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 | ................................................................. | 203 |
| PDP-10 INSTRUCTION SET | ............................................................. | 205 |
| MACRO-20 PSEUDO-OPS | ............................................................. | 216 |
## CONVENTIONS

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_n$ or $B_n-m$</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 $1B_n$</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><code>&lt;value&gt;,&lt;value&gt;</code></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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GACCT</td>
<td>Reads a job’s account</td>
</tr>
<tr>
<td>GACTF</td>
<td>Reads a file’s account</td>
</tr>
<tr>
<td>LOGIN</td>
<td>Logs a job into the system</td>
</tr>
<tr>
<td>SACTF</td>
<td>Sets a file’s account</td>
</tr>
<tr>
<td>USAGE</td>
<td>Writes entries into the system’s accounting file</td>
</tr>
<tr>
<td>VACCT</td>
<td>Validates an account</td>
</tr>
</tbody>
</table>

### File Functions

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCES</td>
<td>Allows access to a directory</td>
</tr>
<tr>
<td>BKJFN</td>
<td>Backspaces file’s pointer</td>
</tr>
<tr>
<td>CHFDB</td>
<td>Changes a File Descriptor Block</td>
</tr>
<tr>
<td>CHKAC</td>
<td>Checks access to a file</td>
</tr>
<tr>
<td>CLOSF</td>
<td>Closes a file</td>
</tr>
<tr>
<td>CLZFF</td>
<td>Closes a process’ files</td>
</tr>
<tr>
<td>CRLNM</td>
<td>Creates a logical name</td>
</tr>
<tr>
<td>DELDF</td>
<td>Expunges deleted files</td>
</tr>
<tr>
<td>DELF</td>
<td>Deletes a file</td>
</tr>
<tr>
<td>DELNF</td>
<td>Retains specified number of generations of file</td>
</tr>
<tr>
<td>DIRST</td>
<td>Translates directory or user number to a string</td>
</tr>
<tr>
<td>FFFFP</td>
<td>Finds first free file page</td>
</tr>
<tr>
<td>FFUFD</td>
<td>Finds first used file page</td>
</tr>
<tr>
<td>GACTF</td>
<td>Reads a file’s account</td>
</tr>
<tr>
<td>GFUST</td>
<td>Reads the author or last writer name string</td>
</tr>
<tr>
<td>GNJFN</td>
<td>Assigns a JFN to the next file</td>
</tr>
<tr>
<td>GPJFN</td>
<td>Returns primary JFN’s</td>
</tr>
<tr>
<td>GTFDB</td>
<td>Reads a File Descriptor Block</td>
</tr>
<tr>
<td>GTJFN</td>
<td>Assigns a JFN to a file</td>
</tr>
<tr>
<td>GTSTS</td>
<td>Reads file’s status</td>
</tr>
<tr>
<td>INLNM</td>
<td>Writes logical names</td>
</tr>
<tr>
<td>JFNS</td>
<td>Translates a JFN to a string</td>
</tr>
<tr>
<td>LNMST</td>
<td>Translates logical name to string</td>
</tr>
<tr>
<td>OPENF</td>
<td>Opens a file</td>
</tr>
<tr>
<td>RCDIR</td>
<td>Translates directory name to number</td>
</tr>
<tr>
<td>RCUSR</td>
<td>Translates user name to number</td>
</tr>
<tr>
<td>RFBSZ</td>
<td>Reads file’s byte size</td>
</tr>
<tr>
<td>RFPTR</td>
<td>Reads file’s pointer</td>
</tr>
<tr>
<td>RFTAD</td>
<td>Reads file’s time and dates</td>
</tr>
<tr>
<td>RLJFN</td>
<td>Releases a JFN</td>
</tr>
<tr>
<td>RNAMF</td>
<td>Renames a file</td>
</tr>
<tr>
<td>SACTF</td>
<td>Sets a file’s account</td>
</tr>
<tr>
<td>SFBSZ</td>
<td>Sets file’s byte size</td>
</tr>
<tr>
<td>SFPTR</td>
<td>Sets file’s pointer</td>
</tr>
<tr>
<td>SFTAD</td>
<td>Sets file’s time and dates</td>
</tr>
<tr>
<td>SFUST</td>
<td>Changes the author or last writer name string</td>
</tr>
<tr>
<td>SIZEF</td>
<td>Obtains file’s length</td>
</tr>
<tr>
<td>SPJFN</td>
<td>Sets primary JFN’s</td>
</tr>
<tr>
<td>STSTS</td>
<td>Sets file’s status</td>
</tr>
<tr>
<td>SWJFN</td>
<td>Transposes two JFN’s</td>
</tr>
<tr>
<td>UFPGS</td>
<td>Updates file’s pages</td>
</tr>
</tbody>
</table>
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
ESOUT    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
RUNTM    Returns the runtime of a job or process
SYSGT    Returns values for a system table
TIME     Returns the time since the system was restarted
Device Control Functions

ASND Assigns a device
ATACH Attaches controlling terminal to a job
CFIBF Clears terminal’s input buffer
CFOBF Clears terminal’s output buffer
DEVST Translates a device designator to a string
DIBE Dismisses until terminal input buffer is empty
DOBE Dismisses until terminal output buffer is empty
DTACH Detaches controlling terminal from a job
DVCHR Returns device characteristics
GDSKC Returns disk usage
GDSTS Returns the device status
GTYP Returns terminal type number
LPINI Loads VFU or translation RAM
MSTR Performs structure-dependent functions
MTOPR Performs device-dependent functions
MTU% Performs functions for logical tape devices
RELD Releases a device
RFCCO Returns control character output control words
RFMOD Returns the JFN mode word
RFPOS Returns current position of the terminal
SDSTS Sets the device status
SFCCO Sets control character output control words
SFMOD Sets program-related fields in the JFN mode word
SFPOS Sets position of the terminal’s cursor
SIBE Skips if input buffer is empty
SOBE Skips if output buffer is empty
SOBF Skips if output buffer is full
SPOOL Defines and initializes input spooling
STDEV Translates a string to a device designator
STPAR Sets device-related fields in the JFN mode word
STYP Sets terminal type number
TLINK Controls terminal linking

Software Interrupt System Functions

AIC Activates interrupt channels
ATI Assigns terminal code to channel
CIS Clears the interrupt system
DEBRK Dismisses current interrupt
DIC Deactivates interrupt channels
DIR Disables the interrupt system
DTI Deassigns terminal code
EIR Enables the interrupt system
GTRPW Returns trap words
IIC Initiates interrupts on specific channels in a process
RCM Reads activated channel word mask
RIR Reads the interrupt table addresses for a single-section program
RIRCM Reads inferior reserved channel mask
RTIW Reads terminal interrupt word
RWM Reads waiting channel word mask
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCTTY</td>
<td>Changes source of terminal interrupts</td>
</tr>
<tr>
<td>SIR</td>
<td>Sets the interrupt table addresses for a single-section process</td>
</tr>
<tr>
<td>SIRCM</td>
<td>Sets inferior reserved channel mask</td>
</tr>
<tr>
<td>SKPISR</td>
<td>Skips if the interrupt system is enabled</td>
</tr>
<tr>
<td>STIW</td>
<td>Sets terminal interrupt word</td>
</tr>
<tr>
<td>XGTPW%</td>
<td>Returns page-fail words</td>
</tr>
<tr>
<td>XRIR%</td>
<td>Reads the interrupt table addresses for a multiple-section program</td>
</tr>
<tr>
<td>XSIR%</td>
<td>Sets the interrupt table addresses for a multiple-section process</td>
</tr>
</tbody>
</table>

