/DEQ capability

**B24** SC\%WZ
User has NET WIZARD (ARPAnet) capability

**B25** SC\%NAS
User has ARPANET ABSOLUTE SOCKET capability

**B26** SC\%DNA
User has access to DECNET

**B27** SC\%ANA
User has access to ARPANET

**MAGTAPE DEVICE TYPES**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.MTT45</td>
<td>TU45 (default)</td>
</tr>
<tr>
<td>17</td>
<td>.MTT70</td>
<td>TU70</td>
</tr>
<tr>
<td>20</td>
<td>.MTT71</td>
<td>TU71</td>
</tr>
<tr>
<td>21</td>
<td>.MTT72</td>
<td>TU72</td>
</tr>
<tr>
<td>13</td>
<td>.MTT77</td>
<td>TU77</td>
</tr>
<tr>
<td>19</td>
<td>.MTT78</td>
<td>TU78</td>
</tr>
</tbody>
</table>

**MAGTAPE DRIVE TYPES**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.TMDR9</td>
<td>9-track tape drive</td>
</tr>
<tr>
<td>2</td>
<td>.TMDR7</td>
<td>7-track tape drive</td>
</tr>
</tbody>
</table>

**MAGTAPE HARDWARE DATA MODES**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SUDDM</td>
<td>Default system data mode</td>
</tr>
<tr>
<td>1</td>
<td>.SUDDC</td>
<td>Dump mode</td>
</tr>
<tr>
<td>2</td>
<td>.SUDDM</td>
<td>SIXBIT mode</td>
</tr>
<tr>
<td>3</td>
<td>.SUDDA</td>
<td>ANSI ASCII mode</td>
</tr>
<tr>
<td>4</td>
<td>.SUDDM</td>
<td>Industry compatible mode</td>
</tr>
<tr>
<td>5</td>
<td>.SUDDH</td>
<td>High-density mode (TU70, TU72 only)</td>
</tr>
</tbody>
</table>

162
### MAGTAPE LABEL STATES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.LSUNL</td>
<td>Unlabeled tape</td>
</tr>
<tr>
<td>1</td>
<td>.LSPRI</td>
<td>Private tape</td>
</tr>
<tr>
<td>2</td>
<td>.LSSCR</td>
<td>Scratch tape</td>
</tr>
<tr>
<td>3</td>
<td>.LSUSC</td>
<td>User scratch tape</td>
</tr>
</tbody>
</table>

### MAGTAPE LABEL TYPES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.LTUNL</td>
<td>Unlabeled</td>
</tr>
<tr>
<td>2</td>
<td>.LTANS</td>
<td>ANSI Standard label</td>
</tr>
<tr>
<td>3</td>
<td>.LTEBC</td>
<td>EBCDIC Standard label</td>
</tr>
<tr>
<td>4</td>
<td>.LTT20</td>
<td>TOPS-20 Standard label</td>
</tr>
</tbody>
</table>

### MAGTAPE RECORD SIZES

<table>
<thead>
<tr>
<th>Data Mode</th>
<th>Maximum Record Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>System-default Dump</td>
<td>--</td>
</tr>
<tr>
<td>SIXBIT</td>
<td>49152. bytes</td>
</tr>
<tr>
<td>ANSI ASCII</td>
<td>40960. bytes</td>
</tr>
<tr>
<td>Industry compatible</td>
<td>32768. bytes</td>
</tr>
<tr>
<td>High density</td>
<td>8192. bytes</td>
</tr>
</tbody>
</table>

### MAGTAPE RECORDING DENSITIES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SJDDN</td>
<td>Default system density</td>
</tr>
<tr>
<td>1</td>
<td>.SJDN2</td>
<td>200 BPI</td>
</tr>
<tr>
<td>2</td>
<td>.SJDN5</td>
<td>556 BPI</td>
</tr>
<tr>
<td>3</td>
<td>.SJDN8</td>
<td>800 BPI</td>
</tr>
<tr>
<td>4</td>
<td>.SJDN16</td>
<td>1600 BPI</td>
</tr>
<tr>
<td>5</td>
<td>.SJDN62</td>
<td>6250 BPI</td>
</tr>
</tbody>
</table>

### PHYSICAL CARD PUNCH (PCDP; ) STATUS BITS

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B10</td>
<td>M0%FER</td>
<td>Fatal error condition</td>
</tr>
<tr>
<td>B12</td>
<td>M0%EDF</td>
<td>All pending output has been processed</td>
</tr>
<tr>
<td>B13</td>
<td>M0%IOP</td>
<td>Output in progress</td>
</tr>
<tr>
<td>B14</td>
<td>M0%SER</td>
<td>Software error</td>
</tr>
<tr>
<td>B15</td>
<td>M0%HE</td>
<td>Hardware error</td>
</tr>
<tr>
<td>B16</td>
<td>M0%OL</td>
<td>Card-punch off-line</td>
</tr>
<tr>
<td>B17</td>
<td>M0%FNX</td>
<td>Card punch doesn’t exist</td>
</tr>
<tr>
<td>B32</td>
<td>M0%HEM</td>
<td>Stacker full or hopper empty</td>
</tr>
<tr>
<td>B33</td>
<td>M0%SCK</td>
<td>Stack check</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
PCDP: Status Bits

**B34  MO%PCK  Pick check**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>MO%COL</td>
<td>Device is on line</td>
</tr>
<tr>
<td>B10</td>
<td>MO%FER</td>
<td>Fatal hardware error</td>
</tr>
<tr>
<td>B12</td>
<td>MO%EOF</td>
<td>Card reader at EOF</td>
</tr>
<tr>
<td>B13</td>
<td>MO%IOP</td>
<td>I/O in progress</td>
</tr>
<tr>
<td>B14</td>
<td>MO%SER</td>
<td>Software error</td>
</tr>
<tr>
<td>B15</td>
<td>MO%HE</td>
<td>Hardware error</td>
</tr>
<tr>
<td>B16</td>
<td>MO%OL</td>
<td>Device is off line</td>
</tr>
<tr>
<td>B17</td>
<td>MO%FNX</td>
<td>Device is nonexistent</td>
</tr>
<tr>
<td>B31</td>
<td>MO%SFL</td>
<td>Output stacker full</td>
</tr>
<tr>
<td>B32</td>
<td>MO%HEM</td>
<td>Input hopper empty</td>
</tr>
<tr>
<td>B33</td>
<td>MO%CK</td>
<td>Stack check</td>
</tr>
<tr>
<td>B34</td>
<td>MO%PCK</td>
<td>Pick check</td>
</tr>
<tr>
<td>B35</td>
<td>MO%RCK</td>
<td>Read check</td>
</tr>
</tbody>
</table>

**PHYSICAL LINE PRINTER (PLPT): CONTROL CHARACTERS**

<table>
<thead>
<tr>
<th>ASCII Code</th>
<th>Default Channel</th>
<th>Name</th>
<th>Default Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td>HT (^I)</td>
<td>Skips to beginning of every 8th column on same line</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>LF (^J)</td>
<td>Skips to column 1 on next line; skips last 6 lines of page</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>VT (^K)</td>
<td>Skips to column 1 on line at next third of page</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>FF (^L)</td>
<td>Skips to column 1 on top of next page</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>CR (^M)</td>
<td>Returns to column 1 of current line; no paper advance</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>Half page</td>
<td>Skips to column 1 on next half page</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>Alternate lines</td>
<td>Skips to column 1 on next even line</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>Three lines</td>
<td>Skips to column 1 on every third line</td>
</tr>
<tr>
<td>23</td>
<td>5</td>
<td>Next line</td>
<td>Skips to column 1 on next line; fills last 6 lines of page</td>
</tr>
<tr>
<td>24</td>
<td>6</td>
<td>Sixth page</td>
<td>Skips to column 1 on next sixth of page</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
PLPT: Status Bits

**PHYSICAL LINE PRINTER (PLPT:) STATUS BITS**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>M0%LCP</td>
<td>Lower case printer</td>
</tr>
<tr>
<td>B10</td>
<td>M0%FER</td>
<td>Fatal hardware error</td>
</tr>
<tr>
<td>B12</td>
<td>M0%EOF</td>
<td>All data sent to printer has been printed</td>
</tr>
<tr>
<td>B13</td>
<td>M0%IDP</td>
<td>I/O in progress</td>
</tr>
<tr>
<td>B14</td>
<td>M0%SER</td>
<td>Software error</td>
</tr>
<tr>
<td>B15</td>
<td>M0%HE</td>
<td>Hardware error</td>
</tr>
<tr>
<td>B16</td>
<td>M0%DL</td>
<td>Device is off line</td>
</tr>
<tr>
<td>B17</td>
<td>M0%FNX</td>
<td>Device is nonexistent</td>
</tr>
<tr>
<td>B30</td>
<td>M0%RPE</td>
<td>RAM parity error</td>
</tr>
<tr>
<td>B31</td>
<td>M0%LVU</td>
<td>Optical VFU</td>
</tr>
<tr>
<td>B33</td>
<td>M0%LVF</td>
<td>VFU error</td>
</tr>
<tr>
<td>B34</td>
<td>M0%LCI</td>
<td>Character interrupt</td>
</tr>
<tr>
<td>B35</td>
<td>M0%LPC</td>
<td>Page counter register overflow</td>
</tr>
</tbody>
</table>

**PHYSICAL MAGTAPE (MTA:) STATUS BITS**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B16</td>
<td>M1%ILW</td>
<td>Drive is write protected</td>
</tr>
<tr>
<td>B19</td>
<td>M1%DVE</td>
<td>Device error (hung or data late)</td>
</tr>
<tr>
<td>B20</td>
<td>M1%DAE</td>
<td>Data error</td>
</tr>
<tr>
<td>B21</td>
<td>M1%SER</td>
<td>Suppress automatic error recovery procedures</td>
</tr>
<tr>
<td>B22</td>
<td>M1%EOF</td>
<td>Device EOF (file) mark</td>
</tr>
<tr>
<td>B23</td>
<td>M1%IRL</td>
<td>Incorrect record length</td>
</tr>
<tr>
<td>B24</td>
<td>M1%DOT</td>
<td>Beginning of tape</td>
</tr>
<tr>
<td>B25</td>
<td>M1%DOT</td>
<td>End of tape</td>
</tr>
<tr>
<td>B26</td>
<td>M1%EVP</td>
<td>Even parity</td>
</tr>
<tr>
<td>B29-31</td>
<td>M1%CC</td>
<td>Character counter if M1%IRL on</td>
</tr>
<tr>
<td>B32</td>
<td>M1%NSH</td>
<td>Selected data mode or density not supported by hardware</td>
</tr>
</tbody>
</table>

**SOFTWARE DATA MODES**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Symbol</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.GSRNM</td>
<td>Normal mode - allows unit-record output</td>
</tr>
<tr>
<td>1</td>
<td>.GSSMB</td>
<td>Small Buffer mode - allows small data segments to be transmitted to terminals</td>
</tr>
<tr>
<td>10</td>
<td>.GISMG</td>
<td>Image mode - sends an &quot;image&quot; of each byte (12-bit)</td>
</tr>
<tr>
<td>17</td>
<td>.GSDMP</td>
<td>Dump mode - unbuffered by default</td>
</tr>
</tbody>
</table>
SOFTWARE INTERRUPT CHANNELS

<table>
<thead>
<tr>
<th>Channel</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5</td>
<td>.ICADV</td>
<td>Assignable by user program</td>
</tr>
<tr>
<td>6</td>
<td>.ICFOV</td>
<td>Arithmetic overflow (includes NDDIV)</td>
</tr>
<tr>
<td>7</td>
<td>.ICFPO</td>
<td>Arithmetic floating point overflow (includes FXU)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>9.*</td>
<td>.ICPOV</td>
<td>Pushdown list (PDL) overflow</td>
</tr>
<tr>
<td>10.</td>
<td>.ICEOF</td>
<td>End of file condition</td>
</tr>
<tr>
<td>11.*</td>
<td>.ICDAE</td>
<td>Data error file condition</td>
</tr>
<tr>
<td>12.*</td>
<td>.ICQTA</td>
<td>Disk full or quota exceeded when creating new page</td>
</tr>
<tr>
<td>13.-14.</td>
<td></td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>15.*</td>
<td>.ICILI</td>
<td>Illegal instruction</td>
</tr>
<tr>
<td>16.*</td>
<td>.ICIRD</td>
<td>Illegal memory read</td>
</tr>
<tr>
<td>17.*</td>
<td>.ICIWR</td>
<td>Illegal memory write</td>
</tr>
<tr>
<td>18.</td>
<td></td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>19.</td>
<td>.ICIFT</td>
<td>Inferior process termination or forced freeze</td>
</tr>
<tr>
<td>20.*</td>
<td>.ICMSE</td>
<td>System resources exhausted</td>
</tr>
<tr>
<td>21.</td>
<td></td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td>22.</td>
<td>.ICNXP</td>
<td>Reference to non-existent page</td>
</tr>
<tr>
<td>23.-35.</td>
<td></td>
<td>Assignable by user program</td>
</tr>
</tbody>
</table>

* Channels are panic channels and cannot be completely deactivated

SYSTEM PIDS

<table>
<thead>
<tr>
<th>PID</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.SPIPC</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>.SPINF</td>
<td>PID of &lt;SYSTEM&gt;INFD</td>
</tr>
<tr>
<td>2</td>
<td>.SPSR</td>
<td>PID of QUASAR</td>
</tr>
<tr>
<td>3</td>
<td>.SPMDA</td>
<td>PID of QSMMDA</td>
</tr>
<tr>
<td>4</td>
<td>.SPDPR</td>
<td>PID of ORION</td>
</tr>
<tr>
<td>5</td>
<td>.SPNSR</td>
<td>PID of NETSER</td>
</tr>
</tbody>
</table>
### SYSTEM TABLES

<table>
<thead>
<tr>
<th>Name</th>
<th>Index</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>APRID</td>
<td></td>
<td>Processor serial number</td>
</tr>
<tr>
<td>BLTD</td>
<td></td>
<td>Date and time system was generated</td>
</tr>
<tr>
<td>DEBUGW</td>
<td></td>
<td>Debugging information</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>State of operator coverage</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>State of BUGCHK handling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unattended</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Attended</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Debugging</td>
</tr>
<tr>
<td></td>
<td></td>
<td>State of BUGCHK handling</td>
</tr>
<tr>
<td>DEVCHR</td>
<td>(P1)</td>
<td>Device characteristics word</td>
</tr>
<tr>
<td>DEVNAM</td>
<td>(P1)</td>
<td>SIXBIT device name including unit number</td>
</tr>
<tr>
<td>DEVUNT</td>
<td>(P1)</td>
<td>BO-17 Job # to which device is assigned; -1 if device is not assigned;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2 if reserved for device allocator</td>
</tr>
<tr>
<td>DRMERR</td>
<td></td>
<td>Unit #: -1 if device has no units</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Information on drum errors</td>
</tr>
<tr>
<td>DSKERR</td>
<td></td>
<td>Information on disk errors</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Number of recoverable disk errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Varies depending on type of disk being used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Varies depending on type of disk being used</td>
</tr>
<tr>
<td>DWNTIM</td>
<td></td>
<td>Downtime information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Date/time of next scheduled system shutdown</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Date/time when system will return</td>
</tr>
<tr>
<td>HOLAV</td>
<td></td>
<td>High queue load averages</td>
</tr>
<tr>
<td>IMPLT1</td>
<td>(P2)</td>
<td>ARPA#ET: 1 fullword for each link</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 receive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 send</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 free</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01 delete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal connection index for NETAWD, NETBAL, NETBTC, NETBUF, NETFSK, NETLSK, NETSTS</td>
</tr>
<tr>
<td>B18-19</td>
<td></td>
<td>receive</td>
</tr>
<tr>
<td>B20-27</td>
<td></td>
<td>Link number</td>
</tr>
<tr>
<td>IMPLT2</td>
<td>(P2)</td>
<td>ARPA#ET: 1 fullword for each link</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 Flags</td>
</tr>
<tr>
<td>B10-17</td>
<td></td>
<td>Byte size of buffer</td>
</tr>
<tr>
<td>B18-35</td>
<td></td>
<td>Address of input buffer</td>
</tr>
<tr>
<td>IMPLT3</td>
<td>(P2)</td>
<td>ARPA#ET: 1 fullword for each link</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 Address of output buffer</td>
</tr>
<tr>
<td>B18-35</td>
<td></td>
<td>Message saved for retransmission</td>
</tr>
<tr>
<td>IMPLT4</td>
<td>(P2)</td>
<td>ARPA#ET: 1 fullword for each link</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 Address of current buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Message allocation in bits</td>
</tr>
<tr>
<td>JBO#</td>
<td></td>
<td>Owning job for CRJDB-created jobs</td>
</tr>
</tbody>
</table>
### TOPS-20 Monitor Calls Quick Reference Guide

#### System Tables

<table>
<thead>
<tr>
<th><strong>JOBNAM</strong></th>
<th><strong>Job #</strong></th>
<th><strong>BO-17</strong></th>
<th>Reserved for DIGITAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JOBPNM</strong></td>
<td><strong>Job #</strong></td>
<td><strong>B18-35</strong></td>
<td>Index into system program tables of system program being used by job</td>
</tr>
<tr>
<td><strong>JOBRT</strong></td>
<td><strong>Job #</strong></td>
<td><strong>SIXBIT name of program running in this job</strong></td>
<td></td>
</tr>
<tr>
<td><strong>JDBTTY</strong></td>
<td><strong>Job #</strong></td>
<td><strong>BO-17</strong></td>
<td>Controlling terminal line #; or -1 if none</td>
</tr>
<tr>
<td><strong>LOGDSE</strong></td>
<td><strong>Job #</strong></td>
<td><strong>B18-35</strong></td>
<td>Reserved for DIGITAL</td>
</tr>
<tr>
<td><strong>LOGDSE</strong></td>
<td><strong>Logging information</strong></td>
<td>0</td>
<td>Designator for logging information</td>
</tr>
<tr>
<td><strong>LQLAV</strong></td>
<td><strong>Designator for Job 0 and error information</strong></td>
<td>1</td>
<td>Designator for logging information</td>
</tr>
<tr>
<td><strong>NETHST</strong></td>
<td><strong>ARPANET: 1 full word for each internal connection; -1 if no foreign host</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETAWD</strong></td>
<td><strong>ARPANET: 1 full word for each internal connection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETBAL</strong></td>
<td><strong>ARPANET: number of bits allocated to each internal connection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETBTC</strong></td>
<td><strong>ARPANET: byte count statistics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETBUF</strong></td>
<td><strong>ARPANET: 1 fullword for each internal connection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETFSK</strong></td>
<td><strong>ARPANET: foreign socket number (32 bits) for each internal connection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETLSK</strong></td>
<td><strong>ARPANET: local socket number for each internal connection</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>ARPANET: operational status table</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>Flags for NETSER</strong></td>
<td>0</td>
<td>IMP down</td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>IMP going down</strong></td>
<td>&gt;0</td>
<td>IMP up</td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>IMP on</strong></td>
<td>-1</td>
<td>Network off</td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>Network on</strong></td>
<td>0</td>
<td>Network on</td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>Flags for NETSER</strong></td>
<td>2</td>
<td>Flags for NETSER</td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>Time of last NCP cycle up</strong></td>
<td>3</td>
<td>Last IMP going down message</td>
</tr>
<tr>
<td><strong>NETRDY</strong></td>
<td><strong>BO-15</strong></td>
<td>Reserved for DIGITAL</td>
<td></td>
</tr>
<tr>
<td><strong>B16-17</strong></td>
<td><strong>Panic</strong></td>
<td>0</td>
<td>Panic</td>
</tr>
<tr>
<td><strong>B16-17</strong></td>
<td><strong>Scheduled hardware PM</strong></td>
<td>1</td>
<td>Scheduled hardware PM</td>
</tr>
<tr>
<td><strong>B16-17</strong></td>
<td><strong>Software reload</strong></td>
<td>2</td>
<td>Software reload</td>
</tr>
<tr>
<td><strong>B16-17</strong></td>
<td><strong>Emergency restart</strong></td>
<td>3</td>
<td>Emergency restart</td>
</tr>
<tr>
<td><strong>B18-31</strong></td>
<td><strong># of 5-minute intervals before IMP goes down</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>B22-31</strong></td>
<td><strong># of 5-minute intervals IMP will be down</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NCPGS</strong></td>
<td><strong>Number of pages of real (physical) user core available in system (1 word)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NSWPGS</strong></td>
<td><strong>Default swapping pages</strong></td>
<td>168</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTYPAR</td>
<td>Pseudo-TTY parameter information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0-17 Number of PTYs in system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18-35</td>
<td>TTY number of first PTY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QTIMES</td>
<td>Accumulated runtime of jobs on n scheduler queues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNAME</td>
<td>SIXBIT name of system program; 0 if entry unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SBLKS</td>
<td>Number of samples in working set size integral</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPFLTS</td>
<td>Total number of page faults of system program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSIZE</td>
<td>Time integral of working set size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STIMES</td>
<td>Total runtime of system program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYMTAB</td>
<td>SIXBIT table names of all GETAB tables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTAT</td>
<td>Monitor statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Time with no runnable jobs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Waiting time with 1 or more runnable jobs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Time spent in scheduler</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Time spent processing pager traps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Number of drum reads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Number of drum writes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Number of disk reads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Number of disk writes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Number of terminal wakeups</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Number of terminal interrupts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Time integral of number of processes in balance set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Time integral of number of runnable processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Exponential 1-minute average of number of runnable processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Exponential 5-minute average of number of runnable processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Exponential 15-minute average of number of runnable processes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Time integral of number of processes waiting for disk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Time integral of number of processes waiting for drum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Number of terminal input characters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Number of terminal output characters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Number of system core management cycles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Time spent doing postpurging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Number of forced balance set process removals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Time integral of number of processes in swap wait</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Scheduler overhead time in high precision units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Idle time in high precision units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Lost time in high precision units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>User time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Time integral of number of processes on high queue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Time integral of number of processes on low queue</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TDPS-20 Monitor Calls Quick Reference Guide
System Tables

35  Sum of process disk-write waits
36  Number of forced adjustments to balance set
37  Integral of number of reserve pages of all
    processes in memory
40  Integral of number of pages on replaceable
    queue
41  High precision pager trap time
42  Number of context switches
43  Time spent on background tasks
44  Total system page traps
45  Total saves from replacement queue
46  Number of pages removed from memory during
    system-wide garbage collection
47  Integral of number of working sets in
    memory
50  Integral of number of wait time without
    swap waits
51  Count of working set loads
52  Count of runnable processes removed from
    balance set
53  Number of pages removed from memory during
    process-wide garbage collection
SYSVER
    ASCIZ string identifying system name, version, and date
TICKPS
    Number of clock ticks per second
TTYJDB  line #
    B0-17  job # for which this is controlling
    terminal:
    -1 for unassigned line;
    -2 for line currently being
    assigned; or
    B18-35  job # to which line is assigned
    # -1 if some process is waiting for
    input from terminal

(Pn) specifies a set of parallel tables where n is a
    unique identifier of the set
    1  specifies an index into a table derived from
        B24-35 of NETAWD
    1  specifies an index into a table derived from
        B0-17 of IMPLTL

TERMINAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Number</th>
<th>Terminal</th>
<th>Symbol</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TTY model 33</td>
<td>.TT33</td>
<td>UPPERCASE only; padding after TAB and FF; page width 72, page length 66.</td>
</tr>
<tr>
<td>1</td>
<td>TTY model 35</td>
<td>.TT35</td>
<td>Mechanical FF and TAB; UPPERCASE only; padding after TAB and FF; page</td>
</tr>
</tbody>
</table>

170
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>TTY model 37 .TT37 width 72., page length 66. Lowercase; padding after TAB and FF; page width 72., page length 66.</td>
</tr>
<tr>
<td>3</td>
<td>TI/EXECUPORT .TTEXE Lowercase; padding after CR only; page width 80., page length 66.</td>
</tr>
<tr>
<td>4-7</td>
<td>Reserved for customer</td>
</tr>
<tr>
<td>8</td>
<td>Default .TTDEF Lowercase; full padding; page width 72., page length 66.</td>
</tr>
<tr>
<td>9</td>
<td>Ideal .TTIDL Mechanical FF and TAB; lowercase; no padding; no specified width or length</td>
</tr>
<tr>
<td>10</td>
<td>VT05 .TTV05 Mechanical TAB; UPPERCASE only; padding after LF and FF; page width 72., page length 20.; cursor control UPPERCASE only; no padding; page width 80., page length 12.; cursor control</td>
</tr>
<tr>
<td>11</td>
<td>VT50 .TTV50 Lowercase; no padding; page width 80., page length 30.</td>
</tr>
<tr>
<td>12</td>
<td>LA30 .TTL30 UPPERCASE only; full padding; page width 80., page length 66.</td>
</tr>
<tr>
<td>13</td>
<td>GT40 .TTG40 Lowercase; no padding; page width 80., page length 30.</td>
</tr>
<tr>
<td>14</td>
<td>LA36 .TTL36 Lowercase; no padding; page width 132., page length 66.</td>
</tr>
<tr>
<td>15</td>
<td>VT52 .TTV52 Mechanical TAB; lowercase; no padding; page width 80., page length 24.</td>
</tr>
<tr>
<td>16</td>
<td>VT100 .TT100 Mechanical TAB; lowercase; no padding; page width 80., page length 24.; cursor control</td>
</tr>
<tr>
<td>17</td>
<td>LA38 .TTL38 Mechanical TAB; lowercase; no padding; page width 132., page length 66.</td>
</tr>
<tr>
<td>18</td>
<td>LA120 .TT120 Mechanical FF and TAB; lowercase; no padding; page width 132., page length 60.</td>
</tr>
<tr>
<td>35</td>
<td>VT125 .TT125 Mechanical TAB; lowercase; no padding; page width 80., page length 24.; cursor control; graphics capabilities</td>
</tr>
<tr>
<td>36</td>
<td>VK100 .TTK10 Mechanical TAB; lowercase; no padding; page width 80., page length 24.; cursor control; color graphics capabilities</td>
</tr>
</tbody>
</table>
**Terminal Characteristics**

<table>
<thead>
<tr>
<th>Mechanical FF and TAB</th>
<th>Page width and length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lowercase</td>
<td>Padding after mechanical FF</td>
</tr>
<tr>
<td>Padding after CR</td>
<td>Cursor control</td>
</tr>
<tr>
<td>Padding after LF</td>
<td>Graphics capabilities</td>
</tr>
<tr>
<td>Padding after mechanical TAB</td>
<td></td>
</tr>
</tbody>
</table>

**TERMINAL INTERRUPT CODES**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Character/Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.TICBK</td>
<td>CTRL/@ or BREAK</td>
</tr>
<tr>
<td>1</td>
<td>.TICCA</td>
<td>CTRL/A</td>
</tr>
<tr>
<td>2</td>
<td>.TICCB</td>
<td>CTRL/B</td>
</tr>
<tr>
<td>3</td>
<td>.TICCC</td>
<td>CTRL/C</td>
</tr>
<tr>
<td>4</td>
<td>.TICCD</td>
<td>CTRL/D</td>
</tr>
<tr>
<td>5</td>
<td>.TICCE</td>
<td>CTRL/E</td>
</tr>
<tr>
<td>6</td>
<td>.TICCF</td>
<td>CTRL/F</td>
</tr>
<tr>
<td>7</td>
<td>.TICCG</td>
<td>CTRL/G</td>
</tr>
<tr>
<td>8</td>
<td>.TICCH</td>
<td>CTRL/H</td>
</tr>
<tr>
<td>9</td>
<td>.TICCI</td>
<td>CTRL/I (TAB)</td>
</tr>
<tr>
<td>10</td>
<td>.TICCJ</td>
<td>CTRL/J (LF)</td>
</tr>
<tr>
<td>11</td>
<td>.TICCK</td>
<td>CTRL/K (vertical TAB)</td>
</tr>
<tr>
<td>12</td>
<td>.TICCL</td>
<td>CTRL/L (FF)</td>
</tr>
<tr>
<td>13</td>
<td>.TICCM</td>
<td>CTRL/M (CR)</td>
</tr>
<tr>
<td>14</td>
<td>.TICCN</td>
<td>CTRL/N</td>
</tr>
<tr>
<td>15</td>
<td>.TICCO</td>
<td>CTRL/O</td>
</tr>
<tr>
<td>16</td>
<td>.TICCP</td>
<td>CTRL/P</td>
</tr>
<tr>
<td>17</td>
<td>.TICCQ</td>
<td>CTRL/Q</td>
</tr>
<tr>
<td>18</td>
<td>.TICCR</td>
<td>CTRL/R</td>
</tr>
<tr>
<td>19</td>
<td>.TICCS</td>
<td>CTRL/S</td>
</tr>
<tr>
<td>20</td>
<td>.TICCT</td>
<td>CTRL/T</td>
</tr>
<tr>
<td>21</td>
<td>.TICCU</td>
<td>CTRL/U</td>
</tr>
<tr>
<td>22</td>
<td>.TICCV</td>
<td>CTRL/V</td>
</tr>
<tr>
<td>23</td>
<td>.TICCW</td>
<td>CTRL/W</td>
</tr>
<tr>
<td>24</td>
<td>.TICCY</td>
<td>CTRL/X</td>
</tr>
<tr>
<td>25</td>
<td>.TICCZ</td>
<td>CTRL/Z</td>
</tr>
<tr>
<td>26</td>
<td>.TICES</td>
<td>ESCAPE (altmode)</td>
</tr>
<tr>
<td>27</td>
<td>.TICRB</td>
<td>DELETE (RUBOUT)</td>
</tr>
<tr>
<td>28</td>
<td>.TICSP</td>
<td>SPACE</td>
</tr>
<tr>
<td>30</td>
<td>.TICRF</td>
<td>Dataset carrier off</td>
</tr>
<tr>
<td>31</td>
<td>.TICTI</td>
<td>Typein</td>
</tr>
<tr>
<td>32</td>
<td>.TICTO</td>
<td>Typeout</td>
</tr>
<tr>
<td>33</td>
<td>-35</td>
<td>Reserved for DIGITAL</td>
</tr>
</tbody>
</table>

172
<table>
<thead>
<tr>
<th>Zone Name</th>
<th>Abbreviation</th>
<th>Left Half</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREENWICH DAYLIGHT TIME</td>
<td>GDT</td>
<td>700000</td>
</tr>
<tr>
<td>GREENWICH MEAN TIME</td>
<td>GMT</td>
<td>500000</td>
</tr>
<tr>
<td>GREENWICH STANDARD TIME</td>
<td>GST</td>
<td>500000</td>
</tr>
<tr>
<td>ATLANTIC DAYLIGHT TIME</td>
<td>ADT</td>
<td>700004</td>
</tr>
<tr>
<td>ATLANTIC STANDARD TIME</td>
<td>AST</td>
<td>500004</td>
</tr>
<tr>
<td>EASTERN DAYLIGHT TIME</td>
<td>EDT</td>
<td>700005</td>
</tr>
<tr>
<td>EASTERN STANDARD TIME</td>
<td>EST</td>
<td>500005</td>
</tr>
<tr>
<td>CENTRAL DAYLIGHT TIME</td>
<td>CDT</td>
<td>700006</td>
</tr>
<tr>
<td>CENTRAL STANDARD TIME</td>
<td>CST</td>
<td>500006</td>
</tr>
<tr>
<td>MOUNTAIN DAYLIGHT TIME</td>
<td>MDT</td>
<td>700007</td>
</tr>
<tr>
<td>MOUNTAIN STANDARD TIME</td>
<td>MST</td>
<td>500007</td>
</tr>
<tr>
<td>PACIFIC DAYLIGHT TIME</td>
<td>PDT</td>
<td>700010</td>
</tr>
<tr>
<td>PACIFIC STANDARD TIME</td>
<td>PST</td>
<td>500010</td>
</tr>
<tr>
<td>YUKON DAYLIGHT TIME</td>
<td>YDT</td>
<td>700011</td>
</tr>
<tr>
<td>YUKON STANDARD TIME</td>
<td>YST</td>
<td>500011</td>
</tr>
<tr>
<td>ALASKA-HAWAII DAYLIGHT TIME</td>
<td>MDT</td>
<td>700012</td>
</tr>
<tr>
<td>ALASKA-HAWAII STANDARD TIME</td>
<td>MST</td>
<td>500012</td>
</tr>
<tr>
<td>BERING DAYLIGHT TIME</td>
<td>BST</td>
<td>700013</td>
</tr>
<tr>
<td>BERING STANDARD TIME</td>
<td>BST</td>
<td>500013</td>
</tr>
<tr>
<td>LOCAL DAYLIGHT TIME</td>
<td>DAYLIGHT</td>
<td>600000</td>
</tr>
</tbody>
</table>

173
### TOPS-20 JSYS ERROR CODES

**Note**

See [TOPS-20 JSYS ERROR MNEMONICS](#) for error strings.

<table>
<thead>
<tr>
<th>Code</th>
<th>Mnemonic</th>
<th>Code</th>
<th>Mnemonic</th>
<th>Code</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>600010</td>
<td>LGINX1</td>
<td>600011</td>
<td>LGINX2</td>
<td>600012</td>
<td>LGINX3</td>
</tr>
<tr>
<td>600013</td>
<td>LGINX4</td>
<td>600014</td>
<td>LGINX5</td>
<td>600020</td>
<td>CRJBX1</td>
</tr>
<tr>
<td>600021</td>
<td>CRJBX2</td>
<td>600023</td>
<td>CRJBX4</td>
<td>600024</td>
<td>CRJBX5</td>
</tr>
<tr>
<td>600025</td>
<td>CRJBX6</td>
<td>600035</td>
<td>LOUTX1</td>
<td>600036</td>
<td>LOUTX2</td>
</tr>
<tr>
<td>600045</td>
<td>CACTX1</td>
<td>600046</td>
<td>CACTX2</td>
<td>600055</td>
<td>GUXF1</td>
</tr>
<tr>
<td>600056</td>
<td>GUXF2</td>
<td>600057</td>
<td>GUXF3</td>
<td>600060</td>
<td>GUXF4</td>
</tr>
<tr>
<td>600061</td>
<td>GUXF5</td>
<td>600062</td>
<td>GUXF6</td>
<td>600063</td>
<td>GUXF7</td>
</tr>
<tr>
<td>600064</td>
<td>GUXF8</td>
<td>600065</td>
<td>GUXF9</td>
<td>600066</td>
<td>GUXF10</td>
</tr>
<tr>
<td>600067</td>
<td>GUXF11</td>
<td>600070</td>
<td>GUXF12</td>
<td>600071</td>
<td>GUXF13</td>
</tr>
<tr>
<td>600072</td>
<td>GUXF14</td>
<td>600073</td>
<td>GUXF15</td>
<td>600074</td>
<td>GUXF16</td>
</tr>
<tr>
<td>600075</td>
<td>GUXF17</td>
<td>600076</td>
<td>GUXF18</td>
<td>600077</td>
<td>GUXF19</td>
</tr>
<tr>
<td>600100</td>
<td>GUXF20</td>
<td>600101</td>
<td>GUXF21</td>
<td>600102</td>
<td>GUXF22</td>
</tr>
<tr>
<td>600103</td>
<td>GUXF23</td>
<td>600104</td>
<td>GUXF24</td>
<td>600107</td>
<td>GUXF27</td>
</tr>
<tr>
<td>600110</td>
<td>GUXF28</td>
<td>600112</td>
<td>GUXF30</td>
<td>600113</td>
<td>GUXF31</td>
</tr>
<tr>
<td>600114</td>
<td>GUXF32</td>
<td>600115</td>
<td>GUXF33</td>
<td>600116</td>
<td>GUXF34</td>
</tr>
<tr>
<td>600117</td>
<td>GUXF35</td>
<td>600120</td>
<td>OPNX1</td>
<td>600121</td>
<td>OPNX2</td>
</tr>
<tr>
<td>600122</td>
<td>OPNX3</td>
<td>600123</td>
<td>OPNX4</td>
<td>600124</td>
<td>OPNX5</td>
</tr>
<tr>
<td>600125</td>
<td>OPNX6</td>
<td>600126</td>
<td>OPNX7</td>
<td>600127</td>
<td>OPNX8</td>
</tr>
<tr>
<td>600130</td>
<td>OPNX9</td>
<td>600131</td>
<td>OPNX10</td>
<td>600133</td>
<td>OPNX12</td>
</tr>
<tr>
<td>600134</td>
<td>OPNX13</td>
<td>600135</td>
<td>OPNX14</td>
<td>600136</td>
<td>OPNX15</td>
</tr>
<tr>
<td>600137</td>
<td>OPNX16</td>
<td>600140</td>
<td>OPNX17</td>
<td>600141</td>
<td>OPNX18</td>
</tr>
<tr>
<td>600142</td>
<td>OPNX19</td>
<td>600143</td>
<td>OPNX20</td>
<td>600144</td>
<td>OPNX21</td>
</tr>
<tr>
<td>600145</td>
<td>OPNX22</td>
<td>600150</td>
<td>DESX1</td>
<td>600151</td>
<td>DESX2</td>
</tr>
<tr>
<td>600152</td>
<td>DESX3</td>
<td>600153</td>
<td>DESX4</td>
<td>600154</td>
<td>DESX5</td>
</tr>
<tr>
<td>600155</td>
<td>DESX6</td>
<td>600156</td>
<td>DESX7</td>
<td>600157</td>
<td>DESX8</td>
</tr>
<tr>
<td>600160</td>
<td>CLSX1</td>
<td>600161</td>
<td>CLSX2</td>
<td>600165</td>
<td>RUXNX1</td>
</tr>
<tr>
<td>600166</td>
<td>RUXNX2</td>
<td>600167</td>
<td>RUXNX3</td>
<td>600170</td>
<td>RUXNX4</td>
</tr>
<tr>
<td>600175</td>
<td>SFPTX1</td>
<td>600176</td>
<td>SFPTX2</td>
<td>600177</td>
<td>SFPTX3</td>
</tr>
<tr>
<td>600200</td>
<td>CSIDX1</td>
<td>600204</td>
<td>CSIDX5</td>
<td>600210</td>
<td>SFBSX1</td>
</tr>
<tr>
<td>600211</td>
<td>SFBSX2</td>
<td>600215</td>
<td>IDX1</td>
<td>600216</td>
<td>IDX2</td>
</tr>
<tr>
<td>600217</td>
<td>IDX3</td>
<td>600220</td>
<td>IDX4</td>
<td>600221</td>
<td>IDX5</td>
</tr>
<tr>
<td>600222</td>
<td>IDX6</td>
<td>600240</td>
<td>PMAPX1</td>
<td>600241</td>
<td>PMAPX2</td>
</tr>
<tr>
<td>600245</td>
<td>SPACX1</td>
<td>600250</td>
<td>FRKHX1</td>
<td>600251</td>
<td>FRKHX2</td>
</tr>
<tr>
<td>600252</td>
<td>FRKHX3</td>
<td>600253</td>
<td>FRKHX4</td>
<td>600254</td>
<td>FRKHX5</td>
</tr>
<tr>
<td>600255</td>
<td>FRKHX6</td>
<td>600260</td>
<td>SPLFX1</td>
<td>600261</td>
<td>SPLFX2</td>
</tr>
<tr>
<td>600262</td>
<td>SPLFX3</td>
<td>600267</td>
<td>GTABX1</td>
<td>600270</td>
<td>GTABX2</td>
</tr>
<tr>
<td>600271</td>
<td>GTABX3</td>
<td>600279</td>
<td>RUNTX1</td>
<td>600275</td>
<td>STAX1</td>
</tr>
<tr>
<td>600276</td>
<td>STAX2</td>
<td>600300</td>
<td>ASNX1</td>
<td>600301</td>
<td>ASNX2</td>
</tr>
<tr>
<td>600302</td>
<td>ASNX3</td>
<td>600320</td>
<td>ATAX1</td>
<td>600321</td>
<td>ATAX2</td>
</tr>
<tr>
<td>600322</td>
<td>ATAX3</td>
<td>600323</td>
<td>ATAX4</td>
<td>600324</td>
<td>ATAX5</td>
</tr>
<tr>
<td>600332</td>
<td>STDVX1</td>
<td>600335</td>
<td>DEVV1</td>
<td>600336</td>
<td>DEVV2</td>
</tr>
<tr>
<td>600337</td>
<td>DEVV3</td>
<td>600350</td>
<td>TERNX1</td>
<td>600351</td>
<td>TERNX2</td>
</tr>
<tr>
<td>600352</td>
<td>ATIX1</td>
<td>600353</td>
<td>ATIX2</td>
<td>600356</td>
<td>TLNXK1</td>
</tr>
<tr>
<td>600357</td>
<td>TLNXKX</td>
<td>600360</td>
<td>TTYX1</td>
<td>600361</td>
<td>RSCNX1</td>
</tr>
<tr>
<td>600362</td>
<td>RSCNX2</td>
<td>600363</td>
<td>CFRKX3</td>
<td>600365</td>
<td>KFRKX1</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>-------</td>
<td>--------------</td>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td>60036</td>
<td>KFRXK2</td>
<td>600370</td>
<td>HFRXK1</td>
<td>600371</td>
<td>GFRXK1</td>
</tr>
<tr>
<td>600373</td>
<td>GETX1</td>
<td>600374</td>
<td>GETX2</td>
<td>600375</td>
<td>TFRXK1</td>
</tr>
<tr>
<td>600376</td>
<td>TFRXK2</td>
<td>600377</td>
<td>SFRXV1</td>
<td>600407</td>
<td>NDUTX1</td>
</tr>
<tr>
<td>600410</td>
<td>NDUTX2</td>
<td>600411</td>
<td>TFRXK3</td>
<td>600414</td>
<td>IFIXX1</td>
</tr>
<tr>
<td>600415</td>
<td>IFIXX2</td>
<td>600416</td>
<td>IFIXX3</td>
<td>600424</td>
<td>GDFBX1</td>
</tr>
<tr>
<td>600425</td>
<td>GDFBX2</td>
<td>600426</td>
<td>GDFBX3</td>
<td>600430</td>
<td>CDFBX1</td>
</tr>
<tr>
<td>600431</td>
<td>CDFBX2</td>
<td>600432</td>
<td>CDFBX3</td>
<td>600439</td>
<td>CDFBX4</td>
</tr>
<tr>
<td>600440</td>
<td>DMPX1</td>
<td>600441</td>
<td>DMPX2</td>
<td>600442</td>
<td>DMPX3</td>
</tr>
<tr>
<td>600449</td>
<td>DUMPX4</td>
<td>600450</td>
<td>RNAXM1</td>
<td>600451</td>
<td>RNAXM2</td>
</tr>
<tr>
<td>600452</td>
<td>RNAXM3</td>
<td>600453</td>
<td>RNAXM4</td>
<td>600454</td>
<td>BKJFX1</td>
</tr>
<tr>
<td>600460</td>
<td>TIMEX1</td>
<td>600461</td>
<td>ZDNEX1</td>
<td>600462</td>
<td>DDTNX1</td>
</tr>
<tr>
<td>600464</td>
<td>DILFX1</td>
<td>600465</td>
<td>TILFX1</td>
<td>600466</td>
<td>DATEX1</td>
</tr>
<tr>
<td>600467</td>
<td>DATEX2</td>
<td>600470</td>
<td>DATEX3</td>
<td>600471</td>
<td>DATEX4</td>
</tr>
<tr>
<td>600472</td>
<td>DATEX5</td>
<td>600473</td>
<td>DATEX6</td>
<td>600475</td>
<td>DATEX7</td>
</tr>
<tr>
<td>600530</td>
<td>SACTX1</td>
<td>600531</td>
<td>SACTX2</td>
<td>600532</td>
<td>SACTX3</td>
</tr>
<tr>
<td>600533</td>
<td>SACTX4</td>
<td>600540</td>
<td>GACTX1</td>
<td>600541</td>
<td>GACTX2</td>
</tr>
<tr>
<td>600544</td>
<td>FFUXF1</td>
<td>600545</td>
<td>FFUXF2</td>
<td>600546</td>
<td>FFUXF3</td>
</tr>
<tr>
<td>600555</td>
<td>D1FX1</td>
<td>600570</td>
<td>SIRX1</td>
<td>600600</td>
<td>SSAXV1</td>
</tr>
<tr>
<td>600601</td>
<td>SSAVX2</td>
<td>600610</td>
<td>SEEX1</td>
<td>600614</td>
<td>WHELX1</td>
</tr>
<tr>
<td>600615</td>
<td>CAPX1</td>
<td>600641</td>
<td>DPEFX2</td>
<td>600620</td>
<td>CRDIX1</td>
</tr>
<tr>
<td>600621</td>
<td>CRDIX2</td>
<td>600622</td>
<td>CRDIX3</td>
<td>600623</td>
<td>CRDIX4</td>
</tr>
<tr>
<td>600624</td>
<td>CRDIX5</td>
<td>600626</td>
<td>CRDIX7</td>
<td>600640</td>
<td>GTDIX1</td>
</tr>
<tr>
<td>600641</td>
<td>GTDIX2</td>
<td>600650</td>
<td>FLINX1</td>
<td>600651</td>
<td>FLINX2</td>
</tr>
<tr>
<td>600652</td>
<td>FFLNX3</td>
<td>600653</td>
<td>FLINX4</td>
<td>600660</td>
<td>FLDTX1</td>
</tr>
<tr>
<td>600661</td>
<td>FLDTX2</td>
<td>600662</td>
<td>FLDTX3</td>
<td>600670</td>
<td>HPTX1</td>
</tr>
<tr>
<td>600704</td>
<td>GTHS1X</td>
<td>600705</td>
<td>GTHS2X</td>
<td>600707</td>
<td>GTHS3X</td>
</tr>
<tr>
<td>600710</td>
<td>ATNX1</td>
<td>600711</td>
<td>ATNX2</td>
<td>600712</td>
<td>ATNX3</td>
</tr>
<tr>
<td>600713</td>
<td>ATNX4</td>
<td>600714</td>
<td>ATNX5</td>
<td>600715</td>
<td>ATNX6</td>
</tr>
<tr>
<td>600716</td>
<td>ATNX7</td>
<td>600717</td>
<td>ATNX8</td>
<td>600720</td>
<td>ATNX9</td>
</tr>
<tr>
<td>600721</td>
<td>ATNX10</td>
<td>600722</td>
<td>ATNX11</td>
<td>600723</td>
<td>ATNX12</td>
</tr>
<tr>
<td>600729</td>
<td>ATNX13</td>
<td>600727</td>
<td>ATNX14</td>
<td>600730</td>
<td>ATNX15</td>
</tr>
<tr>
<td>600731</td>
<td>CVSX1X</td>
<td>600732</td>
<td>SDINX1</td>
<td>600733</td>
<td>SDINX2</td>
</tr>
<tr>
<td>600734</td>
<td>SNAXD3X</td>
<td>600735</td>
<td>SNAXD4</td>
<td>600736</td>
<td>SNAXD5</td>
</tr>
<tr>
<td>600737</td>
<td>NTVWX1</td>
<td>600740</td>
<td>ASNSX1</td>
<td>600741</td>
<td>ASNSX2</td>
</tr>
<tr>
<td>600742</td>
<td>SQX1</td>
<td>600743</td>
<td>SQX2</td>
<td>600746</td>
<td>GTNCX1</td>
</tr>
<tr>
<td>600747</td>
<td>GTNCX2</td>
<td>600750</td>
<td>RNAXM5</td>
<td>600751</td>
<td>RNAXM6</td>
</tr>
<tr>
<td>600752</td>
<td>RNAXM7</td>
<td>600753</td>
<td>RNAXM8</td>
<td>600754</td>
<td>RNAXM9</td>
</tr>
<tr>
<td>600755</td>
<td>RNMX1X</td>
<td>600756</td>
<td>RNMX11</td>
<td>600757</td>
<td>RNMX12</td>
</tr>
<tr>
<td>600760</td>
<td>GUFX3X</td>
<td>600770</td>
<td>ILINS1</td>
<td>600771</td>
<td>ILINS2</td>
</tr>
<tr>
<td>600772</td>
<td>ILINX3</td>
<td>601000</td>
<td>CRLNX1</td>
<td>601001</td>
<td>INLNX1</td>
</tr>
<tr>
<td>601002</td>
<td>LNSTX1</td>
<td>601010</td>
<td>RDTX1</td>
<td>601011</td>
<td>GFKSX1</td>
</tr>
<tr>
<td>601013</td>
<td>GJUX1</td>
<td>601014</td>
<td>GTJUX1</td>
<td>601015</td>
<td>GTJUX3</td>
</tr>
<tr>
<td>601016</td>
<td>IPCFX1</td>
<td>601017</td>
<td>IPCFX2</td>
<td>601020</td>
<td>IPCFX3</td>
</tr>
<tr>
<td>601021</td>
<td>IPCFX4</td>
<td>601022</td>
<td>IPCFX5</td>
<td>601023</td>
<td>IPCFX6</td>
</tr>
<tr>
<td>601024</td>
<td>IPCFX7</td>
<td>601025</td>
<td>IPCFX8</td>
<td>601026</td>
<td>IPCFX9</td>
</tr>
<tr>
<td>601027</td>
<td>IPCX10</td>
<td>601030</td>
<td>IPCX11</td>
<td>601031</td>
<td>IPCX12</td>
</tr>
<tr>
<td>601032</td>
<td>IPCX13</td>
<td>601033</td>
<td>IPCX14</td>
<td>601034</td>
<td>IPCX15</td>
</tr>
<tr>
<td>601035</td>
<td>IPCX16</td>
<td>601036</td>
<td>IPCX17</td>
<td>601037</td>
<td>IPCX18</td>
</tr>
<tr>
<td>601040</td>
<td>IPCX19</td>
<td>601041</td>
<td>IPCX20</td>
<td>601042</td>
<td>IPCX21</td>
</tr>
<tr>
<td>601043</td>
<td>IPCX22</td>
<td>601044</td>
<td>IPCX23</td>
<td>601045</td>
<td>IPCX24</td>
</tr>
<tr>
<td>601046</td>
<td>IPCX25</td>
<td>601047</td>
<td>IPCX26</td>
<td>601050</td>
<td>IPCX27</td>
</tr>
<tr>
<td>601051</td>
<td>IPCX28</td>
<td>601052</td>
<td>IPCX29</td>
<td>601053</td>
<td>IPCX30</td>
</tr>
<tr>
<td>601054</td>
<td>GNBFX1</td>
<td>601055</td>
<td>ENOX1</td>
<td>601056</td>
<td>ENOX2</td>
</tr>
<tr>
<td>601057</td>
<td>ENOX3</td>
<td>601060</td>
<td>ENOX4</td>
<td>601061</td>
<td>ENOX5</td>
</tr>
<tr>
<td>601062</td>
<td>ENOX6</td>
<td>601063</td>
<td>ENOX7</td>
<td>601064</td>
<td>ENOX8</td>
</tr>
<tr>
<td>601065</td>
<td>ENOX9</td>
<td>601066</td>
<td>ENOX10</td>
<td>601067</td>
<td>ENOX11</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601070</td>
<td>ENQX12</td>
<td>601071</td>
<td>ENQX13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601073</td>
<td>ENQX15</td>
<td>601074</td>
<td>ENQX16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601076</td>
<td>ENQX18</td>
<td>601077</td>
<td>ENQX19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601101</td>
<td>ENOX21</td>
<td>601102</td>
<td>IPCF31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601104</td>
<td>PMAPX3</td>
<td>601105</td>
<td>PMAPX4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601107</td>
<td>PMAPX6</td>
<td>601110</td>
<td>SNDPX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601112</td>
<td>SNDPX3</td>
<td>601113</td>
<td>SNDPX4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601115</td>
<td>SNDPX6</td>
<td>601116</td>
<td>SNDPX7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601120</td>
<td>SNDPX9</td>
<td>601121</td>
<td>SNDPX10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601123</td>
<td>SNDP12</td>
<td>601124</td>
<td>SNDP13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601126</td>
<td>SNDP15</td>
<td>601127</td>
<td>SNDP16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601131</td>
<td>SNDP17</td>
<td>601132</td>
<td>OPNX23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601134</td>
<td>CRLNX2</td>
<td>601135</td>
<td>INLNX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601137</td>
<td>ALCX1</td>
<td>601140</td>
<td>ALCX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601142</td>
<td>ALCX4</td>
<td>601143</td>
<td>ALCX5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601145</td>
<td>SPLX2</td>
<td>601146</td>
<td>SPLX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601150</td>
<td>SPLX5</td>
<td>601151</td>
<td>CLSX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601153</td>
<td>ALCX6</td>
<td>601154</td>
<td>CKAX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601156</td>
<td>CKAX3</td>
<td>601157</td>
<td>TIMX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601161</td>
<td>TIMX3</td>
<td>601162</td>
<td>TIMX4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601164</td>
<td>GUFX38</td>
<td>601165</td>
<td>GUFX39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601167</td>
<td>CRDX9</td>
<td>601170</td>
<td>CRDX10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601172</td>
<td>DELDX2</td>
<td>601173</td>
<td>GACTX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601175</td>
<td>DIAGX2</td>
<td>601176</td>
<td>DIAGX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601180</td>
<td>DIAGX5</td>
<td>601201</td>
<td>DIAGX6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601183</td>
<td>DIAGX8</td>
<td>601204</td>
<td>DIAGX9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601206</td>
<td>SYEX1</td>
<td>601207</td>
<td>SYEX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601211</td>
<td>IDX7</td>
<td>601212</td>
<td>IDX8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601214</td>
<td>DUMPX5</td>
<td>601215</td>
<td>DUMPX6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601217</td>
<td>CLSX4</td>
<td>601220</td>
<td>MTDX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601222</td>
<td>MTDX4</td>
<td>601223</td>
<td>MTDX6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601225</td>
<td>GUFX40</td>
<td>601226</td>
<td>MTDX7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601230</td>
<td>LOUTX4</td>
<td>601231</td>
<td>CAPX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601233</td>
<td>SAVX4</td>
<td>601234</td>
<td>TDELX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601236</td>
<td>TADEX2</td>
<td>601237</td>
<td>TLUX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601244</td>
<td>SUBX1</td>
<td>601245</td>
<td>SUBX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601247</td>
<td>TMNX1</td>
<td>601250</td>
<td>SMNX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601252</td>
<td>SUBX5</td>
<td>601253</td>
<td>SUX6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601255</td>
<td>ILNS5</td>
<td>601256</td>
<td>ILNS6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601260</td>
<td>COMNX2</td>
<td>601261</td>
<td>COMNX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601264</td>
<td>PRAX2</td>
<td>601265</td>
<td>COMNX5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601271</td>
<td>CKAX4</td>
<td>601272</td>
<td>GACX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601274</td>
<td>MTDX8</td>
<td>601275</td>
<td>DBRX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601277</td>
<td>GUFX41</td>
<td>601300</td>
<td>GUFX42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601302</td>
<td>TIMEX2</td>
<td>601303</td>
<td>DELFX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601305</td>
<td>DELFX4</td>
<td>601306</td>
<td>DELFX5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601310</td>
<td>DELFX7</td>
<td>601311</td>
<td>DELFX8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601313</td>
<td>DIRX1</td>
<td>601314</td>
<td>DIRX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601316</td>
<td>UPGX1</td>
<td>601317</td>
<td>LNGFX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601321</td>
<td>COMNX8</td>
<td>601322</td>
<td>MTDX9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601324</td>
<td>MTDX11</td>
<td>601325</td>
<td>MTDX12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601327</td>
<td>MTDX14</td>
<td>601330</td>
<td>SAVX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601332</td>
<td>MTDX16</td>
<td>601333</td>
<td>LPINX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601336</td>
<td>LPINX3</td>
<td>601336</td>
<td>MTDX17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601340</td>
<td>DESX9</td>
<td>601341</td>
<td>ACESX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>601344</td>
<td>DSDKX2</td>
<td>601345</td>
<td>MSTRX1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