**Process/Capability Handling Functions**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADBRK</td>
<td>Controls address breaks</td>
</tr>
<tr>
<td>CFORK</td>
<td>Creates inferior process</td>
</tr>
<tr>
<td>DISMS</td>
<td>Dismisses process for specified amount of time</td>
</tr>
<tr>
<td>EPCAP</td>
<td>Enables process capabilities word</td>
</tr>
<tr>
<td>FFORK</td>
<td>Freezes one or more processes</td>
</tr>
<tr>
<td>GFRKH</td>
<td>Gets process handle</td>
</tr>
<tr>
<td>GFRKS</td>
<td>Gets current process structure</td>
</tr>
<tr>
<td>HALTF</td>
<td>Halts a process</td>
</tr>
<tr>
<td>HFORK</td>
<td>Halts an inferior process</td>
</tr>
<tr>
<td>KFORK</td>
<td>Kills one or more processes</td>
</tr>
<tr>
<td>PRARG</td>
<td>Sets or returns process argument block</td>
</tr>
<tr>
<td>RESET</td>
<td>Resets and initializes current process</td>
</tr>
<tr>
<td>RFACS</td>
<td>Returns process' accumulators</td>
</tr>
<tr>
<td>RFORK</td>
<td>Resumes one or more processes</td>
</tr>
<tr>
<td>RFRKH</td>
<td>Releases process handles</td>
</tr>
<tr>
<td>RFSTS</td>
<td>Returns process' status</td>
</tr>
<tr>
<td>RMAP</td>
<td>Obtains a handle on a page in a process</td>
</tr>
<tr>
<td>RPACS</td>
<td>Returns accessibility of page</td>
</tr>
<tr>
<td>RPCAP</td>
<td>Returns process capabilities word</td>
</tr>
<tr>
<td>RSMAP%</td>
<td>Returns information about the mapping of one section of a process</td>
</tr>
<tr>
<td>RTFRK</td>
<td>Returns the handle of a process suspended because of a monitor call intercept</td>
</tr>
<tr>
<td>RWSET</td>
<td>Releases working set</td>
</tr>
<tr>
<td>SFACS</td>
<td>Sets process' accumulators</td>
</tr>
<tr>
<td>SFORK</td>
<td>Starts a process in section zero</td>
</tr>
<tr>
<td>SPACS</td>
<td>Sets accessibility of page</td>
</tr>
<tr>
<td>SPLFK</td>
<td>Splices a process structure</td>
</tr>
<tr>
<td>TFORK</td>
<td>Sets and removes monitor call intercepts</td>
</tr>
<tr>
<td>UFRK</td>
<td>Resumes a process suspended because of a monitor call intercept</td>
</tr>
<tr>
<td>WAIT</td>
<td>Dismisses process until interrupt occurs</td>
</tr>
<tr>
<td>WFORK</td>
<td>Waits for process to terminate</td>
</tr>
<tr>
<td>XSFRK%</td>
<td>Starts a process in a non-zero section</td>
</tr>
</tbody>
</table>
Save File Handling Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCVEC</td>
<td>Gets compatibility package entry vector</td>
</tr>
<tr>
<td>GDVEC</td>
<td>Gets RMS entry vector</td>
</tr>
<tr>
<td>GET</td>
<td>Obtains a saved file</td>
</tr>
<tr>
<td>GEVEC</td>
<td>Gets process entry vector of a single-section program</td>
</tr>
<tr>
<td>SAVE</td>
<td>Saves a process as nonsharable</td>
</tr>
<tr>
<td>SCVEC</td>
<td>Sets compatibility package entry vector</td>
</tr>
<tr>
<td>SDVEC</td>
<td>Sets RMS entry vector</td>
</tr>
<tr>
<td>SEVEC</td>
<td>Sets the entry vector for a single-section program</td>
</tr>
<tr>
<td>SFRKV</td>
<td>Starts process using its entry vector</td>
</tr>
<tr>
<td>SSAVE</td>
<td>Saves a process as sharable</td>
</tr>
<tr>
<td>XGVEC%</td>
<td>Gets process entry vector for a multiple-section program</td>
</tr>
<tr>
<td>XSFRK%</td>
<td>Starts a process using a user-supplied, global PC</td>
</tr>
<tr>
<td>XSVEC%</td>
<td>Sets the entry vector for a multiple-section program</td>
</tr>
</tbody>
</table>

Date/Time Conversion Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTAD</td>
<td>Gets current date and time in internal format</td>
</tr>
<tr>
<td>IDCNV</td>
<td>Converts from day, month, year to internal date and time</td>
</tr>
<tr>
<td>IDTIM</td>
<td>Inputs date and time, converting to internal format</td>
</tr>
<tr>
<td>IDTNC</td>
<td>Inputs date and time without converting to internal format</td>
</tr>
<tr>
<td>ODCNV</td>
<td>Converts from internal date and time to day, month, year</td>
</tr>
<tr>
<td>ODTIM</td>
<td>Outputs date and time, converting from internal format to text</td>
</tr>
<tr>
<td>ODTNC</td>
<td>Outputs date and time in internal format</td>
</tr>
</tbody>
</table>

Archive/Virtual Disk Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCF</td>
<td>Performs archive/virtual-disk operations</td>
</tr>
<tr>
<td>CRDIR</td>
<td>Creates or modifies a directory</td>
</tr>
<tr>
<td>DELDF</td>
<td>Expunges deleted files</td>
</tr>
<tr>
<td>DELNF</td>
<td>Retains specified number of generations of file</td>
</tr>
<tr>
<td>GTJFN</td>
<td>Assigns a JFN to a file</td>
</tr>
<tr>
<td>GNJFN</td>
<td>Assigns a JFN to the next file</td>
</tr>
<tr>
<td>JFNS</td>
<td>Translates a JFN to a string</td>
</tr>
<tr>
<td>OPENF</td>
<td>Opens a file</td>
</tr>
<tr>
<td>RFTAD</td>
<td>Reads file's time and dates</td>
</tr>
<tr>
<td>SETJB</td>
<td>Sets job parameters</td>
</tr>
<tr>
<td>SFTAD</td>
<td>Sets file's time and dates</td>
</tr>
<tr>
<td>SMON</td>
<td>Sets monitor flags</td>
</tr>
<tr>
<td>TMON</td>
<td>Reads monitor flags</td>
</tr>
</tbody>
</table>
Privileged Functions

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

ACCESS* Accesses a directory
ALLOC Allocates a device to a particular job
ARCF* Performs archive/virtual-disk operations
ASNSQ Assigns ARPANET special message queue
ATACH* Attaches job to new controlling terminal
BOOT Performs functions required for loading front-end software
CRDIR* Creates or modifies a directory
CROB* Creates a new job
DELDF* Expunges deleted files
DELF* Deletes files
DIAG Reserves and releases hardware channels
DSKAS Assigns specific disk addresses
DSKOP Allows hardware address specification in disk transfers
ENQ* Places a request in ENQ/DEQ resource queue
ENQC* Returns status of a resource
FLHST Flushes an ARPNET host
GACCT* Returns job account information
GIVOK% Allows/denies access to a protected system resource
GTDIR* Returns directory information
HALTF* Halts a process
HSYS 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
MSFRK Starts a process in monitor mode
MSTR* Performs structure-related functions
MTALN Associates magtape drive with logical unit number
MTOPR* Performs device-related functions
MTU% Performs MT-device functions
MUTIL* Performs IPCF functions
NODE* Performs DECnet functions
NTMAN%* Performs DECnet network management functions
PEEK Reads monitor data
PLOCK Locks physical pages
PMCTL Controls physical memory
RCVOK% Services GETOK% requests
SETJB* Sets job parameters
SFTAD* Sets file date/time
SFUST* Sets file author
SUPRI Sets job priority
SKED%* Manipulates scheduler data base
SMON Sets monitor flags
SNOOP Performs system performance analysis
SPOOL Performs spooling-related functions
SPRIW Sets process priority
STAD* Sets system date/time
STI* Simulates terminal input
SYERR Places information in the System Error file
TTMSG* Sends a message to a terminal
USAGE Makes entries in accounting file
USRI0 Places program in user I/O mode
UTEST Monitors executed instructions
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%CON) Connect job to directory
     B1(AC%OWN) Give job owner access to directory
     B2(AC%REM) Relinquish owner access to directory
     B18-35 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 .ACPSW Byte pointer to ASCIZ string containing
            password of specified directory
  2 .ACJOB Job # or -1 for current job (WHL/DPR if not
            -1)

ABBRK 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 (1Bn 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: JFN
AC2: Function code
AC3: Function-specific argument
**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>.ARRAR</td>
<td>Set/clear user request for archival&lt;br&gt;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&lt;br&gt;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)&lt;br&gt;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&lt;br&gt;AC3: 1BO(AR%NMS) Don’t send msg when restored&lt;br&gt;1B1(AR%WAT) Wait for file</td>
</tr>
<tr>
<td>4</td>
<td>.ARDIS</td>
<td>Discard tape information for file&lt;br&gt;AC3: 1BO(AR%CR1) Clear run 1 information&lt;br&gt;1B1(AR%CR2) Clear run 2 information</td>
</tr>
<tr>
<td>5</td>
<td>.ARSST</td>
<td>Set tape information for file; (enabled WHL/OPR)&lt;br&gt;AC3: Pointer to argblk</td>
</tr>
<tr>
<td>6</td>
<td>.ARRST</td>
<td>Restore file to disk; (enabled WHL/OPR)&lt;br&gt;AC3: JFN for a DUMPER temporary file</td>
</tr>
<tr>
<td>7</td>
<td>.ARGST</td>
<td>Get tape information for file&lt;br&gt;AC3: Pointer to argblk</td>
</tr>
<tr>
<td>10</td>
<td>.ARRFL</td>
<td>Retrieve for file failed; (WHL/OPR)</td>
</tr>
<tr>
<td>11</td>
<td>.ARNAR</td>
<td>Set/clear resist involuntary migration&lt;br&gt;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>.ARFL</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%OFL) 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 saveset 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 saveset 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

ASNSQ  _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(AT%CcJ)  Generate a CTRL/C interrupt to attached job
        B1(AT%NAT)  Do not attach job
        B2(AT%TRM)  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
TOPS-20 Monitor Calls Quick Reference Guide

ATI

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(AT%NTP)  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
terminal
designator in AC1

BIN  JSYS 50

FUNCTION
Inputs the next byte from the specified source.

CALLING SEQUENCE
AC1:  Source designator

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

BKJFN  JSYS 42

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

RESTRICTIONS
Cannot be used with DECNENET devices SRV: or DCN::

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

BOOT  JSYS 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>.BTROM</td>
<td>Put line in MOP mode; activate front end bootstrap ROM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line number</td>
</tr>
<tr>
<td>1</td>
<td>.BTLDS</td>
<td>Load secondary bootstrap program into front end</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Address of bootstrap program</td>
</tr>
<tr>
<td>2</td>
<td>.BTLOD</td>
<td>Load front-end memory using previously loaded secondary or tertiary bootstrap program</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of bytes to transfer</td>
</tr>
<tr>
<td></td>
<td></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></td>
<td></td>
<td>Line number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td></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></td>
<td></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 DECENT node for this front end</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line number</td>
</tr>
<tr>
<td></td>
<td></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>
TOPS-20 Monitor Calls Quick Reference Guide
BOOT

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(BT%CVE) CRAM verify error
    B1(BT%DVE) DRAM verify error
    B2(BT%RVE) 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 1BO, B18-35 contain start address
    BO(BT%KSA) 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 .BTTCC # 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 .BTCPN 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 .BTSDD 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:
       1B0(BT%CTL) +
       .BTSUP (1) - station came up
       .BTSDW (2) - station went down
       .BTCP (3) - transmit complete
       .BTSSF (4) - start-up failed

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

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

KL-10 Processor Functions

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Argblk</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.BTROM</td>
<td>Put line in MOP mode; activate front end bootstrap ROM</td>
</tr>
<tr>
<td></td>
<td>.BTDTE</td>
<td>DTE-20 number</td>
</tr>
<tr>
<td>1</td>
<td>.BTERR</td>
<td>Error flags on failure (RET)</td>
</tr>
<tr>
<td></td>
<td>.BTLDS</td>
<td>Load secondary bootstrap program into front end</td>
</tr>
<tr>
<td></td>
<td>.BTDTE</td>
<td>DTE-20 number</td>
</tr>
<tr>
<td></td>
<td>.BTERR</td>
<td>Error flags on failure (RET)</td>
</tr>
<tr>
<td>2</td>
<td>.BTSEC</td>
<td>Address of bootstrap to load</td>
</tr>
<tr>
<td>3</td>
<td>.BTLOD</td>
<td>Load front-end memory using previously loaded secondary or tertiary bootstrap program</td>
</tr>
<tr>
<td></td>
<td>.BTDTE</td>
<td>DTE-20 number</td>
</tr>
<tr>
<td></td>
<td>.BTERR</td>
<td>Error flags on failure (RET)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not used; must be zero</td>
</tr>
<tr>
<td></td>
<td>.BTFLG</td>
<td>User-supplied flag word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO(BT%BEL) Send to -11 doorbell</td>
</tr>
</tbody>
</table>
after setup

3 .BTDMP Dump front-end memory using ROM bootstrap program
  0 .BTDTE DTE-20 number
  1 .BTERR Error flags on failure (RET)
  2 Not used; must be zero
  3 Not used; must be zero
  4 .BTCNT Number of bytes to transfer
  5 .BTDPT Pointer to dump data destination

4 .BITPR Initialize front end protocol
  0 .BTDTE DTE-20 number
  1 .BTPRV Protocol version number to use

5 .BTTPR Stop protocol currently running on front end or line
  0 .BTDTE DTE-20 number

6 .BTSTS Return status type of protocol running on front end to specified DTE or line, and name of adjacent DECNET node for front end
  0 .BTDTE DTE-20 number
  1 .BTCOD Returned protocol version type; or -1 if no protocol is running

7 .BTBEL Block until signal to TOPS-20 is initiated by front end
  0 .BTDTE DTE-20 number

10 .BTRMP Read data from front end using previously loaded secondary or tertiary bootstrap program
  0 .BTDTE DTE-20 number
  1 .BTERR Error flags on failure (RET)
  2 Not used; must be zero
  3 .BTFLG User-supplied flag word
     BO(BT%BEL) Send doorbell after transfer
  4 .BTCNT Maximum # of bytes to transfer
  5 .BTPMT Pointer to data destination

14 .BTCLI Convert line ID to port number
  1 .BTLID Pointer to ASCIZ line ID

15 .BTCPN Convert NSP port number to line ID
  1 .BTLID Pointer to ASCIZ line ID

16 .BT6DO Send message to or receive message from DN60 front end using .VND60 protocol (Requires DN60 on KL-10 Model B)
  0 .BT6DTE DTE number
  1 .BT6ERR Error flags (RET)

30 D6%BDP Byte pointer is bad
31 D6%ARD -11 attempted to send data
32 D6%TRS DTESRV timed out waiting for response header from -11
33 D6%TDT DTESRV timed out waiting for data from -11
34 D6%TP0 DTESRV timed out
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, OWGBPs must specify 7-bit bytes.