176
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>601347</td>
<td>MSTRX3</td>
<td>601350</td>
<td>MSTRX4</td>
<td>601351</td>
<td>MSTRX5</td>
</tr>
<tr>
<td>601352</td>
<td>MSTRX6</td>
<td>601353</td>
<td>MSTRX7</td>
<td>601354</td>
<td>MSTRX8</td>
</tr>
<tr>
<td>601355</td>
<td>MSTRX9</td>
<td>601356</td>
<td>MSTX10</td>
<td>601357</td>
<td>MSTX11</td>
</tr>
<tr>
<td>601360</td>
<td>MSTX12</td>
<td>601361</td>
<td>MSTX19</td>
<td>601362</td>
<td>MSTX14</td>
</tr>
<tr>
<td>601363</td>
<td>MSTX15</td>
<td>601364</td>
<td>MSTX16</td>
<td>601365</td>
<td>DSKX01</td>
</tr>
<tr>
<td>601367</td>
<td>DSKX03</td>
<td>601371</td>
<td>GFUSX1</td>
<td>601372</td>
<td>GFUSX2</td>
</tr>
<tr>
<td>601373</td>
<td>SFUSX1</td>
<td>601374</td>
<td>SFUSX2</td>
<td>601376</td>
<td>RCDIX1</td>
</tr>
<tr>
<td>601377</td>
<td>RCDIX2</td>
<td>601400</td>
<td>RCDIX3</td>
<td>601401</td>
<td>RCDIX4</td>
</tr>
<tr>
<td>601402</td>
<td>RUSX1</td>
<td>601403</td>
<td>TDELX2</td>
<td>601404</td>
<td>TIMX5</td>
</tr>
<tr>
<td>601405</td>
<td>LSTRX1</td>
<td>601406</td>
<td>SWFX1</td>
<td>601407</td>
<td>MTOY18</td>
</tr>
<tr>
<td>601410</td>
<td>QPNX26</td>
<td>601411</td>
<td>DELFX9</td>
<td>601412</td>
<td>CRDIX6</td>
</tr>
<tr>
<td>601413</td>
<td>CDMX9</td>
<td>601414</td>
<td>STYPX1</td>
<td>601415</td>
<td>PMAPX7</td>
</tr>
<tr>
<td>601416</td>
<td>DSKDX3</td>
<td>601417</td>
<td>DESX10</td>
<td>601420</td>
<td>DSKDX4</td>
</tr>
<tr>
<td>601421</td>
<td>MSTX17</td>
<td>601422</td>
<td>MSTX18</td>
<td>601423</td>
<td>MSTX19</td>
</tr>
<tr>
<td>601424</td>
<td>MSTX20</td>
<td>601425</td>
<td>MSTX21</td>
<td>601426</td>
<td>MSTX22</td>
</tr>
<tr>
<td>601427</td>
<td>CRDI11</td>
<td>601430</td>
<td>MSTX23</td>
<td>601431</td>
<td>ACEX3</td>
</tr>
<tr>
<td>601432</td>
<td>ACEX4</td>
<td>601433</td>
<td>ACEX5</td>
<td>601435</td>
<td>ACEX6</td>
</tr>
<tr>
<td>601436</td>
<td>STRX01</td>
<td>601437</td>
<td>STRX02</td>
<td>601440</td>
<td>IDIX1</td>
</tr>
<tr>
<td>601441</td>
<td>IDX12</td>
<td>601442</td>
<td>STRX03</td>
<td>601443</td>
<td>STRX04</td>
</tr>
<tr>
<td>601444</td>
<td>PPNX1</td>
<td>601445</td>
<td>PPNX2</td>
<td>601450</td>
<td>SPLX6</td>
</tr>
<tr>
<td>601451</td>
<td>CRDI12</td>
<td>601452</td>
<td>GFUSX3</td>
<td>601453</td>
<td>GFUSX4</td>
</tr>
<tr>
<td>601454</td>
<td>RMNX13</td>
<td>601455</td>
<td>SJBX8</td>
<td>601456</td>
<td>DECR5</td>
</tr>
<tr>
<td>601460</td>
<td>WILDX1</td>
<td>601461</td>
<td>MSTX41</td>
<td>601462</td>
<td>MSTX42</td>
</tr>
<tr>
<td>601475</td>
<td>LCBDBP</td>
<td>601477</td>
<td>LCNOND</td>
<td>601500</td>
<td>SSAVX5</td>
</tr>
<tr>
<td>601502</td>
<td>ATACX6</td>
<td>601503</td>
<td>ATACX7</td>
<td>601533</td>
<td>DSKDX5</td>
</tr>
<tr>
<td>601534</td>
<td>DSKDX6</td>
<td>601535</td>
<td>TIMX6</td>
<td>601536</td>
<td>TIMX7</td>
</tr>
<tr>
<td>601550</td>
<td>SCTX1</td>
<td>601551</td>
<td>SCTX2</td>
<td>601552</td>
<td>SCTX3</td>
</tr>
<tr>
<td>601553</td>
<td>SCTX4</td>
<td>601554</td>
<td>PDUX01</td>
<td>601555</td>
<td>PDUX02</td>
</tr>
<tr>
<td>601556</td>
<td>PDUX03</td>
<td>601557</td>
<td>GETX4</td>
<td>601560</td>
<td>GETX5</td>
</tr>
<tr>
<td>601700</td>
<td>SFUSX4</td>
<td>601701</td>
<td>SFUSX5</td>
<td>601702</td>
<td>SFUSX6</td>
</tr>
<tr>
<td>601703</td>
<td>GETX5</td>
<td>601706</td>
<td>CAPX3</td>
<td>601713</td>
<td>ARGX02</td>
</tr>
<tr>
<td>601715</td>
<td>ARG0104</td>
<td>601716</td>
<td>ARGX05</td>
<td>601717</td>
<td>ARGX06</td>
</tr>
<tr>
<td>601720</td>
<td>ARGX07</td>
<td>601721</td>
<td>ARGX08</td>
<td>601722</td>
<td>ARGX09</td>
</tr>
<tr>
<td>601723</td>
<td>ARGX10</td>
<td>601725</td>
<td>ARGX12</td>
<td>601726</td>
<td>ARGX13</td>
</tr>
<tr>
<td>601727</td>
<td>M0NX01</td>
<td>601730</td>
<td>M0NX02</td>
<td>601731</td>
<td>M0NX03</td>
</tr>
<tr>
<td>601732</td>
<td>M0NX04</td>
<td>601733</td>
<td>ARGX14</td>
<td>601734</td>
<td>ARGX15</td>
</tr>
<tr>
<td>601741</td>
<td>ARGX16</td>
<td>601742</td>
<td>ARGX17</td>
<td>601743</td>
<td>ARGX18</td>
</tr>
<tr>
<td>601744</td>
<td>DEVVX5</td>
<td>601747</td>
<td>STROX6</td>
<td>601750</td>
<td>M0TX24</td>
</tr>
<tr>
<td>601751</td>
<td>MSTX25</td>
<td>601752</td>
<td>MSTX26</td>
<td>601753</td>
<td>LOUXT5</td>
</tr>
<tr>
<td>601754</td>
<td>GJFX43</td>
<td>601755</td>
<td>MTDX19</td>
<td>601756</td>
<td>M0TX20</td>
</tr>
<tr>
<td>601757</td>
<td>MSTX27</td>
<td>601760</td>
<td>M0TX28</td>
<td>601761</td>
<td>MSTX29</td>
</tr>
<tr>
<td>601763</td>
<td>DSKX05</td>
<td>601764</td>
<td>DSKX06</td>
<td>601765</td>
<td>DSKX07</td>
</tr>
<tr>
<td>601766</td>
<td>DSKX08</td>
<td>601767</td>
<td>COMX10</td>
<td>601770</td>
<td>COMX30</td>
</tr>
<tr>
<td>601771</td>
<td>L0CKX1</td>
<td>601772</td>
<td>L0CKX2</td>
<td>601774</td>
<td>ILLX01</td>
</tr>
<tr>
<td>601775</td>
<td>ILLX02</td>
<td>601776</td>
<td>ILLX03</td>
<td>601777</td>
<td>ILLX04</td>
</tr>
<tr>
<td>602000</td>
<td>MSTX31</td>
<td>602001</td>
<td>MSTX32</td>
<td>602002</td>
<td>MSTX33</td>
</tr>
<tr>
<td>602003</td>
<td>S0TX1</td>
<td>602004</td>
<td>S0TX17</td>
<td>602005</td>
<td>S0TX18</td>
</tr>
<tr>
<td>602006</td>
<td>PMLX12</td>
<td>602007</td>
<td>PMLX13</td>
<td>602010</td>
<td>DLFX10</td>
</tr>
<tr>
<td>602011</td>
<td>DLFX11</td>
<td>602012</td>
<td>QJFX44</td>
<td>602013</td>
<td>UTSTX1</td>
</tr>
<tr>
<td>602014</td>
<td>UTSTX2</td>
<td>602015</td>
<td>UTSTX3</td>
<td>602016</td>
<td>BTX01</td>
</tr>
<tr>
<td>602017</td>
<td>BTX02</td>
<td>602020</td>
<td>DCNX1</td>
<td>602021</td>
<td>DCNX5</td>
</tr>
<tr>
<td>602022</td>
<td>DCNX3</td>
<td>602023</td>
<td>DCNX4</td>
<td>602024</td>
<td>DCNX9</td>
</tr>
<tr>
<td>602025</td>
<td>DCNX8</td>
<td>602026</td>
<td>DCNX11</td>
<td>602027</td>
<td>DCNX12</td>
</tr>
<tr>
<td>602030</td>
<td>TT0X01</td>
<td>602031</td>
<td>BT0X03</td>
<td>602032</td>
<td>MONX01</td>
</tr>
<tr>
<td>602033</td>
<td>ARGX19</td>
<td>602035</td>
<td>COMX11</td>
<td>602036</td>
<td>COMX12</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>--------</td>
<td>---------------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>602037</td>
<td>COMX13</td>
<td>602040</td>
<td>COMX14</td>
<td>602041</td>
<td>COMX15</td>
</tr>
<tr>
<td>602042</td>
<td>COMX16</td>
<td>602043</td>
<td>COMX17</td>
<td>602044</td>
<td>NPXAMB</td>
</tr>
<tr>
<td>602045</td>
<td>NPXNDSW</td>
<td>602046</td>
<td>NPXNDSM</td>
<td>602047</td>
<td>NPXNUL</td>
</tr>
<tr>
<td>602050</td>
<td>NPX13W</td>
<td>602051</td>
<td>NPX13X</td>
<td>602052</td>
<td>NPX13N</td>
</tr>
<tr>
<td>602053</td>
<td>NPX13DT</td>
<td>602054</td>
<td>NPX13Q</td>
<td>602055</td>
<td>NPX13MT</td>
</tr>
<tr>
<td>602056</td>
<td>NPX13MD</td>
<td>602057</td>
<td>NPX13MA</td>
<td>602060</td>
<td>GJFx45</td>
</tr>
<tr>
<td>602061</td>
<td>GJFx46</td>
<td>602062</td>
<td>GJFx47</td>
<td>602063</td>
<td>MSTR34</td>
</tr>
<tr>
<td>602064</td>
<td>GJFx48</td>
<td>602065</td>
<td>GJFx49</td>
<td>602077</td>
<td>SUBX7</td>
</tr>
<tr>
<td>602100</td>
<td>DELT10</td>
<td>602101</td>
<td>CRDI13</td>
<td>602102</td>
<td>CRDI14</td>
</tr>
<tr>
<td>602103</td>
<td>CRDI15</td>
<td>602104</td>
<td>CRDI16</td>
<td>602105</td>
<td>ENACX1</td>
</tr>
<tr>
<td>602106</td>
<td>ENACX2</td>
<td>602107</td>
<td>ENACX3</td>
<td>602110</td>
<td>ENACX4</td>
</tr>
<tr>
<td>602111</td>
<td>VACX20</td>
<td>602112</td>
<td>VACX21</td>
<td>602119</td>
<td>USGX01</td>
</tr>
<tr>
<td>602114</td>
<td>BDTX04</td>
<td>602116</td>
<td>USGX02</td>
<td>602117</td>
<td>CRDI17</td>
</tr>
<tr>
<td>602120</td>
<td>ENSX23</td>
<td>602121</td>
<td>ENSX22</td>
<td>602122</td>
<td>DCNX2</td>
</tr>
<tr>
<td>602123</td>
<td>ARSXX1</td>
<td>602124</td>
<td>USGX03</td>
<td>602125</td>
<td>IPCF35</td>
</tr>
<tr>
<td>602126</td>
<td>VACX22</td>
<td>602127</td>
<td>CRDI18</td>
<td>602130</td>
<td>CRDI19</td>
</tr>
<tr>
<td>602132</td>
<td>BDTX05</td>
<td>602133</td>
<td>CRDI20</td>
<td>602134</td>
<td>COMX18</td>
</tr>
<tr>
<td>602135</td>
<td>COMX19</td>
<td>602136</td>
<td>CRDI21</td>
<td>602137</td>
<td>ACESX7</td>
</tr>
<tr>
<td>602140</td>
<td>CRDI22</td>
<td>602141</td>
<td>CRDI23</td>
<td>602142</td>
<td>STRX07</td>
</tr>
<tr>
<td>602143</td>
<td>STRX08</td>
<td>602144</td>
<td>CRDI24</td>
<td>602165</td>
<td>PMCLX4</td>
</tr>
<tr>
<td>602170</td>
<td>FRXH2X8</td>
<td>602171</td>
<td>ARGX20</td>
<td>602172</td>
<td>ARGX21</td>
</tr>
<tr>
<td>602173</td>
<td>ARGX22</td>
<td>602177</td>
<td>ARGX22</td>
<td>602200</td>
<td>ARGX24</td>
</tr>
<tr>
<td>602201</td>
<td>MSTR35</td>
<td>602202</td>
<td>DCNX19</td>
<td>602203</td>
<td>DCNX14</td>
</tr>
<tr>
<td>602204</td>
<td>DCNX15</td>
<td>602205</td>
<td>GJFx50</td>
<td>602206</td>
<td>KDRX01</td>
</tr>
<tr>
<td>602207</td>
<td>NDXD10</td>
<td>602210</td>
<td>NDXD03</td>
<td>602211</td>
<td>GJFx51</td>
</tr>
<tr>
<td>602212</td>
<td>COMX20</td>
<td>602220</td>
<td>GDRX01</td>
<td>602221</td>
<td>GDRX02</td>
</tr>
<tr>
<td>602222</td>
<td>STRX09</td>
<td>602223</td>
<td>MSTR36</td>
<td>602224</td>
<td>MSTR37</td>
</tr>
<tr>
<td>602225</td>
<td>MSTR40</td>
<td>602230</td>
<td>MSTR17</td>
<td>602234</td>
<td>MSTR18</td>
</tr>
<tr>
<td>602231</td>
<td>MSTR11</td>
<td>602232</td>
<td>MSTR12</td>
<td>602236</td>
<td>MSTR13</td>
</tr>
<tr>
<td>602235</td>
<td>MSTR15</td>
<td>602236</td>
<td>MSTR16</td>
<td>602238</td>
<td>MSTR17</td>
</tr>
<tr>
<td>602240</td>
<td>MSTR17</td>
<td>602241</td>
<td>MSTR18</td>
<td>602242</td>
<td>MSTR19</td>
</tr>
<tr>
<td>602243</td>
<td>MSTR19</td>
<td>602245</td>
<td>MSTR20</td>
<td>602246</td>
<td>MSTR21</td>
</tr>
<tr>
<td>602247</td>
<td>MSTR21</td>
<td>602257</td>
<td>MSTR22</td>
<td>602258</td>
<td>MSTR23</td>
</tr>
<tr>
<td>602312</td>
<td>ARFX02</td>
<td>602313</td>
<td>ARFX03</td>
<td>602314</td>
<td>ARFX04</td>
</tr>
<tr>
<td>602315</td>
<td>ARFX05</td>
<td>602316</td>
<td>ARFX06</td>
<td>602317</td>
<td>ARFX07</td>
</tr>
<tr>
<td>602320</td>
<td>ARFX08</td>
<td>602321</td>
<td>ARFX09</td>
<td>602322</td>
<td>ARFX10</td>
</tr>
<tr>
<td>602323</td>
<td>ARFX11</td>
<td>602324</td>
<td>ARFX12</td>
<td>602325</td>
<td>ARFX13</td>
</tr>
<tr>
<td>602326</td>
<td>ARFX15</td>
<td>602327</td>
<td>ARFX16</td>
<td>602330</td>
<td>ARFX17</td>
</tr>
<tr>
<td>602331</td>
<td>ARFX18</td>
<td>602332</td>
<td>ARFX19</td>
<td>602333</td>
<td>ARFX20</td>
</tr>
<tr>
<td>602334</td>
<td>ARFX21</td>
<td>602335</td>
<td>ARFX22</td>
<td>602336</td>
<td>ARFX23</td>
</tr>
<tr>
<td>602337</td>
<td>ARFX24</td>
<td>602340</td>
<td>ARFX25</td>
<td>602341</td>
<td>ARFX26</td>
</tr>
<tr>
<td>602342</td>
<td>ARFX27</td>
<td>602343</td>
<td>ARFX28</td>
<td>602344</td>
<td>ARFX29</td>
</tr>
<tr>
<td>602352</td>
<td>ARFX30</td>
<td>602353</td>
<td>ARFX31</td>
<td>602354</td>
<td>ARFX32</td>
</tr>
<tr>
<td>602355</td>
<td>ARFX33</td>
<td>602356</td>
<td>ARFX34</td>
<td>602357</td>
<td>ARFX35</td>
</tr>
<tr>
<td>602360</td>
<td>ARFX36</td>
<td>602361</td>
<td>ARFX37</td>
<td>602362</td>
<td>ARFX38</td>
</tr>
<tr>
<td>602363</td>
<td>ARFX39</td>
<td>602364</td>
<td>ARFX40</td>
<td>602365</td>
<td>ARFX41</td>
</tr>
<tr>
<td>602366</td>
<td>ARFX42</td>
<td>602367</td>
<td>ARFX43</td>
<td>602370</td>
<td>ARFX44</td>
</tr>
<tr>
<td>602371</td>
<td>ARFX45</td>
<td>602372</td>
<td>ARFX46</td>
<td>602373</td>
<td>ARFX47</td>
</tr>
<tr>
<td>602374</td>
<td>ARFX48</td>
<td>602375</td>
<td>ARFX49</td>
<td>602376</td>
<td>ARFX50</td>
</tr>
<tr>
<td>602377</td>
<td>ARFX51</td>
<td>602400</td>
<td>ARFX52</td>
<td>602401</td>
<td>ARFX53</td>
</tr>
<tr>
<td>602406</td>
<td>ARFX54</td>
<td>602407</td>
<td>ARFX55</td>
<td>602408</td>
<td>ARFX56</td>
</tr>
<tr>
<td>602411</td>
<td>ARFX57</td>
<td>602412</td>
<td>ARFX58</td>
<td>602413</td>
<td>ARFX59</td>
</tr>
<tr>
<td>602414</td>
<td>ARFX60</td>
<td>602415</td>
<td>ARFX61</td>
<td>602416</td>
<td>ARFX62</td>
</tr>
<tr>
<td>602417</td>
<td>ARFX63</td>
<td>602420</td>
<td>ARFX64</td>
<td>602421</td>
<td>ARFX65</td>
</tr>
<tr>
<td>602422</td>
<td>ARFX66</td>
<td>602423</td>
<td>ARFX67</td>
<td>602424</td>
<td>ARFX68</td>
</tr>
<tr>
<td>602425</td>
<td>ARFX69</td>
<td>602426</td>
<td>ARFX70</td>
<td>602427</td>
<td>ARFX71</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>-------</td>
<td>--------------</td>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td>602431</td>
<td>SMAPX1</td>
<td>602492</td>
<td>TTMSX1</td>
<td>602433</td>
<td>MONX06</td>
</tr>
<tr>
<td>602434</td>
<td>BDTX06</td>
<td>602435</td>
<td>BDTX07</td>
<td>602436</td>
<td>BDTX08</td>
</tr>
<tr>
<td>602437</td>
<td>BDTX09</td>
<td>602440</td>
<td>BDTX10</td>
<td>602441</td>
<td>BDTX11</td>
</tr>
<tr>
<td>602442</td>
<td>BDTX12</td>
<td>602443</td>
<td>BDTX13</td>
<td>602444</td>
<td>BDTX14</td>
</tr>
<tr>
<td>602445</td>
<td>BDTX15</td>
<td>602446</td>
<td>BDTX16</td>
<td>602447</td>
<td>BDTX17</td>
</tr>
<tr>
<td>602450</td>
<td>BDTX18</td>
<td>602451</td>
<td>NTMX1</td>
<td>602452</td>
<td>CMX21</td>
</tr>
<tr>
<td>602453</td>
<td>DELX13</td>
<td>602454</td>
<td>ANTX01</td>
<td>602455</td>
<td>TTYX02</td>
</tr>
<tr>
<td>602456</td>
<td>NSPX24</td>
<td>602457</td>
<td>NSPX25</td>
<td>602460</td>
<td>NSPX26</td>
</tr>
<tr>
<td>602461</td>
<td>GJFX53</td>
<td>602462</td>
<td>IDX34</td>
<td>602463</td>
<td>IDX35</td>
</tr>
<tr>
<td>602464</td>
<td>PMAPX8</td>
<td>602465</td>
<td>SMAPX2</td>
<td>602467</td>
<td>BDTX19</td>
</tr>
<tr>
<td>602470</td>
<td>BDTX20</td>
<td>602471</td>
<td>ILLX05</td>
<td>602472</td>
<td>XSEVX1</td>
</tr>
<tr>
<td>602473</td>
<td>XSEVX2</td>
<td>602474</td>
<td>XSEVX3</td>
<td>602475</td>
<td>ABRKX2</td>
</tr>
<tr>
<td>602477</td>
<td>ABRKX4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

179
## TOPS-20 JSYS ERROR MNEMONICS

**Note**

See TOPS-20 JSYS ERROR CODES for a list of error codes sorted numerically.

JSYS names ([JSYS]) are listed for those error mnemonics that are called from within a particular JSYS module. Error mnemonics not followed by [JSYS] are not called from within any particular JSYS module, but may be returned while a JSYS is executing.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Code</th>
<th>Text String [JSYS]</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABRKX1:</td>
<td>602123</td>
<td>Address break not available on this system [ABRKR]</td>
</tr>
<tr>
<td>ACESX1:</td>
<td>601341</td>
<td>Argument block too small [ACCES]</td>
</tr>
<tr>
<td>ACESX3:</td>
<td>601431</td>
<td>Password is required [ACCES; CRDIR]</td>
</tr>
<tr>
<td>ACESX4:</td>
<td>601432</td>
<td>Function not allowed for another job [ACCES]</td>
</tr>
<tr>
<td>ACESX5:</td>
<td>601433</td>
<td>No function specified for ACCES [ACCES]</td>
</tr>
<tr>
<td>ACESX6:</td>
<td>601435</td>
<td>Directory is not accessed [ACCES]</td>
</tr>
<tr>
<td>ACESX7:</td>
<td>602137</td>
<td>Directory is &quot;files-only&quot; and cannot be accessed [ACCES]</td>
</tr>
<tr>
<td>ALC1X:</td>
<td>601137</td>
<td>Invalid function [ALLOC]</td>
</tr>
<tr>
<td>ALC2X:</td>
<td>601140</td>
<td>WHEEL or OPERATOR capability required [ALLOC]</td>
</tr>
<tr>
<td>ALC3X:</td>
<td>601141</td>
<td>Device is not assignable [ALLOC]</td>
</tr>
<tr>
<td>ALC4X:</td>
<td>601142</td>
<td>Invalid job number [ALLOC]</td>
</tr>
<tr>
<td>ALC5X:</td>
<td>601143</td>
<td>Device already assigned to another job [ALLOC]</td>
</tr>
<tr>
<td>ALC6X:</td>
<td>601153</td>
<td>Device assigned to user job, but will be given to allocator when released [ALLOC]</td>
</tr>
<tr>
<td>ANTX01:</td>
<td>602454</td>
<td>No more network terminals available [WTDPR]</td>
</tr>
<tr>
<td>ARCFX2:</td>
<td>602312</td>
<td>File already has archive status [ARCF]</td>
</tr>
<tr>
<td>ARCFX3:</td>
<td>602313</td>
<td>Cannot perform ARCF functions on nonmultiple directory devices [ARCF]</td>
</tr>
<tr>
<td>ARCFX4:</td>
<td>602314</td>
<td>File is not on line [ARCF]</td>
</tr>
<tr>
<td>ARCFX5:</td>
<td>602315</td>
<td>Files are not on the same device or structure [ARCF]</td>
</tr>
<tr>
<td>ARCFX6:</td>
<td>602316</td>
<td>File does not have archive status [ARCF]</td>
</tr>
<tr>
<td>ARCFX7:</td>
<td>602317</td>
<td>Invalid parameter for .ARSTT [ARCF]</td>
</tr>
<tr>
<td>ARCFX8:</td>
<td>602320</td>
<td>Archive not complete [ARCF]</td>
</tr>
<tr>
<td>ARCFX9:</td>
<td>602321</td>
<td>File not off line [ARCF]</td>
</tr>
<tr>
<td>ARCX10:</td>
<td>602322</td>
<td>Archive prohibited [ARCF]</td>
</tr>
<tr>
<td>ARCX11:</td>
<td>602323</td>
<td>Archive requested, modification prohibited</td>
</tr>
</tbody>
</table>
ARCX12: 602324 Archive requested, delete prohibited
ARCX13: 602325 Archive system request not completed
ARCX14: 602322 Restore failed [ARCF]
ARCX15: 602333 Migration prohibited [ARCF]
ARCX16: 602334 Cannot exempt off-line, archived, or archive-pending files [ARCF]
ARCX17: 602335 FDB improper format for ARCF [ARCF]
ARCX18: 602336 Retrieval wait cannot be fulfilled for waiting process [ARCF]
ARCX19: 602337 Migration already pending [ARCF]
ARGX02: 601713 Invalid function [ADBRK; BODT; DSKAS; GHST%; GTNCP%; NODE; PMCTL; SKED%; USAGE; WILD%; ARCF; METER%]
ARGX04: 601715 Argument block too small [GETOK%; MTU%; SKED%; USAGE; XSIR%]
ARGX05: 601716 Argument block too long [GETOK%; MTU%; USAGE; XSIR%]
ARGX06: 601717 Invalid page number [PMAP; PMCTL; RPACS]
ARGX07: 601720 Invalid job number [ACCES]
ARGX08: 601721 No such job [ACCES; SKED%]
ARGX09: 601722 Invalid byte size [CRNLN; NTMAN%]
ARGX10: 601723 Invalid access requested [TTMSG]
ARGX12: 601725 Invalid process handle [PLOCK]
ARGX13: 601726 Invalid software interrupt channel number [NODE]
ARGX14: 601733 Invalid account identifier
ARGX15: 601734 Job is not logged in [SKED%]
ARGX16: 601741 Password is required
ARGX17: 601742 Invalid argument block length [NTMAN%; TEXTI; XRMAP%]
ARGX18: 601743 Invalid structure name [MSTR]
ARGX19: 602033 Invalid unit number [NODE]
ARGX20: 602171 Invalid arithmetic trap argument [SWTRP%]
ARGX21: 602172 Invalid IUUD trap argument [SWTRP%]
ARGX22: 602173 Invalid flags [PLOCK; WILD%]
ARGX23: 602177 Invalid section number [RSMAP%; SKPIR]
ARGX24: 602200 Invalid count [PLOCK; SKPIR]
ARGX25: 602246 Invalid class [SKED%]
ARGX26: 602340 File is off line [DELDF; GETOK%]
ARGX27: 602341 Offline expiration time cannot exceed system maximum [SFTAD]
ARGX28: 602412 Not available on this system [RSMAP%]
ARGX29: 602414 Invalid class share [SKED%]
ARGX30: 602415 Invalid KNOB value [SKED%]
ARGX31: 602416 Class scheduler already enabled [SKED%]
ASNDX1: 600300 Device is not assignable [ASND]
ASNDX2: 600301 Illegal to assign this device [ASND]
ASNDX3: 600302 No such device [ASND]
ASNSX1: 600740 Insufficient system resources (All special queues in use) [ASNSQ]
ASNSX2: 600741 Link(s) assigned to another special queue [ASNSQ]
ATACX1: 600320 Invalid job number [ATACH; TWAKE]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

ATACX2: 600321 Job already attached [ATACH]
ATACX3: 600322 Incorrect user number [ATACH]
ATACX4: 600323 Invalid password [ATACH]
ATACX5: 600324 This job has no controlling terminal [ATACH]
ATACX6: 601502 Terminal is already attached to a job [ATACH]
ATACX7: 601503 Illegal terminal number [ATACH]
ATIX1: 600352 Invalid software interrupt channel number [ATI]
ATIX2: 600353 Control-C capability required [ATI]
ATNX10: 600721 Send JFN is not a network connection [ATNVT]
ATNX11: 600722 Send JFN has been used [ATNVT]
ATNX12: 600723 Send connection has been refused [ATNVT]
ATNX13: 600724 Insufficient system resources (no NVTs) [ATNVT]
ATNX1: 600710 Invalid receive JFN [ATNVT]
ATNX2: 600711 Receive JFN is not open for read [ATNVT]
ATNX3: 600712 Receive JFN is not open [ATNVT]
ATNX4: 600713 Receive JFN is not a network connection [ATNVT]
ATNX5: 600714 Receive JFN has been used [ATNVT]
ATNX6: 600715 Receive connection has been refused [ATNVT]
ATNX7: 600716 Invalid send JFN [ATNVT]
ATNX8: 600717 Send JFN is not open for write [ATNVT]
ATNX9: 600720 Send JFN is not open [ATNVT]
BKJFEN: 600454 Illegal to back up terminal pointer twice [BKJFN]
BDTX01: 602016 Invalid DTE-20 number [BOOT]
BDTX02: 602017 Invalid byte size [BOOT]
BDTX03: 602031 Invalid protocol version number [BOOT]
BDTX04: 602114 Byte count is not positive [BOOT]
BDTX05: 602132 Protocol initialization failed [BOOT]
BDTX06: 602434 GTJFN failed for dump file [BOOT]
BDTX07: 602435 OPENF failed for dump file [BOOT]
BDTX08: 602436 Dump failed [BOOT]
BDTX09: 602437 To -10 error on dump [BOOT]
BDTX10: 602440 To -11 error on dump [BOOT]
BDTX11: 602441 Failed to assign page on dump [BOOT]
BDTX12: 602442 Reload failed [BOOT]
BDTX13: 602443 -11 didn’t power down [BOOT]
BDTX14: 602444 -11 didn’t power up [BOOT]
BDTX15: 602445 ROM did not ACK the -10 [BOOT]
BDTX16: 602446 -11 boot program did not make it to -11 [BOOT]
BDTX17: 602447 -11 took more than 1 minute to reload; will cause retry [BOOT]
BDTX18: 602450 Unknown BOOT error [BOOT]
BDTX19: 602467 Overdue TO-11 transfer aborted [BOOT]
BDTX20: 602470 Overdue TO-10 transfer aborted [BOOT]
CACTX1: 600045 Invalid account identifier [CACCT]
CACTX2: 600046 Job is not logged in [CACCT; MSTR]
CAPX1: 600615 WHEEL or OPERATOR capability required [ARCF; ACCES; BOOT; GIVOK%; MSFRK; MTUX;
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 USYS Error Mnemonics

NTMAN%; PEEK; RCVOK%; SFTAD; SFUST; SKED%; SYERR; USAGE |
CAPX2: 601231 WHEEL, OPERATOR, or MAINTENANCE capability required [HSYS; MSTR; NODE; PMCTL; USRIO]
CAPX3: 601706 WHEEL capability required [UTEST]
CFDBX1: 600430 Invalid displacement [CHFDB]
CFDBX2: 600431 Illegal to change specified bits [CHFDB; SFTAD]
CFDBX3: 600432 Write or owner access required [CHFDB]
CFDBX4: 600433 Invalid value for specified bits [CHFDB]
CFRX3: 600363 Insufficient system resources [CFORK; PMAP]
CKAX1: 601154 Argument block too small [CHKAC]
CKAX2: 601155 Invalid directory number [CHKAC]
CKAX3: 601156 Invalid access code [CHKAC]
CKAX4: 601271 File is not on disk [CHKAC]
CLX1: 600160 File is not open [CLOSF]
CLX2: 600161 File cannot be closed by this process [CLOSF]
CLX3: 601151 File still mapped [CLOSF]
CLX4: 601217 Device still active [CLOSF]
CNIX1: 600200 Invalid password [ACCES]
CNIX5: 600204 Job is not logged in [ACCES]
CNIX7: 602004 The CNDIR USYS has been replaced by ACCES
COMNX1: 601257 Invalid CMDN function code [COMND]
COMNX2: 601260 Field too long for internal buffer [COMND]
COMNX3: 601261 Command too long for internal buffer [COMND]
COMNX5: 601265 Invalid string pointer argument [COMND]
COMNX8: 601321 Number base out of range 2-10 [COMND]
COMNX9: 601413 End of input file reached [COMND]
COMX10: 601767 Invalid default string [COMND]
COMX11: 602039 Invalid CMRTY pointer [COMND]
COMX12: 602036 Invalid CMBFP pointer [COMND]
COMX13: 602037 Invalid CMPTR pointer [COMND]
COMX14: 602040 Invalid CMABP pointer [COMND]
COMX15: 602041 Invalid default string pointer [COMND]
COMX16: 602042 Invalid help message pointer [COMND]
COMX17: 602043 Invalid byte pointer in function block [COMND]
COMX18: 602134 Invalid character in node name [COMND]
COMX19: 602135 Too many characters in node name [COMND]
COMX20: 602212 Invalid node name [COMND]
COMX21: 602452 Node name doesn't contain an alphabetic character [COMND]
CRD110: 601170 Maximum directory number exceeded; index table needs expanding [CRDIR]
CRD111: 601427 Invalid terminating bracket on directory [CRDIR]
CRD112: 601451 Structure is not mounted [CRDIR]
CRD113: 602101 Request exceeds superior directory working quota [CRDIR]
CRD114: 602102 Request exceeds superior directory permanent quota [CRDIR]
CRD115: 602103 Request exceeds superior directory subdirectory quota [CRDIR]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

CRDI16: 602104 Invalid user group [CRDIR]
CRDI17: 602117 Illegal to create nonfiles-only subdirectory under files-only directory [CRDIR]
CRDI18: 602127 Illegal to delete logged-in directory [CRDIR]
CRDI19: 602130 Illegal to delete connected directory [CRDIR]
CRDI20: 602133 WHEEL, OPERATOR, or requested capability required [CRDIR]
CRDI21: 602136 Working space insufficient for current allocation [CRDIR]
CRDI22: 602140 Subdirectory quota insufficient for existing subdirectories [CRDIR]
CRDI23: 602141 Superior directory does not exist [CRDIR]
CRDI24: 602144 Invalid subdirectory quota [CRDIR]
CRDIX1: 600620 WHEEL or OPERATOR capability required [CRDIR]
CRDIX2: 600621 Illegal to change number of old directory [CRDIR]
CRDIX3: 600622 Insufficient system resources (Job Storage Block full) [CRDIR]
CRDIX4: 600623 Superior directory full [CRDIR]
CRDIX5: 600624 Directory name not given [CRDIR]
CRDIX6: 601412 Directory file is mapped [CRDIR]
CRDIX7: 600626 File(s) open in directory [CRDIR]
CRDIX8: 601166 Invalid directory number [CRDIR]
CRDIX9: 601167 Internal format of directory is incorrect [CRDIR]
CRJUX1: 600020 Invalid parameter or function bit combination [CRJOB]
CRJUX2: 600021 Illegal for created job to enter MINI-EXEC [CRJOB]
CRJUX4: 600023 Terminal is not available [CRJOB]
CRJUX5: 600024 Unknown name for LOGIN [CRJOB]
CRJUX6: 600025 Insufficient system resources [CRJOB]
CRNX1: 601000 Logical name is not defined [CRLNM]
CRNX2: 601134 WHEEL or OPERATOR capability required [CRLNM]
CRLNX3: 601152 Invalid function [CRLNM]
CVHST1: 600727 No string for that host number [CVHST; CVSKT]
CVSKX1: 600730 Invalid JFN [CVSKT]
CVSKX2: 600731 Local socket invalid in this context [CVSKT]
DATEX1: 600466 Year out of range [IDCNV; IDTIM; ODTNC]
DATEX2: 600467 Month is not less than 12 [IDCNV; ODTNC]
DATEX3: 600470 Day of month too large [IDCNV; IDTIM; ODTNC]
DATEX4: 600471 Day of week is not less than 7 [ODTNC]
DATEX5: 600472 Date out of range [IDCNV; IDTIM]
DATEX6: 600473 System date and time are not set [DDCNV; IDTIM; SFTAD]
DATEX7: 602310 Julian day is out of range [IDCNV]
DBRXX1: 601275 No interrupts in progress [DBRXK]
DCNX1: 602020 Invalid network file name
| DCNX2: | 602122 | Interrupt message must be read first |
| DCNX3: | 602022 | Invalid object |
| DCNX4: | 602023 | Invalid task name |
| DCNX5: | 602021 | No more logical links available |
| DCNX8: | 602025 | Invalid network operation |
| DCNX9: | 602024 | Object is already defined |
| DCNX11: | 602026 | Link aborted |
| DCNX12: | 602027 | String exceeds 16 bytes |
| DCNX19: | 602202 | Node not accessible |
| DCNX14: | 602023 | Previous interrupt message outstanding |
| DCNX15: | 602204 | No interrupt message available |
| DEC85: | 601456 | DEC-reserved bits not zero [DSKOP; RFSTS] |
| DELDX1: | 601171 | WHEEL or OPERATOR capability required [DELF] |
| DELDX2: | 601172 | Invalid directory number [DELF] |
| DELF10: | 602100 | Directory still contains subdirectory [DELF] |
| DELF11: | 600170 | Delete access required [DELF; DELNF] |
| DELF12: | 601303 | File cannot be expunged because it is currently open [DELF; DELF] |
| DELF13: | 601304 | System scratch area depleted; file not deleted [DELF] |
| DELF14: | 601305 | Directory symbol table could not be rebuilt [DELF; DELF] |
| DELF15: | 601306 | Directory symbol table needs rebuilding [DELF; DELF] |
| DELF16: | 601307 | Internal format of directory is incorrect [DELF; DELF; DIRST; GFUST; GTDAL; PPNST; VACCT] |
| DELF17: | 601310 | FDB formatted incorrectly; file not deleted [DELF; DELF] |
| DELF18: | 601311 | FDB not found; file not deleted [DELF; DELF] |
| DELF19: | 601411 | File is not a directory file [DELF] |
| DELF20: | 602330 | File has archive status, delete is not permitted |
| DELF21: | 602331 | File has no pointer to offline storage [DELF] |
| DELF22: | 602453 | File is marked "Never Delete" [DELF; DELF] |
| DESX1: | 600150 | Invalid source/destination designator [BIN; BKJFN; BOUT; CFIBF; CFDHF; CHFDB; CLOSF; DELF; DELNF; DEVST; DFIN; DFOUT; DIBE; DIRST; DOBE; DUMP; DUMP; DVCHR; ERSTR; FFFF; FFUF; FLIN; FLOUT; GACTF; GDSTG; GFUST; GNJFN; GTFDB; GTYP; JFNS; MTDFP; MTUF; NIN; NOUT; OPENF; PBIN; PSOUT; PMAP; PPNST; PSOUT; RCDIR; RFBFS; RFPQS; RFPTR; RFTAD; RIN; RLJFN; RNNM; RQUT; RPACS; SACTF; SCCTY; SDSTG; SFBSZ; SFCOC; SFMOD; SPFTR; STAD; SFUST; SIBE; SIN; SINT; SIZEF; SOBE; SOUT; SOUTR; SPACS; STPAR; STPPN; STSTG; STYP; SWJFN; TLINK] |
| DESX2: | 600151 | Terminal is not available to this job [RCDIR; BIN; BKJFN; BOUT; CLOSF; DEVST; |
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

DFIN; DFOUT; DIRST; DUMP; DUMP0; FLIN;
FLOUT; GACTF; GDSTS; GFUST; GNJFN; JFN;
MTOP; NIN; NOUT; PBIN; PBOUT; PNNST;
PSOUT; RFBSZ; RFPT; RIN; ROUT; SDSTS;
SFBSZ; SFCOC; SFPTR; SFUST; SIN; SIZEF;
SOUT; SPACS; STI; STO; STPPN; STTST;
STTYP; SWJFN]

DESX3:  600152
JFN is not assigned [BIN; BKJFN; BOUT;
CFIB; CFDB; CHFDB; CLOSF; DELF; DELNF;
DEVST; DFIN; DFOUT; DIBE; DIRST; DOBE;
DUMP; DUMP0; DVCHR; FFFF; FFUP; FLIN;
FLOUT; GACTF; GDSTS; GFUST; GNJFN; GTFDB;
JFN; MTOP; NIN; NOUT; OPENF; PBOUT;
PMA; PNNST; PSOUT; RCDDIR; RFPOS;
RFPT; RFTAD; RIN; RLJFN; RNAMF; ROUT;
RPACS; SACTF; SDSTS; SFBSZ; SFCOC; SFMOD;
SFPT; SFTAD; SFUST; SIBE; SIN; SINR;
SIZEF; SOBE; SOUT; SOUR; SPACS; SPJFN;
STPAR; STPPN; STTST; SWJFN; UFPGS; WILD]

DESX4:  600153
Invalid use of terminal designator or
string pointer [CHFDB; CLOSF; DELF; DELNF;
DUMP; DUMP0; DVCHR; FFFF; FFUP; GACTF;
GDSTS; GFUST; GNJFN; GTFDB; JFN; MTOP;
OPENF; RCDDIR; RFBSZ; RFPT; RIN; RLJFN;
RNAMF; ROUT; RPACS; SACTF; SDSTS; SFBSZ;
SFPT; SFTAD; SIZEF; SPACS; STPPN; STTST;
SWJFN; UFPGS]