CALLING SEQUENCE
AC1: Byte pointer to the new account string;
in section 0, may contain <5B2+<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
TOPS-20 Monitor Calls Quick Reference Guide
CFIBF

RETURNS +1: Always

CFOBF 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: B0(CR%MAP) Make inferior process' map same as current process' map
     B1(CR%CAP) Make inferior process' capabilities same as current process'
     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 B18-35 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: B0(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
     B18-35(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
Word  Symbol  Meaning
0     .CKAAC  Code of desired access to files
1     .CKALD  Byte pointer to username string, or 36-bit user number
2     .CKACD  Byte pointer to directory name string, or 36-bit directory number of user's connected directory
3     .CKaec  Enabled capabilities of user
4     .CKAUD  Byte pointer to directory name string, or 36-bit directory number of directory being accessed; if BO(CK%JFN) is on, contains JFN for file being accessed
5     .CKAPR  Protection of files being accessed; (not required if a JFN is supplied in word .CKAUD)

ACCESS CODES
Code  Symbol  Meaning
0     .CKARD  Read existing files
1     .CKAWR  Write existing files
2     .CKaEX  Execute existing files
3     .CKAAP  Append to existing files
4     .CKADL  Obtain directory listing of existing files
6     .CKADR  Read the directory
10    .CKACN  Connect to the directory
11    .CKACF  Create files in the directory
TOPS-20 Monitor Calls Quick Reference Guide
CIS

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(C0%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%NOP 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 .CMIDJ <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 COMND
10 .CMABC Size of atom buffer in bytes
11 .CMGJB Address of 16 word, writable GTJFN argbik

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
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 .CBMRK Address of 4-word break mask that specifies which characters terminate a field

FUNCTIONS FOR WORD .CMFNOP 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>.CMFIL</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>.CMCMN</td>
<td>Parse a comma</td>
</tr>
<tr>
<td>14</td>
<td>.CMINT</td>
<td>Initialize the command line</td>
</tr>
<tr>
<td>15</td>
<td>.CMHFT</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>.CMTAD</td>
<td>Parse a date and/or time field according to setting of bits CM%IDA and CM%TIM</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>.CMOKT</td>
<td>Parse input and compare it with a given string</td>
</tr>
<tr>
<td>24</td>
<td>.CMNIX</td>
<td>Parse a number and terminate on 1st nonnumeric character</td>
</tr>
<tr>
<td>25</td>
<td>.CMACT</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 (CM%FFL) of Word .CMFNOP

<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%PO</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>CM%SDH</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

**Function**

- **.CMKEY**
  - Contents of Word .CMDAT
  - Address of keyword symbol table whose entries point to argblks; B18-35 of Word O of argblk may contain flags:
    - B33(CM%ABR) Keyword is abbreviation
    - B34(CM%NOR) Do not recognize keyword
    - B35(CM%INV) Make keyword invisible

- **.CMNUM**
  - Radix of the number (from 2 to 10)

- **.CMNOI**
  - Byte pointer to an ASCIZ string that contains the guide word

- **.CMSWI**
  - Address of switch keyword table, whose entries point to argblks; B18-35 of Word O of argblk may contain flags:
    - B33(CM%ABR) Keyword is abbreviation
    - B34(CM%NOR) Do not recognize keyword
    - B35(CM%INV) Make keyword invisible

- **.CMDIR**
  - Data bits
    - BO(CM%DWC) Allow wildcard characters in directory name

- **.CMTAD**
  - `<flag bits>,<address of 3-word block>`
    - BO(CM%IDA) Parse a date
    - B1(CM%ITM) Parse a time
    - B2(CM%NCI) Do not convert date/time to internal format

- **.CMUQS**
  - Address of 4-word block of 128. break character mask bits

- **.CMTOK**
  - Byte pointer to the given string

- **.CMNUX**
  - The radix (from 2 to 10) of the number

**DEFAULT HELP MESSAGES**

**Function**

- **.CMKEY (keyword)**
  - Message
    - ONE OF THE FOLLOWING
    - if no keyword matches the currently typed field
    - KEYWORD (NO DEFINED KEYWORDS MATCH THIS INPUT)

- **.CMNUM (number)**
  - OCTAL NUMBER (radix 8)
  - DECIMAL NUMBER (radix 10)
  - A NUMBER IN BASE nn (radix nn)

- **.CMNOI (guide word)**
  - None

- **.CMSWI (switch)**
  - ONE OF THE FOLLOWING

- **.CMOFI (input file)**
  - Depending on flag settings for
  - GTJFN call, OUTPUT FILESPEC or
  - INPUT FILESPEC

- **.CMFIL (any file)**
  - None

- **.CMFLD (any field)**
  - CONFIRM WITH CARRIAGE RETURN

- **.CMDIR (directory)**
  - DIRECTORY NAME

- **.CMUSR (user)**
  - USER NAME

- **.CMCMA (comma)**
  - COMMA

- **.CMIY (initialize)**
  - None

- **.CMFLT (floating point)**
  - NUMBER

- **.CMDEV (device)**
  - DEVICE NAME

- **.CMTXT (text)**
  - TEXT STRING

- **.CMTAD (date)**
  - Depending on bits set in .CMDAT,
DATE, TIME, or DATE AND TIME

CMQST (quoted)  QUOTED STRING
CMUQS (unquoted) UNQUOTED STRING if "?" is a break character
CMTOK (token) None
CMNUX (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
EDLBO. - EOLB3. .CMTXT Yes
KEYBO. - KEYB3. .CMSWI Yes
User-specified .CMDAT Yes
USRBO. - USRB3. .CMUSR No
FILBO. - FILB3. .CMFIL No
FILBO. - FILB3. .CMIFI No
FILBO. - FILB3. .CMOFI No
internal .CMNUM No
FILBO. - FILB3. .CMDIR No
internal .CMFLT No
ACTBO. - ACTB3. .CMACT No

RETURNED BITS IN WORD .CMFLG OF THE FUNCTION DESCRIPTOR BLOCK

Bit Symbol Meaning
0 CM%ESC ESC was typed by user as terminator for this field
1 CM%NOP Field could not be parsed because it did not conform to specified function(s)
2 CM%EQC Field was terminated with a carriage return
3 CM%RPT Characters already parsed need to be reparsed because user edited them
4 CM%SWT Switch field was terminated with a colon
5 CM%PFE Previous field was terminated with an ESC

CRDIR JSYS 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
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%DPT) 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%SDQ) 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%DE) 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

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.CDLEN</td>
<td>&lt;flag bits&gt;,&lt;length of argblk&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0(CD%NSQ) On restore, do not update superior directory's quotas (enabled WHL/OPR required)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1 (CD%CNE) On restore or reconstruction, do not change directory parameters if directory currently exists (enabled WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(CD%NED) Set default on-line expiration date from word .CDDNE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B3(CD%FED) Set default on-line expiration date from word .CDDDNE</td>
</tr>
<tr>
<td>1</td>
<td>.CDPSW</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>.CDMOD</td>
<td>Mode word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0(CD%DIR) Directory is files-only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(CD%ANA) Obsolete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B2(CD%RLM) Repeat messages from file &lt;SYSTEM&gt;MAIL.TXT each time user logs in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B7(CD%DAR) File should be archived rather than migrated when on-line expiration date reached</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 (valid only when creating a directory)</td>
</tr>
<tr>
<td>7</td>
<td>.CDFPT</td>
<td>Default file protection (18 bits, R-U)</td>
</tr>
<tr>
<td>10</td>
<td>.CDPT</td>
<td>Directory protection (18 bits, R-U)</td>
</tr>
<tr>
<td>11</td>
<td>.CDRET</td>
<td>Default generation retention count</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide

CRDIR

12 .CDLLD  Date of last login
13 .CDUGP  Address of user group list for this directory
14 .CDGDGP  Address of directory group list
15 .CSDQ  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%LIQ  250 working pages
  3  CD%PRV  No special capabilities
  4  CD%MOD  Directory name for login
  5  CD%LOQ  250 permanent pages
  6  CD%NUM  First unused directory number
  7  CD%FPT  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%CU  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, OWGBPs 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
       .CJUCA  Use current account of caller
       .CJUAA  Use account from the argblk
       .CJUDA  Use default account of caller
4 **CJ%ETF** Place TOPS-20 command processor in top process of new job
5 **CJ%FIL** Move the file pointed to by word .CUFIL of the argblk into a process in new job
6 **CJ%ACS** Load ACs from the address in argument block; loaded only if the program being run is not the command processor
7 **CJ%OWN** Maintain ownership of the new job
8 **CJ%WTA** Do not start new job until it is attached to a terminal
9 **CJ%NPW** Do not check password given when new job is logged in
10 **CJ%NUD** Do not update LOGIN date for user logging into new job
11 **CJ%SPJ** Set primary I/O designators from argblk before starting job
12 **CJ%CAP** Set allowed capabilities of new job to be same as caller's currently enabled capabilities, until new job is logged in
13 **CJ%CAM** Set allowed capabilities of new job to combination and function of capability mask in argblk and new job's user capabilities
14 **CJ%SLO** Send IPCF message to PID supplied in argblk when new job is logged out
17 **CJ%DSN** Release ownership of previously created job whose number is in AC3; if on, overrides all other bits set in AC1

### ARGUMENT BLOCK

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.CUNAM</td>
<td>Byte pointer to ASCIZ user name string</td>
</tr>
<tr>
<td>1</td>
<td>.CUPSW</td>
<td>Byte pointer to ASCIZ password string</td>
</tr>
<tr>
<td>2</td>
<td>.CJACT</td>
<td>5B2 + account number or byte pointer to account string</td>
</tr>
<tr>
<td>3</td>
<td>.CUFIL</td>
<td>Byte pointer to name of file to be moved into a process of new job</td>
</tr>
<tr>
<td>4</td>
<td>.CJSFW</td>
<td>Offset in entry vector to use as start address of the file to which word .CUFIL points</td>
</tr>
<tr>
<td>5</td>
<td>.CJTTY</td>
<td>TTY designator of new job's controlling terminal</td>
</tr>
<tr>
<td>6</td>
<td>.CUTIM</td>
<td>Connect-time for new job before LGOUT is forced on it; 0 indicates no limit</td>
</tr>
<tr>
<td>7</td>
<td>.CJACS</td>
<td>Address of 16-word block to be loaded in new job's ACs if program other than Command Processor is being run</td>
</tr>
<tr>
<td>10</td>
<td>.CJEXF</td>
<td>Flag bits to be passed to Command Processor in top-level process of new job</td>
</tr>
</tbody>
</table>

**B0** Suppress herald printed by Command Processor

**B1** Move file to which word .CUFIL points into process whose handle is in PRARG block

**B2** Start process at offset in entry vector given in word .CJSFW after Command

27
Processor is initialized

B3 Output text printed when LOGIN command is given

11 .CJPRI Primary input and output device designators for the inferior processes of the new job
12 .CFCPU Run-time limit for new job
13 .CJCAM Capability mask for new job; used only if CJCAM is set
14 .CSLO PID to which IPCF message is to be sent when new job is logged out

Format of IPCF Logout Message
Word Contents
0 0,,.,IPCLD
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%ELO Job executed LGOUT
   3 SP%LO Job was forced to logout
   4 SP%OLO 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
Code Symbol Meaning
0 .CLNJ Delete one logical name from the job
1 .CLNS Delete one logical name from the system
2 .CLNJU Delete all logical names from the job
3 .CLNSA Delete all logical names from the system
4 .CLNJB Create a logical name for the job
5 .CLNSY Create a logical name for the system
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

DELDI  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(DD%DTF)  Delete temporary files (;T) also
B1(DD%DNF)  Delete nonexistent files that are not now open
B2(DD%RST)  Rebuild the symbol table
B3(DD%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.

REQUIREMENTS
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%CNO)  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 JSYS 514

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

RESTRICTIONS
In non-zero sections, DWGBPs 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

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.DEQDR</td>
<td>Remove specified requests from queue; requires argblk</td>
</tr>
<tr>
<td>1</td>
<td>.DEQDA</td>
<td>Remove all requests for this process from queue</td>
</tr>
<tr>
<td>2</td>
<td>.DEQID</td>
<td>Remove all requests corresponding to specified request ID</td>
</tr>
</tbody>
</table>

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 Header length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B6-17 # of locks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Length of argblk</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>
</tbody>
</table>

B0(EN%SHR) 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(EN%LTL) 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; 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/OPR processes

3 .ENQC Pointer to string; or 582+33-bit user code
4 .ENQRS <# of resources in pool>,<# requested>; or 0,<group number> if only one resource of specific type
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>
n-1 Address of a resource mask block

DEVST JSYS 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 JSYS 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

DFDUT JSYS 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 AC4 and updated string pointer in AC1, if pertinent.
     +2: Success, updated byte pointer in AC1, if pertinent

DIAG USYS 530

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>
</tbody>
</table>
100   | .DGGEM | Get memory (for TGHA) |
|      | 0      | Function code |
|      | 1      | 1st page in user address space |
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  
address  device  O  unit  subunit 

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 (1Bn means deactivate channel n)  

RETURNS +1: Always
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

DOBE  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) Deassign specified address
B1(DA%ASF) Assign a free page near specified address
B2(DA%CNV) Convert specified address according to setting of B3(DA%HWA)
B3(DA%HWA) Specified address is a hardware address
B4(DA%INI) Initialize a private copy of bit table
B5(DA%WRT) Write private copy of bit table to new bit table file
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) Field indicating the address type
1(DOPPU) for physical channel/unit addresses, with
B2-6(DOP%CN) channel number
B7-12(DOP%UN) unit number
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%EO) Error if unit off-line
B11(DOP%IL) Inhibit error logging

36
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 DOP%SN; physical channel, controller, and unit numbers if 1B9(DOP%NF) with
BO-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 65

FUNCTION
Reads data words into memory in unbuffered data mode.

RESTRICTIONS
File must be open for data mode 17

CALLING SEQUENCE
AC1:  JFN
AC2:  BO(DM%NWT)  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 O,v  Causes next command to be taken from location v
O  Terminates the command list

DUMPO  JSYS 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%NWT)  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 O,v  Causes next command to be taken from location v
O  Terminates command list