DESX5:  600154
File is not open [BIN; BKJFN; BOUT; CFIB;
CFDB; DEQ; DFIN; DFOUT; DIBE; DIRST;
DOBE; DUMP; DUMP0; ENQ; FFFF; FLIN;
FLOUT; GDSTS; GFUST; GNJFN; GTFDB; JFN;
MTOP; NIN; NOUT; PBIN; PBOUT; PMA;
PNNST; PSOUT; RCDDIR; RFPOS;
RFPT; RFTAD; RIN; ROUT; RPACS; SDSTS;
SFBSZ; SFCOC; SFMOD; SIBE; SIN; SINR;
SOBE; SOUR; SOUR; SPACS; STPAR]

DESX6:  600155
Device is not a terminal [BKJFN, SIBE]

DESX7:  600156
Illegal use of parse-only JFN or output
wildcard-designators [CHFDB; DELF; DELNF;
FFUP; GACTF; GFUST; GTFDB; OPENF; PMA;
RCDDIR; RFTAD; RNAMF; SFTAD; SFUST; STPPN;
UFPGS]

DESX8:  600157
File is not on disk [DEQ; ENQ; ENQC;
GFUST; RCDDIR; RPACS; SFBSZ; SFPT; SFUST;
SPACS; STPPN; UFPGS]

DESX9:  601340
Invalid operation for this device [DELJ;
MTOP; MTU%; SDSTS]

DESX10:  601417
Structure is dismounted [GFUST; RCDDIR;
SFUST; STPPN]

DESX11:  602410
Invalid operation for this label type
[MTOP]

DEVX1:  600335
Invalid device designator [ALLOC; ASND;
DEVST; DVCHR; GDSKC; RELD]

DEVX2:  600336
Device already assigned to another job
[ASND; CFIB; CFDB; DIBE; DOBE; MTOP;
RELD; RFPOS; SCTT; SFCOC; SFMOD; SIBE;
SOBE; STI; STO; STPAR]
DEVX3: 600337 Device is not on-line
DEVX5: 601744 No such device [BOOT]
DEVX6: 602275 Job has open JFN on device [RELD]
DEVX7: 602417 Null device name given [COMND]
DIAG10: 601205 Subunit does not exist [DIAG]
DIAG11: 602406 Device is already on-line [DIAG]
DIAG12: 602407 Unit not on-line [DIAG]
DIAG17: 601174 Invalid function [DIAG]
DIAG2X: 601175 Device is not assigned [DIAG]
DIAG3: 601176 Argument block too small [DIAG]
DIAG4: 601177 Invalid device type [DIAG]
DIAG5: 601200 WHEEL, OPERATOR, or MAINTENANCE capability required [DIAG]
DIAG6: 601201 Invalid channel command list [DIAG]
DIAG7: 601202 Illegal to do I/O across page boundary [DIAG]
DIAG8: 601203 No such device [DIAG]
DIAG9: 601204 Unit does not exist [DIAG]
DILFX1: 600464 Invalid date format [IDTIM; IDTNC]
DIRX1: 601913 Invalid directory number [DIRST; GTDAL; PPNST; VACCT]
DIRX2: 601914 Insufficient system resources [DIRST; GFUST; PPNST]
DIRX3: 601915 Internal format of directory is incorrect [DIRST; GFUST; PPNST; VACCT]
DIRX5: 602342 Directory too large
DILFX10: 602010 Cannot delete directory; file still mapped [DELF]
DILFX11: 602011 Cannot delete directory file in this manner [DELF]
DSKDX1: 601343 Channel number too large [DSKDP]
DSKDX2: 601344 Unit number too large [DSKDP]
DSKDX3: 601416 Invalid structure number [DSKDP]
DSKDX4: 601420 Invalid address type specified [DSKDP]
DSKDX5: 601533 Invalid word count
DSKDX6: 601534 Invalid buffer address
DSKDX7: 601365 Invalid structure number [DSKAS]
DSKDX8: 601367 Bit table has not been initialized [DSKAS]
DSKDX9: 601763 Disk assignments and deallocations are currently prohibited [DSKAS]
DSKDX10: 601764 Invalid disk address [DSKAS]
DSKDX11: 601765 Address cannot be deallocated because it has not been assigned [DSKAS]
DSKDX12: 601766 Address cannot be assigned because it is already assigned [DSKAS]
DSMX1: 60055 File(s) not closed [ASND]
DUMP1: 600440 Command list error [DUMP1; DUMP0]
DUMP2: 600441 JFN is not open in dump mode [DUMP1; DUMP0]
DUMP3: 600442 Address error (too big or crosses end of memory) [DUMP1; DUMP0]
DUMP4: 600443 Access error (cannot read or write data in memory) [DUMP1; DUMP0]
DUMP5: 601214 No-wait dump mode not supported for this device [DUMP1; DUMP0]
DUMP6: 601215 Dump mode not supported for this device
ENACX1: 602105 Account validation data base not completely closed
ENACX2: 602106 Cannot get a JFN for <SYSTEM>ACCOUNTS-TABLE.BIN
ENACX3: 602107 Account validation data base too long
ENACX4: 602110 Cannot get an OFN for <SYSTEM>ACCOUNTS-TABLE.BIN
ENOX10: 601066 Invalid argument block length [DEQ; ENQ; ENQC]
ENOX11: 601067 Invalid software interrupt channel number [DEQ; ENQ; ENQC]
ENOX12: 601070 Invalid number of resources requested [ENQ; ENQC]
ENOX13: 601071 Indirect or indexed byte pointer not allowed [DEQ; ENQ; ENQC]
ENOX14: 601072 Invalid byte size [DEQ; ENQ; ENQC]
ENOX15: 601073 ENQ/DEQ capability required [DEQ; ENQ; ENQC]
ENOX16: 601074 WHEEL or OPERATOR capability required [DEQ; ENQ; ENQC]
ENOX17: 601075 Invalid JFN [DEQ; ENQ; ENQC]
ENOX18: 601076 Quote exceeded [DEQ; ENQ; ENQC]
ENOX19: 601077 String too long [DEQ; ENQ; ENQC]
ENOX20: 601055 Invalid function [DEQ; ENQ; ENQC]
ENOX21: 601100 Locked JFN cannot be closed [CLDSF; DEQ; ENQ; ENQC]
ENOX22: 601101 Job is not logged in [DEQ; ENQC]
ENOX23: 601211 Invalid mask block length [ENQ]
ENOX24: 601212 Mismatched mask block lengths [ENQ]
ENOX25: 601216 Level number too small [DEQ; ENQ; ENQC]
ENOX26: 601217 Request and lock level numbers do not match [DEQ; ENQ; ENQC]
ENOX27: 601218 Number of pool and lock resources do not match [DEQ; ENQ; ENQC]
ENOX28: 601219 Lock already requested [ENQ; ENQC]
ENOX29: 60121A Requested locks are not all locked [DEQ; ENQ; ENQC]
ENOX30: 60121B No ENQ on this lock [DEQ; ENQC]
ENOX31: 60121C Invalid access change requested [ENQ; ENQC]
ENOX32: 60121D Invalid number of blocks specified [DEQ; ENQ; ENQC]
ENOX33: 60121E File is not open [FFUFPP]
ENOX34: 60121F File is not on multiple-directory device [FFUFPP]
ENOX35: 601220 No used page found [FFUFPP]
FLOX1: 600650 First character is not blank or numeric [DFIN; FLIN]
FLOX2: 600651 Number too small [DFIN; FLIN]
FLOX3: 600652 Number too large [DFIN; FLIN]
FLOX4: 600653 Invalid format [DFIN; FLIN]
FLOX5: 600660 Column overflow in field 1 or 2 [DFOUT; FLOUT]
FLOX6: 600661 Column overflow in field 3 [DFOUT; FLOUT]
FLOX7: 600662 Invalid format specified [DFOUT; FLOUT]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JYS Error Mnemonics

FRKH1: 600250 Invalid process handle [ADBRK; AIC; CLZFF; DIC; DIR; EIR; EPCAP; ERSTR; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; HFRKH; HFRKS; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRC; RMAP; RPACS; RPACP; RTIW; RUNIT; RWM; SAVE; SCTTY; SCVEC; SETER; SEVEC; SFRACS; SFRAK; SFRKV; SIR; SIRCM; SPJFN; SPACS; SPS; SPLF; SSAVE; STIW; UTFRK; XFORK; XGVEC; XMAP; XSRK; XSFRK; XSVEC]

FRKH2: 600251 Illegal to manipulate a superior process [ADBRK; AIC; CLZFF; DIC; DIR; EIR; EPCAP; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; HFRKH; HFRKS; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRC; RMAP; RPACS; RPACP; RTIW; RWM; SAVE; SCTTY; SCVEC; SETER; SEVEC; SPAC; SPJFN; SSAVE; STIW; UTFRK; XFORK; XGVEC; XSRK; XSFRK; XSVEC]

FRKH3: 600252 Invalid use of multiple process handle [ADBRK; AIC; CLZFF; DIC; DIR; EIR; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; HFRKH; HFRKS; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRC; RMAP; RPACS; RPACP; RTIW; RWM; SAVE; SCVEC; SETER; SEVEC; SFAC; SFORK; SFRKV; SIR; SIRCM; SPAC; SPJFN; SSAVE; STIW; UTFRK; XGVEC; XSFRK; XSIR; XSVEC]

FRKH4: 600253 Process is running [RFACS; SCVEC; SETER; SFAC; SFORK]

FRKH5: 600254 Process has not been started [SFORK; XSFRK]

FRKH6: 600255 All relative process handles in use [CFORK; GFRKH; GFRKS; RFRK]

FRKH7: 601312 Process page cannot exceed 777 [PMAP]

FRKH8: 602170 Illegal to manipulate an execute-only process [ADBRK; AIC; CFORK; DIC; DIR; EIR; GET; IIC; PMAP; RPACS; RFRKH; SFAC; SFRK; SIR; SIRCM; SPAC; SPJFN; STIW; UTFRK; XSFRK; XSVEC]

GACCC1: 601272 Invalid job number [GACCT]

GACCC2: 601273 No such job [GACCT]

GACCC3: 601301 Confidential Information Access capability required [GACCT]

GACTX1: 600540 File is not on multiple-directory device [GACTF]

GACTX2: 600541 File expunged [GACTF]

GACTX3: 601173 Internal format of directory is incorrect [GACTF]

GETX1: 600273 Invalid save file format [GET]

GETX2: 600374 System Special Pages Table full [GET]

GETX3: 601703 Illegal to overlay existing pages [GET]

GETX4: 601557 Illegal to specify .GBASE for multitsection
file [GET]
GETX5: 601560 EXE file directory entry specifies a section-crossing [GET]
GFDBX1: 600424 Invalid displacement [GFDB]
GFDBX2: 600425 Invalid number of words [GFDB]
GFDBX3: 600426 List access required [GFDB]
GFKX1: 601011 Area too small to hold process structure [GFKXS]
GFRX1: 600371 Invalid process handle [GFRK]
GFUSX1: 601371 Invalid function [GFUST]
GFUSX2: 601372 Insufficient system resources [GFUST]
GFUSX3: 601452 File expunged [GFUST]
GFUSX4: 601453 Internal format of directory is incorrect [GFUST]
GJFX1: 600055 Desired JFN invalid [GTJFN]
GJFX2: 600056 Desired JFN not available [GTJFN]
GJFX3: 600057 No JFNs available [GTJFN]
GJFX4: 600060 Invalid character in file name [CRLNM; GTJFN]
GJFX5: 600061 Field cannot be longer than 39 characters [CRLNM; GTJFN]
GJFX6: 600062 Device field not in a valid position [CRLNM; GTJFN]
GJFX7: 600063 Directory field not in a valid position [CRLNM; GTJFN]
GJFX8: 600064 Directory terminating delimiter is not preceded by a valid beginning delimiter [CRLNM; GTJFN]
GJFX9: 600065 More than one name field is not allowed [CRLNM; GTJFN]
GJFX10: 600066 Generation number is not numeric [CRLNM; GTJFN]
GJFX11: 600067 More than one generation number field is not allowed [CRLNM; GTJFN]
GJFX12: 600070 More than one account field is not allowed [CRLNM; GTJFN]
GJFX13: 600071 More than one protection field is not allowed [CRLNM; GTJFN]
GJFX14: 600072 Invalid protection [CRLNM; GTJFN]
GJFX15: 600073 Invalid confirmation character [CRLNM; GTJFN]
GJFX16: 600074 No such device [GTJFN]
GJFX17: 600075 No such directory name [GTJFN]
GJFX18: 600076 No such filename [GTJFN]
GJFX19: 600077 No such file type [GTJFN]
GJFX20: 600100 No such generation [GTJFN]
GJFX21: 600101 File was expunged [GTJFN]
GJFX22: 600102 Insufficient system resources (Job Storage Block full) [CRLNM; GTJFN; LNMST; PPHST]
GJFX23: 600103 Exceeded maximum number of files per directory [GTJFN]
GJFX24: 600104 File not found [GTJFN]
GJFX27: 600107 File already exists (new file required) [GTJFN]
GJFX28: 600110 Device is not on-line [GTJFN]
GJFX30: 600112 Account is not numeric [GTJFN]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

GJFX31: 600113 Invalid wildcard designator [CRLNM; GTJFN]
GJFX32: 600114 No files match this specification [GTJFN]
GJFX33: 600115 Filename was not specified [GTJFN]
GJFX34: 600116 Invalid character "?" in file specification [GTJFN]
GJFX35: 600117 Directory access privileges required [GTJFN]
GJFX36: 600760 Internal format of directory is incorrect [GTJFN]
GJFX37: 601133 Input deleted [GTJFN]
GJFX38: 601164 File not found because output-only device was specified [GTJFN]
GJFX39: 601165 Logical name loop detected [GTJFN]
GJFX40: 601225 Undefined attribute in file specification [GTJFN]
GJFX41: 601277 File name must not exceed 6 characters [GTJFN]
GJFX42: 601300 File type must not exceed 3 characters [GTJFN]
GJFX43: 601754 More than one :T specification is not allowed [GTJFN]
GJFX44: 602012 Account string does not match [GTJFN]
GJFX45: 602060 Illegal to request multiple specifications for the same attribute [GTJFN]
GJFX46: 602061 Attribute value is required [GTJFN]
GJFX47: 602062 Attribute does not take a value [GTJFN]
GJFX48: 602064 GTJFN input buffer is empty [GTJFN]
GJFX49: 602065 Invalid attribute for this device [GTJFN]
GJFX50: 602205 Invalid argument for attribute [GTJFN]
GJFX51: 602211 Byte count too small [GTJFN]
GJFX52: 602420 End of tape encountered while searching for file
GJFX53: 602461 Tape label filename specification exceeds 17 characters [GTJFN]
GNJFX1: 601054 No more files in this specification [GNJFN]
GDKER1: 602220 Illegal function [GETOK%]
GDKER2: 602221 Request denied by Access Control Facility [GETOK%]
GDKER3: 602421 JSYS not executed within ACJ fork [GIVOK%; RCVOK%]
GTABX1: 600267 Invalid table number [GETAB]
GTABX2: 600270 Invalid table index [GETAB]
GTABX3: 600271 GETAB privileges required [GETAB]
GTDX1: 600640 WHEEL or OPERATOR capability required [GTDIR; TIMSG]
GTDX2: 600641 Invalid directory number [GTDIR]
GTHS1: 600704 Unknown host number [GTHST%]
GTHS2: 600705 No number for that host name [GTHST%]
GTHS3: 600707 No string for that host number [GTHST%]
GTJIX1: 601013 Invalid index [GETJI; GTHST%; GTNCP%]
GTJIX2: 601014 Invalid terminal line number [GETJI]
GTJIX3: 601015 Invalid job number [GETJI]
GTJIX4: 601254 No such job [GETJI]
GTNCX1: 600746 Invalid network JFN [GTNCP%]
GTNCX2: 600747 Invalid or inactive NVT [GTNCP%]
HFRX1:  600370 Illegal to halt self with HFRX [HFRK]
MPTX1:  600670 Undefined clock number [MPTIM]
IFIXX1:  600414 Radix is not in range 2 to 10 [NIN]
IFIXX2:  600415 First nonspace character is not a digit [NIN]
IFIXX3:  600416 Overflow (number is greater than 2**35) [NIN]
ILINS1:  600770 Undefined system function [ASNSQ; ATNVT; CVHST; CVSKT; FLHST; GTHST; GTNCP; RCVIM; RELSQ; SNDIM]
ILINS2:  600771 Undefined JSYS [ASNSQ; ATNVT; CVHST; CVSKT; FLHST; GTHST; GTNCP; RCVIM; RELSQ; SNDIM]
ILINS3:  600772 UUD simulation facility not available
ILINS4:  601255 UUD simulation is disabled
ILINS5:  601256 RMS facility is not available [GDVEC; SDVEC]
ILLXO1:  601774 Illegal memory read [SFTAD]
ILLXO2:  601775 Illegal memory write
ILLXO3:  601776 Memory data parity error
ILLXO4:  601777 Reference to non-existent page [PLOCK]
ILLXO5:  602471 Illegal memory reference, section greater than 37
INLNX1:  601001 Index is beyond end of logical name table [INLNM]
INLNX2:  601135 Invalid function [INLNM]
IOX1:  600215 File is not open for reading [BIN; PBIN; RIN; SIN; SINT; DUMP]
IOX2:  600216 File is not open for writing [BOUT; PBOUT; PSOUT; SOUT; DUMP; ROUT; SOUT]
IOX3:  600217 Illegal to change pointer for this opening of file [RIN; ROUT]
IOX4:  600220 End of file reached [BIN; DUMP; PBIN; RIN; SIN; SINT; DUMP]
IOX5:  600221 Device or data error [BIN; BOUT; DUMP; DUMP; DUMP; PBIN; PBOUT; PSOUT; RIN; ROUT; SIN; SINT; SOUT; SOUT]
IOX6:  600222 Illegal to write beyond absolute end of file [PBOUT; PSOUT; ROUT; SOUT; SOUT; BOUT]
IOX7:  601211 Insufficient system resources (Job Storage Block full) [SIN; SINR; SOUT; SOUT]
IOX8:  601212 Monitor internal error [MTUR; SIN; SINT; SOUT; SOUT]
IOX9:  601216 Function legal for sequential write only [SOUT]
IOX10:  601240 Record is longer than user requested [SINR]
IOX11:  601440 Quota exceeded [BOUT; CLOSF; CLZFF; DIVST; DOUT; DIRST; DUMP; ERSTR; FLOUT; GTJNF; JNFS; NOUT; PBOUT; PMP; PMPST; PSOUT; RDTTY; ROUT; SAVE; SOUT; SOUT; SSAVE; TEXTI; UFPGS]
IOX12:  601441 Insufficient system resources (swapping space full)
IOX13:  602227 Invalid segment type
IOX14:  602230 Invalid segment size
IOX15:  602231 Illegal tape format for dump mode [MTOPR]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 USYS Error Mnemonics

IDX17: 602233 Invalid tape label [MTOPR]
IDX20: 602234 Illegal tape record size [MTOPR]
IDX21: 602235 Tape HDR1 missing [MTOPR]
IDX22: 602236 Invalid tape HDR1 sequence number [MTOPR]
IDX23: 602237 Tape label read error [MTOPR]
IDX24: 602240 Logical end of tape encountered [MTOPR]
IDX25: 602241 Invalid tape format [MTOPR]
IDX28: 602243 Tape write date has not expired [MTOPR]
IDX30: 602245 Tape has invalid access character [MTOPR]
IDX31: 602343 Invalid record descriptor in labeled tape [MTOPR]
IDX32: 602422 Tape position is indeterminate [MTOPR]
IDX33: 602423 TTY input buffer full [BOUT; SOUT]
IDX34: 602462 Disk full [BOUT; CLOS; CLZFF; DEVST; DFOUT; DIRST; DUMP; ERSTR; FLOUT; GTFNF; INVS; NOUT; PDSOUT; PMAP; PPSNT; PSTOUT; RDTTY; ROUT; SAVE; SOUT; SOUTR; SSAVE; TEXTI; UFPGS]
IDX35: 602463 Unable to allocate disk - structure damaged [BOUT; CLOS; CLZFF; DEVST; DFOUT; DIRST; DUMP; ERSTR; FLOUT; GTFNF; INVS; NOUT; PDSOUT; PMAP; PPSNT; PSTOUT; RDTTY; ROUT; SAVE; SOUT; SOUTR; SSAVE; TEXTI; UFPGS]

IPCF10: 601027 WHEEL capability required [MUTIL]
IPCF11: 601030 WHEEL or IPCF capability required [MREC; MSSEND; MUTIL]
IPCF12: 601031 No free PID's available [MSSEND; MUTIL]
IPCF13: 601032 PID quota exceeded [MSSEND; MUTIL]
IPCF14: 601033 No PID's available to this job [MREC; MSSEND; MUTIL]
IPCF15: 601034 No PID's available to this process [MREC; MSSEND; MUTIL]
IPCF16: 601035 Receive and message data modes do not match [MREC; MUTIL]
IPCF17: 601036 Argument block too small [MUTIL]
IPCF18: 601037 Invalid MUTIL USYS function [MUTIL]
IPCF19: 601040 No PID for [SYSTEM]INFO [MSSEND; MUTIL]
IPCF20: 601041 Invalid process handle [MUTIL]
IPCF21: 601042 Invalid job number [MUTIL]
IPCF22: 601043 Invalid software interrupt channel number [MUTIL]
IPCF23: 601044 [SYSTEM]INFO already exists [MUTIL]
IPCF24: 601045 Invalid message size [MREC; MSSEND; MUTIL]
IPCF25: 601046 PID does not belong to this job [MREC; MSSEND; MUTIL]
IPCF26: 601047 PID does not belong to this process [MREC; MSSEND; MUTIL]
IPCF27: 601050 PID is not defined [MREC; MSSEND; MUTIL]
IPCF28: 601051 PID not accessible by this process [MREC; MSSEND; MUTIL]
IPCF29: 601052 PID already being used by another process [MREC; MSSEND; MUTIL]
IPCF30: 601053 Job is not logged in [MUTIL]
IPCF31: 601102 Invalid page number [MREC; MSSEND]
IPCF32: 601103 Page is not private [MREC; MSSEND; MUTIL]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