DVCHR  JSYS 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>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DV%OUT</td>
<td>Device can do output</td>
</tr>
<tr>
<td>1</td>
<td>DV%IN</td>
<td>Device can do input</td>
</tr>
<tr>
<td>2</td>
<td>DV%DIR</td>
<td>Device has a directory</td>
</tr>
<tr>
<td>3</td>
<td>DV%AS</td>
<td>Device is assignable with ASND</td>
</tr>
<tr>
<td>4</td>
<td>DV%MDD</td>
<td>Device has multiple directories</td>
</tr>
<tr>
<td>5</td>
<td>DV%AV</td>
<td>Device is available or assigned to this job</td>
</tr>
<tr>
<td>6</td>
<td>DV%ASN</td>
<td>Device is assigned by ASND</td>
</tr>
<tr>
<td>8</td>
<td>DV%MNT</td>
<td>Device is mounted</td>
</tr>
<tr>
<td>9-17</td>
<td>DV%TYP</td>
<td>Device type</td>
</tr>
<tr>
<td>0</td>
<td>.DVDSK</td>
<td>Disk</td>
</tr>
<tr>
<td>2</td>
<td>.DVMTA</td>
<td>Magtape</td>
</tr>
<tr>
<td>7</td>
<td>.DVLP</td>
<td>Line printer</td>
</tr>
<tr>
<td>10</td>
<td>.DVCDR</td>
<td>Card reader</td>
</tr>
<tr>
<td>11</td>
<td>.DVFE</td>
<td>Front-end pseudo-device</td>
</tr>
<tr>
<td>12</td>
<td>.DVTY</td>
<td>Terminal</td>
</tr>
<tr>
<td>13</td>
<td>.DVPTY</td>
<td>Pseudo-terminal</td>
</tr>
<tr>
<td>15</td>
<td>.DVNU</td>
<td>Null device</td>
</tr>
<tr>
<td>16</td>
<td>.DVNET</td>
<td>ARPA network</td>
</tr>
<tr>
<td>22</td>
<td>.DVDCN</td>
<td>DECnet active component</td>
</tr>
<tr>
<td>23</td>
<td>.DVSRV</td>
<td>DECnet passive component</td>
</tr>
<tr>
<td>20-35</td>
<td>DV%MOD</td>
<td>Data mode in which device can be opened</td>
</tr>
<tr>
<td>B20</td>
<td>DV%M17</td>
<td>Dump mode</td>
</tr>
<tr>
<td>B27</td>
<td>DV%M10</td>
<td>Image mode</td>
</tr>
<tr>
<td>B34</td>
<td>DV%M1</td>
<td>Small buffer mode</td>
</tr>
<tr>
<td>B35</td>
<td>DV%M0</td>
<td>Normal mode</td>
</tr>
</tbody>
</table>

### EIR JSYS 126

**FUNCTION**

Enables the software interrupt system for a process.

**CALLING SEQUENCE**

AC1: Process handle

**RETURNS** +1: Always

### ENQ JSYS 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
     Success

FUNCTION CODES
Code Symbol Meaning
0 .ENQBL Queue requests and block process until all requested locks are acquired
1 .ENQAA 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 B0-5 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(EN%SHR) 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(EN%LTL) Allow a long-term lock on this resource
           B9-17(EN%LVL) Level number associated with this resource
18-35 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 5B2+33-bit user code
           <# of resources in pool>,<# requested>
           or 0,<group number> if only one resource of type exists
4 .ENQRS <flags & level number>,<JFN>[-1][-2][-3]
           Pointer to string or 5B2+33-bit user code
           <# of resources in pool>,<# requested>
           or 0,<group number>
5 .ENQMS Address of a resource mask block
n-4 Address of a resource mask block
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 (.ENOCS 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>Resource status flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Error has occurred in corresponding resource request; B18-35 contain error code</td>
</tr>
<tr>
<td>B1(EN%QCE)</td>
<td>Process owns the lock</td>
</tr>
<tr>
<td>B2(EN%QCQ)</td>
<td>Process is in queue waiting for this resource</td>
</tr>
<tr>
<td>B3(EN%QCX)</td>
<td>Lock has been allocated for exclusive access</td>
</tr>
<tr>
<td>B4(EN%QCB)</td>
<td>Process is in queue waiting for exclusive access to resource</td>
</tr>
<tr>
<td>B9-17(EN%LVL)</td>
<td>Level number of the resource</td>
</tr>
<tr>
<td>B18-35(EN%JOB)</td>
<td>Job # of lock owner</td>
</tr>
<tr>
<td>1</td>
<td>36-bit time stamp</td>
</tr>
<tr>
<td>2</td>
<td>&lt;# of processes with lock&gt;,&lt;request ID&gt;</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 JSYS 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>0/2</td>
<td>Length of block</td>
</tr>
<tr>
<td></td>
<td>n</td>
<td>Returned data</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
ENQC

Data Returned by Function .ENQCD

Lock Data

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.ENQDF</td>
<td>BO-8  Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B9-17 Level number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 OFN, 40000+job #, -2, or -3</td>
</tr>
<tr>
<td>1</td>
<td>.ENQDR</td>
<td>&lt;total resources in pool&gt;, &lt;# remaining&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or 0, &lt;group number&gt;</td>
</tr>
<tr>
<td>2</td>
<td>.ENQDT</td>
<td>Time stamp of last request locked</td>
</tr>
<tr>
<td>3</td>
<td>.ENQDC</td>
<td>User code of lock or beginning of string</td>
</tr>
</tbody>
</table>

Queue Data

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.ENQDF</td>
<td>BO-8  Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B9-17 PSI channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 job # queue entry creator</td>
</tr>
<tr>
<td>1</td>
<td>.ENQDI</td>
<td>&lt;group # or number requested&gt;, &lt;request ID&gt;</td>
</tr>
</tbody>
</table>

Flags Returned in Word 0 for Function .ENQCD

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>EN%QCL</td>
<td>Block contains lock data (if off, queue data)</td>
</tr>
<tr>
<td>1</td>
<td>EN%QCD</td>
<td>Process owns the lock</td>
</tr>
<tr>
<td>2</td>
<td>EN%QCT</td>
<td>Lock contains a text string</td>
</tr>
<tr>
<td>3</td>
<td>EN%QCX</td>
<td>Lock is for exclusive access</td>
</tr>
<tr>
<td>4</td>
<td>EN%QCX</td>
<td>Process is blocked until exclusive access is available</td>
</tr>
</tbody>
</table>

EPCAP  JSYS 15:

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  JSYS 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

FFFFP _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

FFUFP _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
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

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

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
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, 5B2+account number returned in AC2

GCVEC  JSYS 300

FUNCTION
Returns the entry vector and the UUD 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: <UUD location>,,<PC location>
GDSKC

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

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 BO-3
  AC3: Foreign host number (octal)
  AC4: Foreign socket number (octal)

GDVEC

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: BO-17 Entry vector length
     B18-35 Entry vector address
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%NOV) Do not overlay existing mapped pages; return error
      B22(GT%ARG) 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

Word | Symbol | Meaning
-----|--------|--------------------------------------------------
0 .GFLAG | Flags for remainder of argblk
0 | GT%LDW | .GLOW contains lowest page number within process to use
1 | GT%HGH | .GHIGH contains highest page number within process to use
2 | GT%BAS | .GBASE contains the section number to use
3 | GT%CCH | Clear system's program cache (WHL/OPR)
4 | GT%CSH | Place in cache the program name being loaded into memory (WHL/OPR)
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).
CALLING SEQUENCE
AC1:  <index into table>,<table number>

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

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>

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>.JIUNO</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>.JIUNO</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>.JSNM</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>.JICPJ</td>
<td>Controlling PTY job #; -1 if no PTY</td>
</tr>
<tr>
<td>10</td>
<td>.JIRTL</td>
<td>Runtime limit; -1 if no time limit</td>
</tr>
<tr>
<td>11</td>
<td>.JIBAT</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>.JIPAR</td>
<td>Default magtape parity</td>
</tr>
<tr>
<td>14</td>
<td>.JIDM</td>
<td>Default magtape data mode</td>
</tr>
<tr>
<td>15</td>
<td>.JIRS</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>

48
TOPS-20 Monitor Calls Quick Reference Guide

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 .DBALL WTO and WTOR
                   1 .OBWNR No WTO allowed
                   2 .OBNOM No message allowed
           B10 OB%BSS 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
Code  Symbol  Meaning/Argblk
1    .GOASD  Assign a device
       0    .GEERB Error block address
       1    .GEADD Device designator
2    .GOCAP  Enable capabilities (right half privileges only)
       0    .GEERB Error block address
       1    .GENCP New capability word
3    .GOCJB  Allow CRJOB JSYS to be executed
4  .GOLOG  Allow LOGIN
   0  .GEERB  Error block address
   1  .GELUN  User number