IPCF33: 601130 Invalid index into system PID table
         [MUTIL]
IPCF34: 601320 Cannot receive into an existing page
         [MRECV]
IPCF35: 602125 Invalid IPCF quota [MUTIL]
IPCFX1: 601016 Length of packet descriptor block cannot
         be less than 4 [MRECV; MSEND]
IPCFX2: 601017 No message for this PID [MRECV; MUTIL]
IPCFX3: 601020 Data too long for user's buffer [MRECV;
         MUTIL]
IPCFX4: 601021 Receiver's PID invalid [MRECV; MSEND;
         MUTIL]
IPCFX5: 601022 Receiver's PID disabled [MRECV; MSEND;
         MUTIL]
IPCFX6: 601023 Send quota exceeded [MSEND; MUTIL]
IPCFX7: 601024 Receiver quota exceeded [MSEND; MUTIL]
IPCFX8: 601025 IPCF free space exhausted [MSEND; MUTIL]
IPCFX9: 601026 Sender's PID invalid [MSEND; MUTIL]
KDPX01: 602206 KMC11 not running [BOOT]
KFRKX1: 600365 Illegal to kill top level process [K Fork]
KFRKX2: 600366 Illegal to kill self [K Fork]
LCDBBP: 601475 Bad byte pointer passed to LCS
LCNODD: 601477 LCS No such node
LGNX1: 600010 Invalid account identifier [LOGIN]
LGNX2: 600011 Directory is "files-only" and cannot be
         logged into [ACES; LOGIN]
LGNX3: 600012 Internal format of directory is incorrect
         [LOGIN]
LGNX4: 600013 Invalid password [LOGIN]
LGNX5: 600014 Job is already logged in [LOGIN]
LGNX6: 601337 No more job slots available for logging in
         [LOGIN]
LNGFX1: 601317 Page table does not exist and file not
         open for write [PMAP; UPGS]
LNSTX1: 601002 No such logical name [LN MST]
LNSTX2: 601136 Invalid function [LN MST]
LOCKX1: 601771 Illegal to lock other than a private page
         [PLock]
LOCKX2: 601772 Requested page unavailable [PLock]
LOUTX1: 600035 Invalid job number when logging out own job
         [LGOUT]
LOUTX2: 600036 Invalid job number [LGOUT; MSTR]
LOUTX3: 601227 WHEEL or OPERATOR capability required
         [LGOUT]
LOUTX4: 601230 LGD capability required [LGOUT]
LOUTX5: 601753 Illegal to log out job 0 [LGOUT]
LPINX1: 601333 Invalid unit number [LPINI]
LPINX2: 601334 WHEEL or OPERATOR capability required
         [LPINI]
LPINX3: 601335 Illegal to load RAM or VFU while device is
         OPEN [LPINI]
LSTRX1: 601405 Process has not encountered any errors
         [GETER]
LTLBLX: 602347 Too many user labels
METRX1: 602352 METER% not implemented for this processor
         [METER%]
MONX01:  601727 Insufficient system resources [GETOK%; STR]
MONX02:  601730 Insufficient system resources (USB full) [PVDO%; VACCT]
MONX03:  601731 Monitor Internal error [CRDIR]
MONX04:  601732 Insufficient system resources (swapping space full)
MONX05:  602032 Insufficient system resources (no resident free space) [GETOK%; STR]
MONX06:  602433 Insufficient system resources (no swappable free space) [NODE]
MSTRX1:  601345 Invalid function [MSTR]
MSTRX2:  601346 WHEEL or OPERATOR capability required [MSTR]
MSTRX3:  601347 Argument block too small [MSTR]
MSTRX4:  601350 Insufficient system resources [MSTR]
MSTRX5:  601351 Drive is not on-line [MSTR]
MSTRX6:  601352 HOME blocks are bad [MSTR]
MSTRX7:  601353 Invalid structure name [MSTR]
MSTRX8:  601354 Could not get OFN for ROOT-DIRECTORY [MSTR]
MSTRX9:  601355 Could not MAP ROOT-DIRECTORY [MSTR]
MSTX10:  601356 ROOT-DIRECTORY bad [MSTR]
MSTX11:  601357 Could not initialize Index Table [MSTR]
MSTX12:  601360 Could not OPEN Bit Table File [MSTR]
MSTX13:  601361 Backup copy of ROOT-DIRECTORY is bad [MSTR]
MSTX14:  601362 Invalid channel number [MSTR]
MSTX15:  601363 Invalid unit number [MSTR]
MSTX16:  601364 Invalid controller number [MSTR]
MSTX17:  601421 All units in a structure must be of the same type [MSTR]
MSTX18:  601422 No more units in system [MSTR]
MSTX19:  601423 Unit is already part of a mounted structure [MSTR]
MSTX20:  601424 Data error reading HOME blocks [MSTR]
MSTX21:  601425 Structure is not mounted [MSTR]
MSTX22:  601426 Illegal to change specified bits [MSTR]
MSTX23:  601430 Could not write HOME blocks [MSTR]
MSTX24:  601750 Illegal to dismount the System Structure [MSTR]
MSTX25:  601751 Invalid number of swapping pages [MSTR]
MSTX26:  601752 Invalid number of Front-End-Filesystem pages [MSTR]
MSTX27:  601757 Specified unit is not a disk [MSTR]
MSTX28:  601760 Could not initialize bit table for structure [MSTR]
MSTX29:  601761 Could not reconstruct ROOT-DIRECTORY [MSTR]
MSTX30:  601770 Incorrect Bit Table counts on structure [MSTR]
MSTX31:  602000 Structure already mounted [MSTR]
MSTX32:  602001 Structure was not mounted [GTDIR; MSTR]
MSTX33:  602002 Structure is unavailable for mounting [MSTR]
MSTX34:  602063 Unit is write-locked [MSTR]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

MSTX35:  602201 Too many units in structure [MSTR]
MSTX36:  602223 Illegal while JFNs assigned [MSTR]
MSTX37:  602224 Illegal while connected to structure [MSTR]
MSTX40:  602225 Invalid PS1 channel number given [MSTR]
MSTX41:  601461 Channel does not exist [MSTR]
MSTX42:  601462 Controller does not exist [MSTR]
MTOX10:  601323 VFU or RAM file cannot be OPENed [MTOPR]
MTOX11:  601324 Data too large for buffers [MTOPR]
MTOX12:  601325 Input error or not all data read [MTOPR]
MTOX13:  601326 Argument block too small [MTOPR]
MTOX14:  601327 Invalid software interrupt channel number [MTOPR]
MTOX15:  601331 Device does not have Direct Access [MTOPR]
          [programmable] VFU [MTOPR]
MTOX16:  601332 VFU or Translation RAM file must be on disk [MTOPR]
MTOX17:  601336 Device is not on line [MTOPR]
MTOX18:  601407 Invalid software interrupt channel number [MTOPR]
MTOX19:  601755 Invalid terminal line width [MTOPR]
MTOX20:  601576 Invalid terminal line length [MTOPR]
MTOX21:  601220 Record size was not set before I/O was done [MTOPR]
MTOX22:  601221 Function not legal in dump mode [MTOPR]
MTOX23:  601222 Invalid record size [MTOPR]
MTOX24:  601213 Invalid hardware data mode for magtape [MTOPR]
MTOX25:  601223 Invalid magneto density [MTOPR]
MTOX26:  601226 WHEEL or OPERATOR capability required [MTOPR]
MTOX27:  601274 Argument block too long [MTOPR]
MTOX28:  601322 Output still pending [MTOPR]
NODX02:  602207 Line not turned off [NODE]
NODX03:  602210 Another line already looped [NODE]
NOUTX1:  600407 Radix is not in range 2 to 36 [NOUT]
NOUTX2:  600410 Column overflow [NOUT]
NPX2CL:  602413 Two colons required on node name [COMND]
NPXAMB:  602044 Ambiguous [COMND]
NPXCMJ:  602057 Comma not given [COMND]
NPXCN1:  602062 Invalid character in number [COMND]
NPXN1T:  602053 Invalid device terminator [COMND]
NPXN1W:  602050 Invalid guide word [COMND]
NPXNC:  602051 Not confirmed [COMND]
NPXNMD:  602056 Does not match directory or user name [COMND]
NPXNMN:  602055 Does not match token [COMND]
NPXNOM:  602046 Does not match switch or keyword [COMND]
NPXNOS:  602054 Not a quoted string - quote missing at beginning or end [COMND]
NPXNSM:  602045 Not a switch - does not begin with slash [COMND]
NPXNU1:  602047 Null switch or keyword given [COMND]
NPSX00:  602353 Connection not accepted
NPSX01:  602354 Resource allocation failure
NSPX02: 602355 Destination node does not exist
NSPX03: 602356 Node shutting down
NSPX04: 602357 Destination process does not exist
NSPX05: 602360 Invalid process name
NSPX06: 602361 Destination process queue overflow
NSPX07: 602362 Unspecified error
NSPX08: 602363 Connection aborted by third party
NSPX09: 602364 Link aborted by process
NSPX10: 602365 NSP failure - Flow control violation
NSPX11: 602366 Too many connections to node
NSPX12: 602367 Too many connections to destination process
NSPX13: 602370 Access denied due to unacceptable user name or password
NSPX14: 602371 NSP failure - invalid SERVICES field
NSPX15: 602372 Invalid account
NSPX16: 602373 NSP failure - invalid SEGSIZ field
NSPX17: 602374 Process aborted, timed out, or cancelled request
NSPX18: 602375 No path to destination node
NSPX19: 602376 NSP failure - flow control failure
NSPX20: 602377 NSP failure - invalid DSTADDR
NSPX21: 602400 Disconnect confirmation
NSPX22: 602401 NSP failure - image data field too long
NSPX23: 602411 Invalid NSP reason code
NSPX24: 602456 Node name not assigned to a network node
NSPX25: 602457 Illegal DEChet node number [NODE]
NSPX26: 602460 Table of topology watchers is full [NODE]
NTMX1: 602451 Network Management unable to complete request [NTMAN]
NTWX1: 600737 NET WIZARD capability required [ASNSQ]
ODTX1: 600462 Time zone must be USA or Greenwich [ODTNC]
DPNX1: 600120 File is already open [GNJFN; MTU%; OPENF; RLJFN; RNAMF]
DPNX2: 600121 File does not exist [GET; OPENF]
DPNX3: 600122 Read access required [OPENF]
DPNX4: 600123 Write access required [OPENF]
DPNX5: 600124 Execute access required [OPENF]
DPNX6: 600125 Append access required [OPENF]
DPNX7: 600126 Device already assigned to another job [OPENF]
DPNX8: 600127 Device is not on line [MTU%; OPENF]
DPNX9: 600130 Invalid simultaneous access [OPENF; VACCT]
DPNX10: 600131 Entire file structure full [OPENF]
DPNX12: 600133 List access required [OPENF]
DPNX13: 600134 Invalid access requested [OPENF]
DPNX14: 600135 Invalid mode requested [OPENF]
DPNX15: 600136 Read/write access required [OPENF]
DPNX16: 600137 File has bad index block [OPENF; VACCT]
DPNX17: 600140 No room in job for long file page table [OPENF]
DPNX18: 600141 Unit Record Devices are not available [OPENF]
DPNX19: 600142 IMP is not up
DPNX20: 600143 Host is not up
DPNX21: 600144 Connection refused

197
OPNX22: 600145 Connection byte size does not match
OPNX23: 601132 Disk quota exceeded [OPENF]
OPNX25: 601224 Device is write locked [OPENF; SFTAD]
OPNX26: 601410 Illegal to open a string pointer [OPENF]
OPNX30: 602326 File has archive status, modification is prohibited [ARCF]
OPNX31: 602327 File is off-line [ARCF; OPENF]
PDXO21: 600036 Create operation impossible [PDXO2]
PDXO23: 601555 Address in .PODAT block must be in strict ascending order [PVDOP%]
PDXO30: 601556 Address in .PODAR must be a program data vector address [PVDOP%]
PEKX2: 600617 Read access failure on monitor page [PEEK]
PMAPX1: 600240 Invalid access requested [PMAP]
PMAPX2: 600241 Invalid use of PMAP [PMAP]
PMAPX3: 601104 Illegal to move shared page into file [PMAP]
PMAPX4: 601105 Illegal to move file page into process [PMAP]
PMAPX5: 601106 Illegal to move special page into file [PMAP]
PMAPX6: 601107 Disk quota exceeded [PMAP]
PMAPX7: 601415 Illegal to map file on dismounted structure [PMAP]
PMAPX8: 602464 Indirect page map loop detected [PMAP]
PMCLX1: 602005 Invalid page state or state transition [PMCLT]
PMCLX2: 602006 Requested physical page is unavailable [PMCLT]
PMCLX3: 602007 Requested physical page contains errors [PMCLT]
PMCLX4: 602165 No more error information [PMCLT]
PPNX1: 601444 Invalid PPN [PPNST]
PPNX2: 601445 Structure is not mounted [PPNST]
PRAX1: 601263 Invalid PRARG function code [PRARG]
PRAX2: 601264 No room in monitor data base for argument block [PRARG]
PRAX3: 601270 PRARG argument block too large [PRARG]
RCDX1: 601376 Insufficient system resources [RCDIR]
RCDX2: 601377 Invalid directory specification [ACCES; RCDIR]
RCDX3: 601400 Invalid structure name [RCDIR]
RCDX4: 601401 Monitor internal error [RCDIR; RCUSR]
RCUX1: 601402 Insufficient system resources [RCUSR]
RDTX1: 601010 Invalid string pointer [RDTTY; TEXTI; WILD%]
RIRX1: 602426 RIR SYS incompatible with previous XSIRX% [RIR]
RJFNX1: 600165 File is not closed [RLJFNN]
RJFNX2: 600166 JFN is being used to accumulate filename [RLJFNN]
RJFNX3: 600167 JFN is not accessible by this process [RLJFNN]
RNAMX1: 600450 Files are not on same device [RNAMF]
RNAMX2: 600451 Destination file expunged [RNAMF]
<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNAMX3</td>
<td>600452</td>
<td>Write or owner access to destination file required [RNAMF]</td>
</tr>
<tr>
<td>RNAMX4</td>
<td>600453</td>
<td>Quota exceeded in destination of rename [RNAMF]</td>
</tr>
<tr>
<td>RNAMX5</td>
<td>600750</td>
<td>Destination file is not closed [RNAMF]</td>
</tr>
<tr>
<td>RNAMX6</td>
<td>600751</td>
<td>Destination file has bad page table [RNAMF]</td>
</tr>
<tr>
<td>RNAMX7</td>
<td>600752</td>
<td>Source file expunged [RNAMF]</td>
</tr>
<tr>
<td>RNAMX8</td>
<td>600753</td>
<td>Write or owner access to source file required [RNAMF]</td>
</tr>
<tr>
<td>RNAMX9</td>
<td>600754</td>
<td>Source file is nonexistent [RNAMF]</td>
</tr>
<tr>
<td>RNMX10</td>
<td>600755</td>
<td>Source file is not closed [RNAMF]</td>
</tr>
<tr>
<td>RNMX11</td>
<td>600756</td>
<td>Source file has bad page table [RNAMF]</td>
</tr>
<tr>
<td>RNMX12</td>
<td>600757</td>
<td>Illegal to rename to self [RNAMF]</td>
</tr>
<tr>
<td>RNMX13</td>
<td>601454</td>
<td>Insufficient system resources [RNAMF]</td>
</tr>
<tr>
<td>RSCNX1</td>
<td>600361</td>
<td>Overflown rescan buffer, input string truncated [RSCAN]</td>
</tr>
<tr>
<td>RSCNX2</td>
<td>600362</td>
<td>Invalid function code [RSCAN]</td>
</tr>
<tr>
<td>RUNTX1</td>
<td>600273</td>
<td>Invalid process handle -3 or -4 [RUNTM]</td>
</tr>
<tr>
<td>SACTX1</td>
<td>600530</td>
<td>File is not on multiple-directory device [SACTF]</td>
</tr>
<tr>
<td>SACTX2</td>
<td>600531</td>
<td>Insufficient system resources (job storage block full) [SACTF]</td>
</tr>
<tr>
<td>SACTX3</td>
<td>600532</td>
<td>Directory requires numeric account [SACTF]</td>
</tr>
<tr>
<td>SACTX4</td>
<td>600533</td>
<td>Write or owner access required [SACTF]</td>
</tr>
<tr>
<td>SAVX1</td>
<td>601330</td>
<td>Illegal to save files on this device [SAVE]</td>
</tr>
<tr>
<td>SCTX1</td>
<td>601550</td>
<td>Invalid function code [SCTTY]</td>
</tr>
<tr>
<td>SCTX2</td>
<td>601551</td>
<td>Terminal already in use as controlling terminal [SCTTY]</td>
</tr>
<tr>
<td>SCTX3</td>
<td>601552</td>
<td>Illegal to redefine the job's controlling terminal [SCTTY]</td>
</tr>
<tr>
<td>SCTX4</td>
<td>601553</td>
<td>SC/SCT capability required [SCTTY]</td>
</tr>
<tr>
<td>SEVEX1</td>
<td>600610</td>
<td>Entry vector length is not less than 1000 [SEVEC; XSVEC%]</td>
</tr>
<tr>
<td>SFBXS1</td>
<td>600210</td>
<td>Illegal to change byte size for this opening of file [SFBSZ]</td>
</tr>
<tr>
<td>SFBXS2</td>
<td>600211</td>
<td>Invalid byte size [OPENF; SFBSZ]</td>
</tr>
<tr>
<td>SFTXT1</td>
<td>600175</td>
<td>File is not open [SFPTR]</td>
</tr>
<tr>
<td>SFTXT2</td>
<td>600176</td>
<td>Illegal to reset pointer for this file [BKJFN; SFTPR]</td>
</tr>
<tr>
<td>SFTXT3</td>
<td>600177</td>
<td>Invalid byte number [BKJFN; SFTPR]</td>
</tr>
<tr>
<td>SFRVX1</td>
<td>600377</td>
<td>Invalid position in entry vector [SFRKV]</td>
</tr>
<tr>
<td>SFSUX1</td>
<td>601372</td>
<td>Invalid function [SFUST]</td>
</tr>
<tr>
<td>SFSUX2</td>
<td>601373</td>
<td>Insufficient system resources [SFUST]</td>
</tr>
<tr>
<td>SFSUX4</td>
<td>601700</td>
<td>File expunged [SFUST]</td>
</tr>
<tr>
<td>SFSUX5</td>
<td>601701</td>
<td>Write or owner access required [SFUST]</td>
</tr>
<tr>
<td>SFSUX6</td>
<td>601702</td>
<td>No such user name [SFUST]</td>
</tr>
<tr>
<td>SIRX1</td>
<td>600670</td>
<td>Table address is not greater than 20 [SIR; XSIR%]</td>
</tr>
<tr>
<td>SIRX2</td>
<td>602425</td>
<td>SIR USYS invoked from non-zero section [SIR]</td>
</tr>
<tr>
<td>SJBX1</td>
<td>601244</td>
<td>Invalid function [SETJB]</td>
</tr>
<tr>
<td>SJBX2</td>
<td>601245</td>
<td>Invalid magtape density [SETUB]</td>
</tr>
<tr>
<td>SJBX3</td>
<td>601246</td>
<td>Invalid magtape data mode [SETJB]</td>
</tr>
<tr>
<td>SJBX4</td>
<td>601251</td>
<td>Invalid job number [SETUB]</td>
</tr>
</tbody>
</table>
SUBJX5: 601252 Job is not logged in [SETJB]
SUBJX6: 601253 WHEEL or OPERATOR capability required [SETJB]
SUBX7: 602077 Remark exceeds 30 characters [SETJB]
SUBX8: 601455 Illegal to perform this function [SETJB]
SUPRX1: 601276 Job is not logged in [SUPRI]
SKDX1: 602247 Cannot change class [SKED%]
SMAPX1: 602431 Attempt to delete a section still shared [SKPIR]
SMAPX2: 602465 Indirect section map loop detected [SKPIR]
SMONX1: 600516 WHEEL or OPERATOR capability required [SMON]
SMONX2: 601250 Invalid SMON function [SMON]
SNIDX1: 600732 Invalid message size [SNDIM]
SNIDX2: 600733 Insufficient system resources (No buffers available) [SNDIM]
SNIDX3: 600734 Illegal to specify NCP links 0 - 72 [SNDIM]
SNIDX4: 600735 Invalid header value for this queue [SNDIM]
SNIDX5: 600736 IMP down [SNDIM]
SNOP10: 601121 Breakpoints already inserted [SNOOP]
SNOP11: 601122 Breakpoints not inserted [SNOOP]
SNOP12: 601123 Invalid format for program name symbol [SNOOP]
SNOP13: 601124 No such program name symbol [SNOOP]
SNOP14: 601125 No such symbol [SNOOP]
SNOP15: 601126 Not enough free pages for snooping [SNOOP]
SNOP16: 601127 Multiply-defined symbol [SNOOP]
SNOP17: 601131 Breakpoint already defined [SNOOP]
SNOP18: 601163 Data page is not private or copy-or-write [SNOOP]
SNOPX1: 601110 WHEEL or OPERATOR capability required [SNOOP]
SNOPX2: 601111 Invalid function [SNOOP]
SNOPX3: 601112 .SNPLC function must be first [SNOOP]
SNOPX4: 601113 Only one .SNPLC function allowed [SNOOP]
SNOPX5: 601114 Invalid page number [SNOOP]
SNOPX6: 601115 Invalid number of pages to lock [SNOOP]
SNOPX7: 601116 Illegal to define breakpoints after inserting them [SNOOP]
SNOPX8: 601117 Breakpoint is not set on instruction [SNOOP]
SNOPX9: 601120 No more breakpoints allowed [SNOOP]
SPACX1: 600245 Invalid access requested [SPACS]
SPLFX1: 600250 Process is not inferior or equal to self [SPLFK]
SPLFX2: 600261 Process is not inferior to self [SPLFK]
SPLFX3: 600262 New superior process is inferior to intended inferior [SPLFK]
SPLX1: 601144 Invalid function [SPDOL]
SPLX2: 601145 Argument block too small [SPDOL]
SPLX3: 601146 Invalid device designator [SPDOL]
SPLX4: 601147 WHEEL or OPERATOR capability required [SPDOL]
SPLX5: 601150 Illegal to specify 0 as generation number
for first file [SPOOL]
SPLX6: 601450 No directory to write spooled files into [SPOOL]
SQX1: 600742 Special network queue handle out of range [RCVIM; SNDIM]
SQX2: 600743 Special network queue not assigned [RCVIM; SNDIM]
SSAVX1: 600600 Illegal to save files on this device [GET; SSAVE]
SSAVX2: 600601 Page count (left half of table entry) must be negative [SSAVE]
SSAVX3: 601232 Insufficient system resources (Job Storage Block full) [SSAVE]
SSAVX4: 601233 Directory area of EXE file is more than one page [SSAVE]
SSAVX5: 601500 Number of PDVs grew during save [SSAVE]
STADX1: 600275 WHEEL or OPERATOR capability required [STAD]
STADX2: 600276 Invalid date or time [STAD; STAD]
STDI1X: 600203 The STDIR JSYS has been replaced by RCDIR and RCUSR
STDVX1: 600332 No such device [MSTR; PPNST; STDEV]
STRX01: 601436 Structure is not mounted [ACCES; DIRST; MSTR; PPNST; RCDIR; VACCT]
STRX02: 601437 Insufficient system resources [ACCES; MSTR; STPPN]
STRX03: 601442 No such directory name [ACCES; STPPN]
STRX04: 601443 Ambiguous directory specification [ACCES; STPPN]
STRX06: 601747 No such user number [PPNST]
STRX07: 602142 Invalid user number [RCUSR]
STRX08: 602143 Invalid user name [RCUSR]
STRX09: 602222 Prior structure mount required [ACCES; GNUM; GTJFN]
STYPX1: 601414 Invalid terminal type [STTYP]
SWJFX1: 601406 Illegal to swap same JFN [SWJFN]
SWJFX2: 602242 Illegal to swap ATS JFN
SYEX1: 601206 Unreasonable SPEAR block size [SYERR]
SYEX2: 601207 No buffer space available for SPEAR [SYERR]
TADDX1: 601235 Table is full [TBADD]
TADDX2: 601236 Entry is already in table [TBADD]
TDELS1: 601234 Table is empty [TBDEL]
TDELS2: 601403 Invalid table entry location [TBDEL]
TERMX1: 600350 Invalid terminal code [ATI; DTI]
TFRKX1: 600375 Invalid function code [TFORK]
TFRKX2: 600376 Unassigned process handle or not immediate inferior [TFORK]
TFRKX3: 600411 Process not frozen [TFORK]
TILFX1: 600465 Invalid time format [IDTIM; IDTNC]
TIMEX1: 600460 Time cannot be greater than 24 hours [HSYS; IDCNV; OD CNV; DDTIM]
TIMEX2: 601302 Downtime cannot be more than 7 days in the future [HSYS]
TIMX1: 601541 No system date and time [TIMER]
TIMX1: 601157 Invalid function [TIMER]
TIMX2: 601160 Invalid process handle [TIMER]
TIMX2: 601161 Time limit already set [TIMER]
TIMX4: 601162 Illegal to clear time limit [TIMER]
TIMX5: 601404 Invalid software interrupt channel number [TIMER]
TIMX6: 601535 Time has already passed [TIMER]
TIMX7: 601536 No space available for a clock [TIMER]
TIMX8: 601537 User clock allocation exceeded [TIMER]
TIMX9: 601540 No such clock entry found [TIMER]
TLNKX1: 600351 Illegal to set remote to object before object to remote [TLINK]
TLNKX2: 600356 Link was not received within 15 seconds [TLINK]
TLNKX3: 600357 Links full [TLINK]
TLUKX1: 601237 Internal format of table is incorrect [TLUKX]
TMONX1: 601247 Invalid TMON function [TMON]
TTMSX1: 602432 Could not send message within timeout interval [TTMSX1]
TTYXO1: 602030 Line is not active [BFUJF; CFIOB; CFB; DIBE; DOBE; GTYP; MTPR; OPENF; RFCDC; RFMOD; RFPOS; SFCOC; SFMOD; SPOS; SOBE; ST; STD; STPAR; STTYP; TLINK]
TTYXO2: 602455 Illegal character specified
TTYX1: 600360 Device is not a terminal [STI; STD]
UFPGX1: 601316 File is not opened for write [UFPGX]
USGX01: 602113 Invalid USAGE entry type code [USAGE]
USGX02: 602116 Item not found in argument list [USAGE]
USGX03: 602124 Default item not allowed [USAGE]
UTSTX1: 602013 Invalid function code [UTEST]
UTSTX2: 602014 Area of code too large to test [UTEST]
UTSTX3: 602015 UTEST facility in use by another process [UTEST]
VACCCX: 602111 Invalid account [CACCT; SACTF; VACCT]
VACCCX1: 602112 Account string exceeds 39 characters [CACCT; SACTF; VACCT; CMD]
VACCCX2: 602126 Account has expired [SACTF; VACCT]
WHELX1: 600614 WHEEL or OPERATOR capability required [DEL; DSKAS; DSKOP; MDT%; SUPRI; SPIW; ST]
WILDX1: 601460 Second JFN cannot be wild [WILDX]
XSEVX1: 602472 Illegal entry vector type [XSEVX; XSGEVEV]
XSEVX2: 602473 Illegal entry vector length [XSEVX]
XSEVX3: 602474 Cannot get extended values with this monitor call [GCVE; GDVE]
XSIRX1: 602424 Channel table crosses section boundary [XSIRX]
XSIRX2: 602427 Level table crosses section boundary [XSIRX]
ZONEX1: 600461 Time zone out of range [IDCNV; ODCNV; ODTN]
### Pointer Formats