5  .GOCKF  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  .GEQUO  Directory quota
   3  .GERLG  Job # logging out; -1 if caller

10 .GOENQ  Allow setting of ENQ quota
   0  .GEERB  Error block address
   1  .GEEQU  Desired quota
   2  .GEEUN  Job #

11 .GOCRQ  Allow directory creation
   0  .GEERB  Error block address

12 .GOSMT  Allow MOUNT of structure
   0  .GEERB  Error block address
   1  .GESDE  Device designator

13 .GOMDD  Allow entry to MDDT
   0  .GEERB  Error block address

14 .GOCLS  Set scheduler class for a job
   0  .GEERB  Error block address
   1  .GEJOB  Job #
   2  .GECLS  Class desired

15 .GOCLO  Set scheduler class at login
   0  .GEERB  Error block address

16 .GOMTA  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  .GEACD  Desired access bits (FP%xxx)
   5  .GELTP  Label type (.LTxxx)

17 .GOACC  Allow ACCESS or CONNECT
   0  .GEERB  Error block address
   1  .GOACO  Flags from ACCES JSYS
   2  .GDAC1  Directory number

20 .GOOAD  Allow device assignment due to OPENF
   0  .GEERB  Error block address
   1  .GEADD  Device designator

21 .GODNA  Allow DECNET access
   0  .GEERB  Error block address

22 .GOANA  Allow ARPANET access
   0  .GEERB  Error block address

23 .GOATJ  Allow ATTACH
   0  .GDJTB  Target job #
   1  .GEADD  Source TTY number

400000+n  Customer-reserved functions
ERROR BLOCK FORMAT (RET)
Word Symbol   Contents
0  .GESIZ  Count of words in block (including this word)
1  .GEERN  Error number
2  .GEPTR  Byte pointer to error string location
3  .GEBSZ  Maximum bytes user can accept in error string

FUNCTION
GETOK%

GEVEC JSYS 205

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

GFRKH JSYS 164

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

GFRKS JSYS 166

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%GFH) Return relative process handles for each process
      B1(GF%GFS) Return status for each process

51
TOPS-20 Monitor Calls Quick Reference Guide

GFRKS

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
      1B1B + error number to refuse request
AC3:  Pointer to ASCIZ message string (80 characters
      maximum); or 0

RETURNS  +1:  Always

GJINF    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 JSYS 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 JSYS 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)
Word Symbol Meaning
  0 .CDLEN Length of argblk; defaults to 15
  1 .CDPSW Byte pointer to password string
  2 .CDLIQ Working disk storage quota
  3 .CDPRV Capabilities for this user
  4 .CDMOD Mode word
      BO(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
  7 .CDFPT Default file protection (18 bits, R-J)
 10 .CDDPT Directory protection (18 bits, R-J)
 11 .CDRET Default generation retention count
 12 .CDLLD Date of last login
 13 .CDUUGP Address of user group list for this directory
 14 .CDDGP Address of directory group list
 15 .CDSDQ Maximum number of sub-directories allowed
 16 .CDCUG Address of user group list
 17 .CDDAC 0
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 .GTHSZ Returns host table sizes
    Returns
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
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
 1B0 HS%UP Host is up
 1B1 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%NCK Nickname
B21-26 HS%STY System type mask
 1B27 HS%NEW RAS, RAR, RAP, etc

System Type Flags (HS%STY)
Bits Symbol Meaning
 1B26 .HS10X TENEX
 2B26 .HSITS ITS
 3B26 .HSDEC TOPS-10
 4B26 .HSTIP TIP
 5B26 .HSMTIP MTIP
 6B26 .HSELF ELF
 7B26 .HSANT ANTS
10B26 .HSMLT MULTICS
11B26 .HST2O TOPS-20
12B26 .HSUNX UNIX
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

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GJ%FOU</td>
<td>File is to be assigned next higher generation</td>
</tr>
<tr>
<td>1</td>
<td>GJ%NEW</td>
<td>File must not exist (no effect on a parse-only JFN)</td>
</tr>
<tr>
<td>2</td>
<td>GJ%OLD</td>
<td>File must exist (no effect on a parse-only JFN)</td>
</tr>
<tr>
<td>3</td>
<td>GJ%MSG</td>
<td>Print message after filespec if user types ESC; possible messages: !NEW FILE! !NEW GENERATION! !OLD GENERATION! !OK! !CONFIRM!</td>
</tr>
<tr>
<td>4</td>
<td>GJ%CFM</td>
<td>Require confirmation from user (if GJ%FNS is on) to verify filespec</td>
</tr>
<tr>
<td>5</td>
<td>GJ%TMP</td>
<td>File specified is a temporary file</td>
</tr>
<tr>
<td>6</td>
<td>GJ%NS</td>
<td>Search only the 1st specification in a multiple logical name assignment for file</td>
</tr>
<tr>
<td>7</td>
<td>GJ%ACC</td>
<td>JFN cannot be accessed by inferior processes</td>
</tr>
<tr>
<td>8</td>
<td>GJ%DEL</td>
<td>Consider deleted files when searching for file</td>
</tr>
<tr>
<td>9-10</td>
<td>GJ%JFN</td>
<td>Associate JFN in word 10 (.GJJFN) of argblk with filespec according to value (long form only)</td>
</tr>
<tr>
<td>O</td>
<td>(.GJDNU)</td>
<td>Ignore JFN supplied</td>
</tr>
<tr>
<td>2</td>
<td>(.GJERR)</td>
<td>Assign JFN supplied; return error if not available</td>
</tr>
<tr>
<td>3</td>
<td>(.GJALT)</td>
<td>Assign JFN supplied; assign alternate if not available</td>
</tr>
<tr>
<td>11</td>
<td>GJ%IFG</td>
<td>Allow use of wildcards in fields of filespec</td>
</tr>
<tr>
<td>12</td>
<td>GJ%OFG</td>
<td>Associate JFN with filespec string only, not file</td>
</tr>
<tr>
<td>13</td>
<td>GJ%FLG</td>
<td>Return flags in the left half of AC1 if</td>
</tr>
</tbody>
</table>
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
1BO (.GJ%FQL); use highest existing
 generation if OBO (.GJ%FQU)
-1 (.GJ%NHG) Use next higher generation if
none supplied
-2 (.GJLEG) Use lowest existing generation
-3 (.GJALL) Use all generations and assign
JFN to 1st file in group (Gj%IFG
must be set)

ARGUMENT BLOCK (Long Form Only)

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.GJGEN</td>
<td>Flags,.&lt;generation&gt;</td>
</tr>
<tr>
<td>1</td>
<td>.GJSRC</td>
<td>&lt;input JFN&gt;,&lt;output JFN&gt;</td>
</tr>
<tr>
<td>2</td>
<td>.GJDEV</td>
<td>Byte pointer to ASCIZ default device string; or 0 for user’s connected structure</td>
</tr>
<tr>
<td>3</td>
<td>.GJDIR</td>
<td>Byte pointer to ASCIZ default directory string; or 0 for user’s connected directory</td>
</tr>
<tr>
<td>4</td>
<td>.GJNAM</td>
<td>Byte pointer to ASCIZ default filename string; if 0, string in AC2 or input JFN must supply filename</td>
</tr>
<tr>
<td>5</td>
<td>.GJEXT</td>
<td>Byte pointer to ASCIZ default file type string; or 0 for null file type</td>
</tr>
<tr>
<td>6</td>
<td>.GJPRO</td>
<td>Byte pointer to ASCIZ default protection string; or 0 for default directory protection or protection of next lower generation</td>
</tr>
<tr>
<td>7</td>
<td>.GJACT</td>
<td>Byte pointer to ASCIZ default account string; or 0 for user’s LOGIN account (unless changed)</td>
</tr>
<tr>
<td>10</td>
<td>.GJJFN</td>
<td>JFN to associate with filespec if Gj%JFN is set in word 0 (.GJGEN)</td>
</tr>
<tr>
<td>11</td>
<td>.GJF2</td>
<td>Flags,.&lt;count of remaining words in block&gt; if Gj%XTN is set in word 0 (.GJGEN) (OPT)</td>
</tr>
</tbody>
</table>