**One Word Global Byte Pointer**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>6</td>
<td>17</td>
<td>18</td>
<td>35</td>
</tr>
</tbody>
</table>

Legal sizes and positions are as follows:

<table>
<thead>
<tr>
<th>Size</th>
<th>Positions (Octal)</th>
<th>P &amp; S Value (Octal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>44 45 46 47 50 51 62 63</td>
<td>64 65 66</td>
</tr>
<tr>
<td>7</td>
<td>44 35 26 17 10 1 66</td>
<td>67 68</td>
</tr>
<tr>
<td>8</td>
<td>44 34 24 14 4 60</td>
<td>67 68</td>
</tr>
<tr>
<td>9</td>
<td>44 33 22 11 0 73</td>
<td>74 75</td>
</tr>
<tr>
<td>18</td>
<td>44 22 0</td>
<td>74 75</td>
</tr>
</tbody>
</table>

**Global Format Indirect Word**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>17</td>
</tr>
</tbody>
</table>

---

203
PDP-10 INSTRUCTION SET

Arithmetic Testing Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOBJP</td>
<td>252</td>
<td>(AC) + 1, 1 --&gt; (AC); If AC &gt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>AOBUN</td>
<td>253</td>
<td>(AC) + 1, 1 --&gt; (AC); If AC &lt; 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>CAI</td>
<td>300</td>
<td>No-op</td>
</tr>
<tr>
<td>CAIL</td>
<td>301</td>
<td>If (AC) &lt; E: skip</td>
</tr>
<tr>
<td>CAIE</td>
<td>302</td>
<td>If (AC) = E: skip</td>
</tr>
<tr>
<td>CAILE</td>
<td>303</td>
<td>If (AC) &lt;= E: skip</td>
</tr>
<tr>
<td>CAIA</td>
<td>304</td>
<td>Skip</td>
</tr>
<tr>
<td>CAIGE</td>
<td>305</td>
<td>If (AC) &gt;= E: skip</td>
</tr>
<tr>
<td>CAIIN</td>
<td>306</td>
<td>If (AC) ≠ E: skip</td>
</tr>
<tr>
<td>CAIG</td>
<td>307</td>
<td>If (AC) &gt; E: skip</td>
</tr>
<tr>
<td>CAM</td>
<td>310</td>
<td>No-op</td>
</tr>
<tr>
<td>CAML</td>
<td>311</td>
<td>If (AC) &lt; (E): skip</td>
</tr>
<tr>
<td>CAMLE</td>
<td>312</td>
<td>If (AC) = (E): skip</td>
</tr>
<tr>
<td>CAMLE</td>
<td>313</td>
<td>If (AC) &lt;= (E): skip</td>
</tr>
<tr>
<td>CAMA</td>
<td>314</td>
<td>Skip</td>
</tr>
<tr>
<td>CAME</td>
<td>315</td>
<td>If (AC) &gt;= (E): skip</td>
</tr>
<tr>
<td>CAMN</td>
<td>316</td>
<td>If (AC) ≠ (E): skip</td>
</tr>
<tr>
<td>CAMG</td>
<td>317</td>
<td>If (AC) &gt; (E): skip</td>
</tr>
<tr>
<td>JUMP</td>
<td>320</td>
<td>No-op</td>
</tr>
<tr>
<td>JUMPL</td>
<td>321</td>
<td>If (AC) &lt; 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>JUMPE</td>
<td>322</td>
<td>If (AC) = 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>JUMPLE</td>
<td>323</td>
<td>If (AC) &lt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>JUMPA</td>
<td>324</td>
<td>E --&gt; (PC)</td>
</tr>
<tr>
<td>JUMPG</td>
<td>325</td>
<td>If (AC) &gt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>JUMPN</td>
<td>326</td>
<td>If (AC) ≠ 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>JUMPG</td>
<td>327</td>
<td>If (AC) &gt; 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>SKIP</td>
<td>330</td>
<td>If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPL</td>
<td>331</td>
<td>If (E) &lt; 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPE</td>
<td>332</td>
<td>If (E) = 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPE</td>
<td>333</td>
<td>If (E) &lt;= 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPA</td>
<td>334</td>
<td>Skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPE</td>
<td>335</td>
<td>If (E) &gt;= 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPE</td>
<td>336</td>
<td>If (E) ≠ 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIP</td>
<td>337</td>
<td>If (E) &gt; 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>ADJ</td>
<td>340</td>
<td>(AC) + 1 --&gt; (AC)</td>
</tr>
<tr>
<td>ADJL</td>
<td>341</td>
<td>(AC) + 1 --&gt; (AC); If (AC) &lt; 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJE</td>
<td>342</td>
<td>(AC) + 1 --&gt; (AC); If (AC) = 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJL</td>
<td>343</td>
<td>(AC) + 1 --&gt; (AC); If (AC) &lt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJA</td>
<td>344</td>
<td>(AC) + 1 --&gt; (AC); E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJG</td>
<td>345</td>
<td>(AC) + 1 --&gt; (AC); If (AC) &gt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJG</td>
<td>346</td>
<td>(AC) + 1 --&gt; (AC); If (AC) ≠ 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJG</td>
<td>347</td>
<td>(AC) + 1 --&gt; (AC); If (AC) &gt; 0: E --&gt; (PC)</td>
</tr>
</tbody>
</table>

205
TOPS-20 Monitor Calls Quick Reference Guide
PDP-10 Instruction Set

A0S 350 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC)
A0SL 351 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
         If (E) < O: skip
A0SE 352 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
         If (E) = O: skip
A0SLE 353 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
       If (E) ≤ 0: skip
A0SA 354 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
       skip
A0SGE 355 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
    If (E) ≥ O: skip
A0SN 356 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
    If (E) ≤ 0: skip
A0SG 357 (E) + 1 --> (E); If (AC) ≠ O: (E) --> (AC);
     If (E) > O: skip

S0J 360 (AC) - 1 --> (AC)
S0JL 361 (AC) - 1 --> (AC); If (AC) < O: E --> (PC)
S0JE 362 (AC) - 1 --> (AC); If (AC) = O: E --> (PC)
S0JLE 363 (AC) - 1 --> (AC); If (AC) ≤ 0: E --> (PC)
S0JA 364 (AC) - 1 --> (AC); E --> (PC)
S0JGE 365 (AC) - 1 --> (AC); If (AC) ≥ O: E --> (PC)
S0JN 366 (AC) - 1 --> (AC); If (AC) ≠ O: E --> (PC)
S0JG 367 (AC) - 1 --> (AC); If (AC) > O: E --> (PC)

S0S 370 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC)
S0SL 371 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         If (E) < O: skip
S0SE 372 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         If (E) = O: skip
S0SLE 373 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         If (E) ≤ 0: skip
S0SA 374 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         skip
S0SG 375 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         If (E) ≥ O: skip
S0SN 376 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         If (E) ≠ O: skip
S0SG 377 (E) - 1 --> (E); If AC ≠ O: (E) --> (AC);
         If (E) > O: skip

Boolean Instructions

SETZ 400 O --> (AC)
SETZI 401 O --> (AC)
SETZM 402 O --> (E)
SETZB 403 O --> (AC)(E)
SETM 414 (E) --> (AC)
SETMI 415 O,E --> (AC)
SETHM 416 (E) --> (E) [no-op]
SETHB 417 (E) --> (AC)(E)
SETA 424 (AC) --> (AC) [no-op]
SETAI 425 (AC) --> (AC) [no-op]
SETAM 426 (AC) --> (E)

206
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETAB</td>
<td>(AC) --&gt; (E)</td>
</tr>
<tr>
<td>SETO</td>
<td>7777777777777777 --&gt; (AC)</td>
</tr>
<tr>
<td>SETOI</td>
<td>7777777777777777 --&gt; (AC)</td>
</tr>
<tr>
<td>SETOM</td>
<td>7777777777777777 --&gt; (E)</td>
</tr>
<tr>
<td>SETOB</td>
<td>7777777777777777 --&gt; (AC)(E)</td>
</tr>
<tr>
<td>SETCA</td>
<td>~(AC) --&gt; (AC)</td>
</tr>
<tr>
<td>SETCAI</td>
<td>~(AC) --&gt; (AC)</td>
</tr>
<tr>
<td>SETCAM</td>
<td>~(AC) --&gt; (E)</td>
</tr>
<tr>
<td>SETCAB</td>
<td>~(AC) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>SETCM</td>
<td>~(E) --&gt; (AC)</td>
</tr>
<tr>
<td>SETCMI</td>
<td>~(0,E] --&gt; (AC)</td>
</tr>
<tr>
<td>SECTCM</td>
<td>~(E) --&gt; (E)</td>
</tr>
<tr>
<td>SECTCMB</td>
<td>~(E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>AND</td>
<td>(AC) &amp; (E) --&gt; (AC)</td>
</tr>
<tr>
<td>ANDI</td>
<td>(AC) &amp; 0,E --&gt; (AC)</td>
</tr>
<tr>
<td>ANDM</td>
<td>(AC) &amp; (E) --&gt; (E)</td>
</tr>
<tr>
<td>ANDB</td>
<td>(AC) &amp; (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>ANDCA</td>
<td>~(AC) &amp; (E) --&gt; (AC)</td>
</tr>
<tr>
<td>ANDCAI</td>
<td>~(AC) &amp; 0,E --&gt; (AC)</td>
</tr>
<tr>
<td>ANDCAM</td>
<td>~(AC) &amp; (E) --&gt; (E)</td>
</tr>
<tr>
<td>ANDCAB</td>
<td>~(AC) &amp; (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>ANDCM</td>
<td>~(AC) &amp; ~(E) --&gt; (AC)</td>
</tr>
<tr>
<td>ANDCMI</td>
<td>~(AC) &amp; ~(0,E] --&gt; (AC)</td>
</tr>
<tr>
<td>ANDCM2</td>
<td>~(AC) &amp; ~(E) --&gt; (E)</td>
</tr>
<tr>
<td>ANDCMB</td>
<td>~(AC) &amp; ~(E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>ANDCB</td>
<td>~(AC) &amp; ~(E) --&gt; (AC)</td>
</tr>
<tr>
<td>ANDCBI</td>
<td>~(AC) &amp; ~(0,E] --&gt; (AC)</td>
</tr>
<tr>
<td>ANDCBM</td>
<td>~(AC) &amp; ~(E) --&gt; (E)</td>
</tr>
<tr>
<td>ANDCBB</td>
<td>~(AC) &amp; ~(E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>IOR</td>
<td>(AC) I (E) --&gt; (AC)</td>
</tr>
<tr>
<td>IDRI</td>
<td>(AC) I 0,E --&gt; (AC)</td>
</tr>
<tr>
<td>IORM</td>
<td>(AC) I (E) --&gt; (E)</td>
</tr>
<tr>
<td>IORB</td>
<td>(AC) I (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>ORCA</td>
<td>~(AC) I (E) --&gt; (AC)</td>
</tr>
<tr>
<td>ORCAI</td>
<td>~(AC) I 0,E --&gt; (AC)</td>
</tr>
<tr>
<td>ORCAM</td>
<td>~(AC) I (E) --&gt; (E)</td>
</tr>
<tr>
<td>ORCAB</td>
<td>~(AC) I (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>ORCM</td>
<td>(AC) I ~(E) --&gt; (AC)</td>
</tr>
<tr>
<td>ORCMI</td>
<td>(AC) I ~(0,E] --&gt; (AC)</td>
</tr>
<tr>
<td>ORCM2</td>
<td>(AC) I ~(E) --&gt; (E)</td>
</tr>
<tr>
<td>ORCMB</td>
<td>(AC) I ~(E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>ORCB</td>
<td>~(AC) I ~(E) --&gt; (Ac)</td>
</tr>
<tr>
<td>ORCBI</td>
<td>~(AC) I ~(0,E] --&gt; (AC)</td>
</tr>
<tr>
<td>ORCBB</td>
<td>~(AC) I ~(E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>XOR</td>
<td>(AC) ^I (E) --&gt; (AC)</td>
</tr>
<tr>
<td>XORI</td>
<td>(AC) ^I 0,E --&gt; (AC)</td>
</tr>
<tr>
<td>XORM</td>
<td>(AC) ^I (E) --&gt; (E)</td>
</tr>
<tr>
<td>XORB</td>
<td>(AC) ^I (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>EQV</td>
<td>~(AC) ^I (E] --&gt; (AC)</td>
</tr>
<tr>
<td>EQVI</td>
<td>~(AC) ^I 0,E --&gt; (AC)</td>
</tr>
</tbody>
</table>
EQWM 446 \(~[AC] \land (E) \rightarrow (E)\)
EQVB 447 \(~[(AC) \land (E)] \rightarrow (AC)(E)\)

**Byte Instructions**

IBP 133 Linear operations on pointer in E or E+1
AC=0  if P-S >= 0: P-S \rightarrow P
       if P-S < 0: Y+1 \rightarrow Y; 36-S \rightarrow P

ADJP 133 Array operations on pointer in E or E+1
AC \neq 0  Let A = REMAINDER ((36-P)/S)
        if S > 36-A: 1 \rightarrow NO DIVIDE
        if S = 0: (E) \rightarrow (AC) or
               (E,E+1) \rightarrow (AC,AC+1)
        if 0 < S < 36-A: make copy C of (E) or
               (E,E+1)
        Compute: (AC) + ((36-P)/S) * Q * BYTES/WORD + R
               1 <= R <= BYTES/WORD = ((36-P)/S) + (P/S)
        Y[C] + Q \rightarrow Y[C]
        36 - (R = S) - A \rightarrow P[C]
        C \rightarrow (AC) or (AC,AC+1)

LDB 135 BYTE IN ((E)) \rightarrow (AC)
DPB 137 BYTE IN (AC) \rightarrow BYTE IN ((E))
ILDB 134 IBP and LDB
IDPB 136 IBP and DPB

**Fixed-point Arithmetic Instructions**

ADD 270 \((AC) + (E) \rightarrow (AC)\)
ADDI 271 \((AC) + O,E \rightarrow (AC)\)
ADDM 272 \((AC) + (E) \rightarrow (E)\)
ADDB 273 \((AC) + (E) \rightarrow (AC)(E)\)

SUB 274 \((AC) - (E) \rightarrow (AC)\)
SUBI 275 \((AC) - O,E \rightarrow (AC)\)
SUBM 276 \((AC) - (E) \rightarrow (E)\)
SUBB 277 \((AC) - (E) \rightarrow (AC)(E)\)

IMUL 220 \((AC) * (E) \rightarrow (AC)[1]\)
IMULI 221 \((AC) * O,E \rightarrow (AC)[1]\)
IMULM 222 \((AC) * (E) \rightarrow (E)[1]\)
IMULB 223 \((AC) * (E) \rightarrow (AC)(E)[1]\)

MUL 224 \((AC) * (E) \rightarrow (AC,AC+1)\)
MULI 225 \((AC) * O,E \rightarrow (AC,AC+1)\)
MULM 226 \((AC) * (E) \rightarrow (E)[2]\)
MULB 227 \((AC) * (E) \rightarrow (AC,AC+1)(E)\)

IDIV 230 \((AC) \div (E) \rightarrow (AC); \text{REMAINDER} \rightarrow (AC+1)\)
[1] High order word of product is discarded.
[2] LOW order word of product is discarded.

208
### TOPS-20 Monitor Calls Quick Reference Guide

#### PDP-10 Instruction Set

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDIVI 231</td>
<td>(AC) / O,E --&gt; (AC); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>IDIVM 232</td>
<td>(AC) / (E) --&gt; (E); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>IDIVB 233</td>
<td>(AC) / (E) --&gt; (AC)(E); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>DIV 234</td>
<td>(AC,AC+1) / (E) --&gt; (AC); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>DIVI 235</td>
<td>(AC,AC+1) / 0,E --&gt; (AC); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>DIVM 236</td>
<td>(AC,AC+1) / (E) --&gt; (E); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>DIVB 237</td>
<td>(AC,AC+1) / (E) --&gt; (AC)(E); REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>DADD 114</td>
<td>(AC,AC+1) + (E,E+1) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DSUB 115</td>
<td>(AC,AC+1) - (E,E+1) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DMUL 116</td>
<td>(AC,AC+1) * (E,E+1) --&gt; (AC,AC+1,AC+2,AC+3)</td>
</tr>
<tr>
<td>DDIV 117</td>
<td>(AC,AC+1,AC+2,AC+3) / (E,E+1) --&gt; (AC,AC+1)</td>
</tr>
</tbody>
</table>

### Floating-point Arithmetic Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAD 140</td>
<td>(AC) + (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FADL 141</td>
<td>(AC) + (E) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>FADM 142</td>
<td>(AC) + (E) --&gt; (E)</td>
</tr>
<tr>
<td>FADB 143</td>
<td>(AC) + (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FADR 144</td>
<td>(AC) + (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FADRI 145</td>
<td>(AC) + E.O --&gt; (AC)</td>
</tr>
<tr>
<td>FADRM 146</td>
<td>(AC) + (E) --&gt; (E)</td>
</tr>
<tr>
<td>FADRB 147</td>
<td>(AC) + (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FSB 150</td>
<td>(AC) - (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FSBL 151</td>
<td>(AC) - (E) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>FSBM 152</td>
<td>(AC) - (E) --&gt; (E)</td>
</tr>
<tr>
<td>FSBB 153</td>
<td>(AC) - (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FSBR 154</td>
<td>(AC) - (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FSBRI 155</td>
<td>(AC) - E.O --&gt; (AC)</td>
</tr>
<tr>
<td>FSBRM 156</td>
<td>(AC) - (E) --&gt; (E)</td>
</tr>
<tr>
<td>FSBRB 157</td>
<td>(AC) - (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FMP 160</td>
<td>(AC) * (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FMPL 161</td>
<td>(AC) * (E) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>FMPM 162</td>
<td>(AC) * (E) --&gt; (E)</td>
</tr>
<tr>
<td>FMPS 163</td>
<td>(AC) * (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FMPR 164</td>
<td>(AC) * (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FMPRI 165</td>
<td>(AC) * E.O --&gt; (AC)</td>
</tr>
<tr>
<td>FMPRM 166</td>
<td>(AC) * (E) --&gt; (E)</td>
</tr>
<tr>
<td>FMPRB 167</td>
<td>(AC) * (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FDV 170</td>
<td>(AC) / (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FDVL 171</td>
<td>(AC) / (E) --&gt; (AC) REMAINDER --&gt; (AC+1)</td>
</tr>
<tr>
<td>FDVM 172</td>
<td>(AC) / (E) --&gt; (E)</td>
</tr>
<tr>
<td>FDVB 173</td>
<td>(AC) / (E) --&gt; (AC)(E)</td>
</tr>
<tr>
<td>FDVR 174</td>
<td>(AC) / (E) --&gt; (AC)</td>
</tr>
<tr>
<td>FDRV 175</td>
<td>(AC) / E.O --&gt; (AC)</td>
</tr>
<tr>
<td>FDRV 176</td>
<td>(AC) / (E) --&gt; (E)</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
PDP-10 Instruction Set

FDVRB 177 (AC) / (E) --> (AC)(E)
UFA 130 (AC) + (E) --> (AC+1) without normalization
DFN 131 -(AC,E) --> (AC,E)
FSC 132 (AC) * 2**E --> (AC)
GFSC 031 (AC,AC+1) * 2**E --> (AC,AC+1)
FLTR 127 (E) floated, rounded --> (AC)
GFLTR 030 (E) floated, rounded --> (AC,AC+1)
DGFLTR 027 (E,E+1) floated, rounded --> (AC,AC+1)
FIX 122 (E) fixed --> (AC)
FIXR 126 (E) fixed, rounded --> (AC)
GFIX 024 (E,E+1) fixed --> (AC)
GFXR 026 (E,E+1) fixed, rounded --> (AC)
GDFIX 023 (E,E+1) fixed --> (AC,AC+1)
GD FXR 025 (E,E+1) fixed, rounded --> (AC,AC+1)
GSNGL 021 (E,E+1) converted --> (AC)
GDBLE 022 (E) converted --> (AC,AC+1)

DFAD 110 (AC,AC+1) + (E,E+1) --> (AC,AC+1)
DFS B 111 (AC,AC+1) - (E,E+1) --> (AC,AC+1)
DFMP 112 (AC,AC+1) * (E,E+1) --> (AC,AC+1)
DFDV 113 (AC,AC+1) / (E,E+1) --> (AC,AC+1)

GFAD 102 (AC,AC+1) + (E,E+1) --> (AC,AC+1)
GF S B 103 (AC,AC+1) - (E,E+1) --> (AC,AC+1)
GFM P 106 (AC,AC+1) * (E,E+1) --> (AC,AC+1)
GFDV 107 (AC,AC+1) / (E,E+1) --> (AC,AC+1)

Fullword Instructions

EXCH 250 (AC) <-- (E)
MOVE 200 (E) --> (AC)
M O V E 201 O,E --> (AC)
MO V E M 202 (AC) --> (E)
MOVES 203 IF AC != O: (E) --> (AC)
MDVS 204 (E)S--> (AC)
MOVSI 205 E,O --> (AC)
MO V S M 206 (AC)S --> (E)
MO V S S 207 (E)S --> (E) IF AC != O: (E) --> (AC)
MDYN 210 -(E) --> (AC)
MOD V I 211 -[O,E] --> (AC)
MOVNM 212 -(AC) --> (E)
MOVNS 213 -(E) --> (E) IF AC != O: (E) --> (AC)
MOV M 214 [(E)] --> (AC)
MOVMI 215 O,E --> (AC)
MOV M M 216 [(AC)] --> (E)
MDVMS 217 [(E)] --> (E) IF AC != O: (E) --> (AC)
XM O V EI 415 E --> (AC) Non-local AC reference
1,E --> (AC) Local AC reference

210
# TOPS-20 Monitor Calls Quick Reference Guide

## PDP-10 Instruction Set

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Opcodes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMOVE</td>
<td>120</td>
<td>(E,E+1) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DMOVN</td>
<td>121</td>
<td>-(E,E+1) --&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DMOVEM</td>
<td>124</td>
<td>(AC,AC+1) --&gt; (E,E+1)</td>
</tr>
<tr>
<td>DMOVNM</td>
<td>125</td>
<td>-(AC,AC+1) --&gt; (E,E+1)</td>
</tr>
</tbody>
</table>

### BLT

251 Move E(R) - (AC)R + 1 words starting with:

\[ (AC)L \Rightarrow (AC)R \]

### XBLT

020 Move |(AC)| words

If (AC) > 0:
- start with \((AC+1) \Rightarrow (AC+2)\) and go up
If (AC) < 0:
- start with \((AC+1)-1 \Rightarrow (AC+2)-1\)
  and go down

## Halfword Instructions (Source Left)

### HLL 500

(E)L --> (AC)L

### HLLI 501

0 --> (AC)L

### HLLM 502

(AC)L --> (E)L

### HLLS 503

If AC ≠ 0: (E) --> (AC)

### HLLZ 510

(E)L,0 --> (AC)

### HLLZI 511

0 --> (AC)

### HLLZM 512

(AC)L,0 --> (E)

### HLLZS 513

0 --> (E)R

### HLLO 520

(E)L,777777 --> (AC)

### HLLOI 521

0,777777 --> (AC)

### HLLOM 522

(AC)L,777777 --> (E)

### HLLOS 523

777777 --> (E); If AC ≠ 0: (E) --> (AC)

### HLLL 590

(E)L,[(E)O * 777777] --> (AC)

### HLLI 551

0 --> (AC)

### HLLM 532

(AC)L,[(AC)O * 777777] --> (E)

### HLLS 533

(E)O * 777777 --> (E)R;

- If AC ≠ 0: (E) --> (AC)

### HLR 544

(E)L --> (AC)R

### HLRI 545

0 --> (AC)R

### HLRM 546

(AC)L --> (E)R

### HLR 547

(E)L --> (E)R; If AC ≠ 0: (E) --> (AC)

### HLRZ 554

0,(E)L --> (AC)

### HLRI 555

0 --> (AC)

### HLRZM 556

0,(AC)L --> (E)

### HLRS 557

0,(E)L --> (E); If AC ≠ 0: (E) --> (AC)

### HLRZ 564

777777,(E)L --> (AC)

### HLRZI 565

777777,0 --> (AC)

### HLROM 566

777777,(AC)L --> (E)

### HLRDS 567

777777,(E)L --> (E); If AC ≠ 0: (E) --> (AC)

### HLLRE 574

[(E)O * 777777],(E)L --> (AC)

### HLREI 575

0 --> (AC)

### HLLREM 576

[(AC)O * 777777],(AC)L --> (E)

### HLLRES 577

[(E)O * 777777],(E)L --> (E);

- If AC ≠ 0: (E) --> (AC)

### XHLLI 501

If zero section: E(L) --> (AC)L

If non-zero section:

211
1. O --> (AC bits 0-5)
2. section # --> (AC bits 6-17); if E is a local AC address, section # = 1

**Halfword Instructions (Source Right)**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRR</td>
<td>(E)R --&gt; (AC)R</td>
</tr>
<tr>
<td>HRRI</td>
<td>E --&gt; (AC)R</td>
</tr>
<tr>
<td>HRRM</td>
<td>(AC)R --&gt; (E)R</td>
</tr>
<tr>
<td>HRRS</td>
<td>If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HRRZ</td>
<td>O.(E)R --&gt; (AC)</td>
</tr>
<tr>
<td>HRRZI</td>
<td>0.E --&gt; (AC)</td>
</tr>
<tr>
<td>HRRZM</td>
<td>O.(AC)R --&gt; (E)</td>
</tr>
<tr>
<td>HRRZS</td>
<td>O --&gt; (E)L</td>
</tr>
<tr>
<td>HRRD</td>
<td>777777.(E)R --&gt; (AC)</td>
</tr>
<tr>
<td>HRRDI</td>
<td>777777.E --&gt; (AC)</td>
</tr>
<tr>
<td>HRRDM</td>
<td>777777.(AC)R --&gt; (E)</td>
</tr>
<tr>
<td>HRRDS</td>
<td>777777. --&gt; (E)L</td>
</tr>
<tr>
<td>HRRF</td>
<td>[(E)18 * 777777].(E)R --&gt; (AC)</td>
</tr>
<tr>
<td>HRRFI</td>
<td>[E18 * 777777].E --&gt; (AC)</td>
</tr>
<tr>
<td>HRRM</td>
<td>[(AC)18 * 777777].(AC)R --&gt; (E)</td>
</tr>
<tr>
<td>HRRMS</td>
<td>[(E)18 * 777777] --&gt; (E)L; If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HRL</td>
<td>(E)R --&gt; (AC)L</td>
</tr>
<tr>
<td>HRLI</td>
<td>E --&gt; (AC)L</td>
</tr>
<tr>
<td>HRLM</td>
<td>(AC)R --&gt; (E)L</td>
</tr>
<tr>
<td>HRLS</td>
<td>(E)R --&gt; (E)L; If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HRLZ</td>
<td>(E)R,O --&gt; (AC)</td>
</tr>
<tr>
<td>HRLI</td>
<td>E,O --&gt; (AC)</td>
</tr>
<tr>
<td>HRLZI</td>
<td>(AC)R,O --&gt; (E)</td>
</tr>
<tr>
<td>HRLZS</td>
<td>(E)R,O --&gt; (E); If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HRLD</td>
<td>(E)R,777777 --&gt; (AC)</td>
</tr>
<tr>
<td>HRLDI</td>
<td>E,777777 --&gt; (AC)</td>
</tr>
<tr>
<td>HRLM</td>
<td>(AC)R,777777 --&gt; (E)</td>
</tr>
<tr>
<td>HRLMS</td>
<td>(E)R,777777 --&gt; (E); If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HRLD</td>
<td>(E)R,[(E)18 * 777777] --&gt; (AC)</td>
</tr>
<tr>
<td>HRLDI</td>
<td>E,[E18 * 777777] --&gt; (AC)</td>
</tr>
<tr>
<td>HRLM</td>
<td>[(AC)R],[(AC)18 * 777777] --&gt; (E)</td>
</tr>
<tr>
<td>HRLMS</td>
<td>(E)R,[E18 * 777777] --&gt; (E); If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
</tbody>
</table>

**I/O Instructions**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAO</td>
<td>70014 (E) --&gt; DATA</td>
</tr>
<tr>
<td>DATAI</td>
<td>70004 DATA --&gt; (E)</td>
</tr>
<tr>
<td>CONO</td>
<td>70020 E --&gt; COMMAND</td>
</tr>
<tr>
<td>CONI</td>
<td>70024 STATUS --&gt; (E)</td>
</tr>
<tr>
<td>CONSZ</td>
<td>70030 If STATUS(R) &amp; E = 0: skip</td>
</tr>
<tr>
<td>CONSOD</td>
<td>70034 If STATUS(R) &amp; E ≠ 0: skip</td>
</tr>
<tr>
<td>BLKI</td>
<td>70000 (E) + 1,1 --&gt; (E); DATA --&gt; ((E)R); If (E)L ≠ O: skip</td>
</tr>
</tbody>
</table>

212
BLKD 70010 (E) + 1,1 --> (E); ((E)R) --> DATA;
     If (E)L ≠ 0: skip

Logical Testing Instructions

TLN 601 No-op
TLNE 603 If (AC)L & E = 0: skip
TLNA 605 Skip
TLNN 607 If (AC)L & E ≠ 0: skip
TLZ 621 (AC)L & ~E --> (AC)L
TLZE 623 If (AC)L & E = 0: skip; (AC)L & ~E --> (AC)L
TLZA 625 (AC)L & ~E --> (AC)L; skip
TLNZ 627 If (AC)L & E ≠ 0: skip; (AC)L & ~E --> (AC)L
TLC 641 (AC)L ^ E --> (AC)L
TLCE 643 If (AC)L & E = 0: skip; (AC)L ^ E --> (AC)L
TLCA 645 (AC)L ^ E --> (AC)L; skip
TLCN 647 If (AC)L & E ≠ 0: skip; (AC)L ^ E --> (AC)L
TLZ ! 661 (AC)L ! E --> (AC)L
TLZO 663 If (AC)L & E = 0: skip; (AC)L ! E --> (AC)L
TLIO 665 (AC)L ! E --> (AC)L; skip
TLON 667 If (AC)L & E ≠ 0: skip; (AC)L ! E --> (AC)L

TRN 600 No-op
TRNE 602 If (AC)R & E = 0: skip
TRNA 604 Skip
TRNN 606 If (AC)R & E ≠ 0: skip
TRZ 620 (AC)R & ~E --> (AC)R
TRZE 622 If (AC)R & E = 0: skip; (AC)R & ~E --> (AC)R
TRZA 624 (AC)R & ~E --> (AC)R; skip
TRZN 626 If (AC)R & E ≠ 0: skip; (AC)R & ~E --> (AC)R
TRC 640 (AC)R ^ E --> (AC)R
TRCE 642 If (AC)R & E = 0: skip; (AC)R ^ E --> (AC)R
TRCA 644 (AC)R ^ E --> (AC)R; skip
TRCN 646 If (AC)R & E ≠ 0: skip; (AC)R ^ E --> (AC)R
TRD 660 (AC)R ! E --> (AC)R
TRDE 662 If (AC)R & E = 0: skip; (AC)R ! E --> (AC)R
TRGA 664 (AC)R ! E --> (AC)R; skip
TRGN 666 If (AC)R & E ≠ 0: skip; (AC)R ! E --> (AC)R

TDN 610 No-op
TDNE 612 If (AC) & (E) = 0: skip
TDNA 614 Skip
TDNN 616 If (AC) & (E) ≠ 0: skip
TDZ 630 (AC) & ~(E) --> (AC)
TDZE 632 If (AC) & (E) = 0: skip;
     (AC) & ~(E) --> (AC)
TDZA 634 (AC) & ~(E) --> (AC); skip
TDZN 636 If (AC) & (E) ≠ 0: skip; (AC) & ~(E) --> (AC)
TDC 650 (AC) ^ (E) --> (AC)
TDCE 652 If (AC) & (E) = 0: skip; (AC) ^ (E) --> (AC)
TDCA 654 (AC) ^ (E) --> (AC); skip
TDIN 656 If (AC) & (E) ≠ 0: skip; (AC) ^ (E) --> (AC)
TDD 670 (AC) ! (E) --> (AC)
TDDE 672 If (AC) & (E) = 0: skip; (AC) ! (E) --> (AC)
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDGA</td>
<td>674</td>
<td>(AC) ! (E) --&gt; (AC); skip</td>
</tr>
<tr>
<td>TDON</td>
<td>676</td>
<td>If (AC) &amp; (E) ≠ 0: skip; (AC) ! (E) --&gt; (AC)</td>
</tr>
<tr>
<td>TSN</td>
<td>611</td>
<td>No-op</td>
</tr>
<tr>
<td>TSNE</td>
<td>613</td>
<td>If (AC) &amp; (E)S = 0: skip</td>
</tr>
<tr>
<td>TSNA</td>
<td>615</td>
<td>Skip</td>
</tr>
<tr>
<td>TSNN</td>
<td>617</td>
<td>If (AC) &amp; (E)S ≠ 0: skip</td>
</tr>
<tr>
<td>TSZ</td>
<td>631</td>
<td>(AC) &amp; ~(E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSZE</td>
<td>633</td>
<td>If (AC) &amp; (E)S = 0: skip; (AC) &amp; ~(E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSZA</td>
<td>635</td>
<td>(AC) &amp; ~(E)S --&gt; (AC); skip</td>
</tr>
<tr>
<td>TSNZ</td>
<td>637</td>
<td>If (AC) &amp; (E)S ≠ 0: skip; (AC) &amp; ~(E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSC</td>
<td>651</td>
<td>(AC) ^! (E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSCN</td>
<td>655</td>
<td>(AC) ^! (E)S --&gt; (AC); skip</td>
</tr>
<tr>
<td>TSCN</td>
<td>657</td>
<td>If (AC) &amp; (E)S ≠ 0: skip; (AC) ^! (E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSO</td>
<td>671</td>
<td>(AC) ! (E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSOE</td>
<td>673</td>
<td>If (AC) &amp; (E)S = 0: skip; (AC) ! (E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSOA</td>
<td>675</td>
<td>(AC) ! (E)S --&gt; (AC); skip</td>
</tr>
<tr>
<td>TSON</td>
<td>677</td>
<td>If (AC) &amp; (E)S ≠ 0: skip; (AC) ! (E)S --&gt; (AC)</td>
</tr>
</tbody>
</table>

---

**Program-control Instructions**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCT</td>
<td>256</td>
<td>Execute (E)</td>
</tr>
<tr>
<td>JFFD</td>
<td>243</td>
<td>If (AC) = 0: O --&gt; (AC+1)</td>
</tr>
<tr>
<td>JFCL</td>
<td>255</td>
<td>If AC &amp; FLAGS ≠ 0: E --&gt; (PC); ~AC &amp; FLAGS --&gt; FLAGS</td>
</tr>
<tr>
<td>JRST</td>
<td>25400</td>
<td>E --&gt; (PC)</td>
</tr>
<tr>
<td>PORTAL</td>
<td>25404</td>
<td>O --&gt; PUBLIC; E --&gt; (PC)</td>
</tr>
<tr>
<td>JRSTF</td>
<td>25410</td>
<td>(X)L or (Y)L --&gt; FLAGS; E --&gt; (PC)</td>
</tr>
<tr>
<td>HALT</td>
<td>25420</td>
<td>E --&gt; (PC); stop</td>
</tr>
<tr>
<td>XJRF</td>
<td>25424</td>
<td>(E)L --&gt; FLAGS; (E+1) --&gt; (PC)</td>
</tr>
<tr>
<td>XJEK</td>
<td>25430</td>
<td>Dismiss PI; (E)L --&gt; FLAGS; (E+1) --&gt; (PC)</td>
</tr>
<tr>
<td>XPCW</td>
<td>25434</td>
<td>FLAGS,O --&gt; (E); PC+1 --&gt; (E+1); (E+2)L --&gt; FLAGS; (E+3) --&gt; (PC)</td>
</tr>
<tr>
<td>JFEN</td>
<td>25450</td>
<td>Dismiss PI; (X)L or (Y)L --&gt; FLAGS; E --&gt; (PC)</td>
</tr>
<tr>
<td>SFM</td>
<td>25460</td>
<td>FLAGS,O --&gt; (E)</td>
</tr>
<tr>
<td>JSR</td>
<td>264</td>
<td>If PC(L) = 0: FLAGS,PC(R)+1 --&gt; (E); E+1 --&gt; (PC)</td>
</tr>
<tr>
<td>JSP</td>
<td>265</td>
<td>If PC(L) ≠ 0: PC+1 --&gt; (E); E+1 --&gt; (PC)</td>
</tr>
</tbody>
</table>

214
TOPS-20 Monitor Calls Quick Reference Guide
PDP-10 Instruction Set

If PC(L) # 0: PC+1 --> (AC); E --> (PC)

JSA 266  (AC) --> (E); E(R),PC(R)+1 --> (AC);
         E+1 --> (PC)

JRA 267  ((AC)L) --> (AC); E --> (PC)

MAP 257  PHYSICAL MAP DATA --> (AC)

Shift And Rotate Instructions

ASH 240  (AC) * 2**E --> (AC)

RDT 241  Rotate (AC) E places

LSH 242  Shift (AC) E places

ASHC 244  (AC,AC+1) * 2**E --> (AC,AC+1)

RDTC 245  Rotate (AC,AC+1) E places

LSHC 246  Shift (AC,AC+1) E places

Stack Instructions

PUSH 261  If PC(L) = 0 or (AC)0,6-17 <= 0:
         (AC) + 1,1 --> (AC); (E) --> ((AC)R)
         If PC(L) # 0 and (AC)0,6-17 > 0:
         (AC) + 1 --> (AC); (E) --> ((AC))

PDP 262  If PC(L) = 0 or (AC)0,6-17 <= 0:
         ((AC)R) --> (E); (AC) - 1,1 --> (AC)
         If PC(L) # 0 and (AC)0,6-17 > 0:
         ((AC)) --> (E); (AC) - 1 --> (AC)

PUSHJ 260  If PC(L) = 0: (AC) + 1,1 --> (AC);
         FLAGS,PC+1 --> ((AC)R)
         If PC(L) # 0 and (AC)0,6-17 <= 0:
         (AC) + 1,1 --> (AC); PC+1 --> ((AC)R)
         If PC(L) # 0 and (AC)0,6-17 > 0:
         (AC) + 1 --> (AC); PC+1 --> ((AC))
         E --> PC

PDPJ 263  If PC(L) = 0: ((AC)R)R --> (PC);
         (AC) - 1,1 --> (AC)
         If PC(L) # 0 and (AC)0,6-17 <= 0:
         ((AC)R) --> (PC); (AC) - 1,1 --> (AC)
         If PC(L) # 0 and (AC)0,6-17 > 0:
         ((AC)) --> (PC); (AC) - 1 --> (AC)

ADJSP 105  If PC(L) = 0 or (AC)0,6-17 <= 0:
         (AC) + [+1]E(R),E(R) --> (AC)
         If PC(L) # 0 and (AC)0,6-17 > 0:
         (AC) + [+1]E(R) --> (AC)

215
### MACRO-20 PSEUDO-OPS

<table>
<thead>
<tr>
<th>Pseudo-op/Arguments</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ARRAY addr[expr]</strong></td>
<td>Reserves a block of storage with length expr at address addr.</td>
</tr>
<tr>
<td><strong>ASCII dtextd</strong></td>
<td>Enters ASCII text; d is any delimiter not in text.</td>
</tr>
<tr>
<td><strong>ASCIZ dtextd</strong></td>
<td>Enters ASCII text with guaranteed trailing null; d is any delimiter not in text.</td>
</tr>
<tr>
<td><strong>.ASSIGN sym1,sym2,increment</strong></td>
<td>Assigns value of sym2 to sym1 and adds increment to sym2.</td>
</tr>
<tr>
<td><strong>ASUPPRESS</strong></td>
<td>Causes all local or INTERNAL symbols not referenced after ASUPPRESS to be deleted from symbol table.</td>
</tr>
<tr>
<td><strong>BLOCK expr</strong></td>
<td>Reserves a block of length expression.</td>
</tr>
<tr>
<td><strong>BYTE (n)expr</strong></td>
<td>Stores value of expression in n-byte bytes.</td>
</tr>
<tr>
<td><strong>COMMENT dtextd</strong></td>
<td>Treats text as comment; d is any delimiter not in text.</td>
</tr>
<tr>
<td><strong>.COMMON symbol[expr]</strong></td>
<td>Defines FORTRAN or FORTRAN-compatible COMMON block.</td>
</tr>
<tr>
<td><strong>.CREF</strong></td>
<td>Resumes output of suspended cross-referencing.</td>
</tr>
<tr>
<td><strong>DEC expr,....expr</strong></td>
<td>Defines local radix as decimal.</td>
</tr>
<tr>
<td><strong>DEFINE macro(dummyarg)&lt;macrotext&gt;</strong></td>
<td>Defines macro macro.</td>
</tr>
<tr>
<td><strong>DEPHASE</strong></td>
<td>Suspends effect of PHASE pseudo-op.</td>
</tr>
<tr>
<td><strong>.DIRECTIVE directive,....directive</strong></td>
<td>Sets switches to enable/disable MACRO features.</td>
</tr>
<tr>
<td><strong>.ITABM</strong></td>
<td>Include spaces and tabs in passed arguments.</td>
</tr>
<tr>
<td><strong>.XTABM</strong></td>
<td>Strip leading/trailing spaces and tabs from passed arguments.</td>
</tr>
<tr>
<td><strong>MACMPD</strong></td>
<td>Match paired delimiters in MACRO call.</td>
</tr>
<tr>
<td><strong>LITLST</strong></td>
<td>List binary code for in-line literals.</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
MACRO-20 Pseudo-Ops

FLBLST List binary code for 1st line of multi-line text

.LKDVL Allow arithmetic overflow

.ERDVL Disallow arithmetic overflow

MACRF Prefer MACRO symbol definition over other definitions

SFCOND Suppress source listing for failing conditional assembly

.NOBIN Suppress binary generation

KA10 Enter KA10 as CPU type in binary file header block

KI10 Enter KI10 as CPU type in binary file header block

KL10 Enter KL10 as CPU type in binary file header block

END expr

.ENDPS

ENTRY symbol,...,symbol

EXP expr,...,expr

EXTERN symbol,...,symbol

.HWFRT

.IF expr,qualifier,<code>

.IFN expr,qualifier,<code>

IFx expr,<code>

E

N

G

217
GE
L
LE
IF1 <code>
IF2 <code>
IFDEF symbol<code>
IFNDEF symbol<code>
IFIDN <string1><string2>,<code>
IFDIF <string1><string2><code>
IFB <string>,<code>
IFNB <string>,<code>
INTEGER symbol,...,symbol
INTERN symbol,...,symbol
IOWD expr1,expr2
IRP dummyarg,<code>
IRPC dummyarg,<code>
LALL
.LINK chainnum,addr,chainaddr
LIST
LIT
.LNKEND chainnum,addr
LOC expr
.MFRMT
MLOFF

Assemble if expr>=0
Assemble if expr<0
Assemble on Pass 1
Assemble on Pass 2
Assemble if symbol defined
Assemble if symbol not defined
Assemble if string1 matches string2
Assemble if string1 does not match string2
Assemble if blanks and tabs in string
Assemble if not only blanks and tabs in string
Reserves storage locations at end of program at one-per-symbol
Declares each symbol as INTERNAL to current program
Generates I/O transfer word such that word = \(2^{31} - \text{complement(}expr1\text{)}\)\(\times expr2 - 1\)
Generates expansion of code for each subargument of dummyarg
Generates expansion of code for each character of dummyarg
Lists all expansions (including text and macros) in program
Generates static chains at load time for links with same chainnum at addr and optionally places chain at chainaddr
Resumes listing following XLIST
Assembles literals beginning at current address
Ends a static chain for links with same chainnum at addr
Sets location counter to expr
Causes multi-format listing of binary code Terminates literals at EOL even if no closing bracket (})
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLON</td>
<td>Suspends MLOFF</td>
</tr>
<tr>
<td>.NODDT symbol,...,symbol</td>
<td>Suppresses DDT recognition of symbol</td>
</tr>
<tr>
<td>NOSYM</td>
<td>Suppress listing of symbol table in listing file</td>
</tr>
<tr>
<td>DCT expc,...,expr</td>
<td>Defines radix of expc as octal</td>
</tr>
<tr>
<td>OPDEF symbol[exp]</td>
<td>Defines symbol as equivalent to exp</td>
</tr>
<tr>
<td>.ORG addr</td>
<td>Sets location counter to addr</td>
</tr>
<tr>
<td>PAGE</td>
<td>Causes assembler to skip to top of next listing page</td>
</tr>
<tr>
<td>PASS2</td>
<td>Switches assembler to Pass 2 processing of remaining code</td>
</tr>
<tr>
<td>PHASE addr</td>
<td>Assembles part of program so that it can be moved to other location for execution</td>
</tr>
<tr>
<td>POINT bytesize,addr,bitplace</td>
<td>Generates a byte pointer for machine byte instructions</td>
</tr>
<tr>
<td>PRGEND</td>
<td>Replaces END for all but last program in multi-program assembly</td>
</tr>
<tr>
<td>PRINTX text</td>
<td>Causes text to be output during assembly to TTY and/or listing device</td>
</tr>
<tr>
<td>.PSECT name/attribute,origin</td>
<td>Specifies relocation counter for code following</td>
</tr>
<tr>
<td>PURGE symbol,...,symbol</td>
<td>Deletes symbol from symbol table</td>
</tr>
<tr>
<td>RADIX n</td>
<td>Sets radix to value of n</td>
</tr>
<tr>
<td>RADIX50 code, symbol</td>
<td>Packs symbol into B4-35 of storage symbol, with code in B0-3</td>
</tr>
<tr>
<td>RELOC expr</td>
<td>Sets location counter to value of expr and assigns relocatable addresses to code following</td>
</tr>
<tr>
<td>REMARK text</td>
<td>Marks text as comment</td>
</tr>
<tr>
<td>REPEAT n &lt;code&gt;</td>
<td>Generates code n times</td>
</tr>
<tr>
<td>.REQUEST filespec</td>
<td>Causes file filespec to be loaded to satisfy a global request</td>
</tr>
<tr>
<td>.REQUEST filespec</td>
<td>Causes file filespec to be loaded automatically</td>
</tr>
<tr>
<td>SALL</td>
<td>Causes suppression of all macro and REPEAT expansions</td>
</tr>
<tr>
<td>SEARCH tabnam(filespec)</td>
<td>Defines list of symbol tables to be searched</td>
</tr>
</tbody>
</table>
SIXBIT dtextd
Enters string text in 6-bit format; d is any delimiter not in text

SQUEZE code,symbol
Same as RADIX50

STOP
Ends IRP or IRPC before all subarguments or characters are used

SUBTTL subtitle
Defines subtitle (80 chars max) to be printed at top of each listing page

SUPPRESS symbol,...,symbol
Turns on suppress bit for symbol in symbol table; turned off when symbol referenced

SYN sym1,sym2
Defines sym2 as synonymous with sym1

TAPE
Causes assembler to begin assembling next source file in MACRO command string

.TEXT dtextd
Generates ASCIZ REL block for LINK and inserts text directly into .REL file; d is any delimiter not in text

TITLE title
Names program title and causes title to be printed on each page of listing

TWOSEG addr
Directs MACRO to assemble two-segment program with HISEG beginning at addr

UNIVERSAL tabnam
Declares symbol table of current program as available to other programs and stores tabnam in MACRO’s internal UNIVERSAL table

VAR
Causes variable symbols defined with symbol, ARRAY, or INTEGER to be assembled as BLOCK statements

XALL
Resumes standard listing after LALL or SALL

.XCREF symbol,...,symbol
Suspends output of cross-referencing for symbol

XLIST
Suspends output to program listing file for Pass 2 until next LIST

XPUNGE
Deletes all local symbols during Pass 2

XWD word1,word2
Enters low-order 16 bits of each word into a
single storage word; high-order bits are ignored
Generates instruction word with 0 in opcode field (B0-8), ac in accumulator field (B9-12), and addr in address field (B16-35)
NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

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

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

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

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

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

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

Name ___________________________________________ Date ______________________________

Organization _____________________________ Telephone _____________________________

Street _____________________________________________

City ___________________________________________ State _______ Zip Code ________
or Country