BO (.G1%RND) Return if filename buffer empty and user attempts to delete character

B2 (.G1%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 (.G1%RCM) Return confirmation message in destination buffer

B4 (.G1%RIE) Return if input buffer empty, and user attempts to delete character
B5(G1%IN) Consider invisible files when searching for file
B6(G1%SLN) Prohibit expansion of logical names

<table>
<thead>
<tr>
<th>No.</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>.GUCPP</td>
<td>Byte pointer to destination string for copy of user’s typescript</td>
</tr>
<tr>
<td>13</td>
<td>.GUCPC</td>
<td>Number of bytes available in destination string; if 0, 130 bytes assumed</td>
</tr>
<tr>
<td>14</td>
<td>.GURITY</td>
<td>Byte pointer to CTRL/R buffer</td>
</tr>
<tr>
<td>15</td>
<td>.GUFBFP</td>
<td>Obsolete</td>
</tr>
<tr>
<td>16</td>
<td>.GJJATR</td>
<td>Pointer to filespec attribute block</td>
</tr>
</tbody>
</table>

ATTRIBUTE BLOCK (Long Form Only)

<table>
<thead>
<tr>
<th>Word</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Count of words in block (including this word)</td>
</tr>
<tr>
<td>1+n</td>
<td>Byte pointer to argument string</td>
</tr>
</tbody>
</table>

ATTRIBUTE VALUES (Long Form Only)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Installation-defined account string</td>
</tr>
<tr>
<td>BDATA</td>
<td>DECnet binary optional data</td>
</tr>
<tr>
<td>BLOCK-LENGTH</td>
<td>Magnetic-tape block length (in bytes)</td>
</tr>
<tr>
<td>BPASSWORD</td>
<td>DECnet binary password</td>
</tr>
<tr>
<td>CHARGE</td>
<td>DECnet account string</td>
</tr>
<tr>
<td>DATA</td>
<td>DECnet optional data</td>
</tr>
<tr>
<td>EXPIRATION-DATE</td>
<td>Magnetic-tape expiration date</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Magnetic-tape record format</td>
</tr>
<tr>
<td>F</td>
<td>Fixed-length records</td>
</tr>
<tr>
<td>D</td>
<td>Variable-length records</td>
</tr>
<tr>
<td>S</td>
<td>Spanned records</td>
</tr>
<tr>
<td>U</td>
<td>Binary files with 36-bit words</td>
</tr>
<tr>
<td>OFF-LINE</td>
<td>NONE - display-only keyword</td>
</tr>
<tr>
<td>P</td>
<td>File protection value (octal)</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>DECnet password string</td>
</tr>
<tr>
<td>POSITION</td>
<td>File sequence number for positioning magnetic-tape</td>
</tr>
<tr>
<td>RECORD-LENGTH</td>
<td>Magnetic-tape record length (in bytes)</td>
</tr>
<tr>
<td>T</td>
<td>NONE - display-only keyword</td>
</tr>
<tr>
<td>USERID</td>
<td>DECnet user ID string</td>
</tr>
</tbody>
</table>

Flags Returned in AC1

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GJ%DEV</td>
<td>Device field of filespec contained wildcards</td>
</tr>
<tr>
<td>1</td>
<td>GJ%UNT</td>
<td>Unit field of filespec contained wildcards</td>
</tr>
<tr>
<td>2</td>
<td>GJ%DIR</td>
<td>Directory field of filespec contained wildcards</td>
</tr>
<tr>
<td>3</td>
<td>GJ%NAM</td>
<td>Filename field of filespec contained wildcards</td>
</tr>
<tr>
<td>4</td>
<td>GJ%EXT</td>
<td>File type field of filespec contained wildcards</td>
</tr>
<tr>
<td>5</td>
<td>GJ%VER</td>
<td>Generation field of filespec contained wildcards</td>
</tr>
<tr>
<td>6</td>
<td>GJ%UHV</td>
<td>File used has highest generation</td>
</tr>
<tr>
<td>7</td>
<td>GJ%NHV</td>
<td>File used has next higher generation</td>
</tr>
<tr>
<td>8</td>
<td>GJ%ULV</td>
<td>File used has lowest generation</td>
</tr>
</tbody>
</table>
9 GJ%PRO Protection field of filespec was given
10 GJ%ACT Account field of filespec was given
11 GJ%TFS Filespec is for temporary file
12 GJ%GND Deleted files were not considered when assigning JFNs
17 GJ%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
Code  Symbol  Function
0     .GTNSZ  Returns negative number of NCP connections
        Returns
AC2: -<# of NCP connections>,0
AC3: -<# of NVTs>,<line # of 1st NVT>

1 .GTNIX  Returns status of connection number
Arguments
AC2: Connection number
AC3: 30-bit address of data block
AC4: -<block length>,<index of 1st item>
Returns
Data in data block
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 .GTNND 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 .GTNJJF 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

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.NCIDX</td>
<td>NCP connection index</td>
</tr>
<tr>
<td>1</td>
<td>.NCFHS</td>
<td>Foreign host</td>
</tr>
<tr>
<td>2</td>
<td>.NCLSX</td>
<td>Local socket</td>
</tr>
<tr>
<td>3</td>
<td>.NCFSK</td>
<td>Foreign socket</td>
</tr>
<tr>
<td>4</td>
<td>.NCFSM</td>
<td>State of connection</td>
</tr>
<tr>
<td>5</td>
<td>.NCLNK</td>
<td>Link</td>
</tr>
<tr>
<td>6</td>
<td>.NCNVX</td>
<td>NVT, or -1 if none</td>
</tr>
<tr>
<td>7</td>
<td>.NCSix</td>
<td>Byte size of connection</td>
</tr>
<tr>
<td>10</td>
<td>.NCMSX</td>
<td>Message allocation</td>
</tr>
<tr>
<td>11</td>
<td>.NCBX</td>
<td>Bit allocation</td>
</tr>
<tr>
<td>12</td>
<td>.NCDA</td>
<td>Desired allocation</td>
</tr>
<tr>
<td>13</td>
<td>.NCBTF</td>
<td>Bits transferred</td>
</tr>
<tr>
<td>14</td>
<td>.NCBPB</td>
<td>Bytes per buffer</td>
</tr>
<tr>
<td>15</td>
<td>.NCCLK</td>
<td>Time-out countdown</td>
</tr>
<tr>
<td>16</td>
<td>.NCSTS</td>
<td>Connection status</td>
</tr>
</tbody>
</table>

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
TOPS-20 Monitor Calls Quick Reference Guide
GTRPW

STATUS WORD

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0(PF%USR)</td>
<td>Page failure - user mode reference</td>
</tr>
<tr>
<td>B5(PF%WRT)</td>
<td>Page failure - write reference</td>
</tr>
<tr>
<td>B14(TSW%RD)</td>
<td>Trap status - read (always on)</td>
</tr>
<tr>
<td>B15(TSW%WT)</td>
<td>Trap status - write (same setting as B5)</td>
</tr>
<tr>
<td>B16(TSW%EX)</td>
<td>Trap status - execute (always on)</td>
</tr>
<tr>
<td>B17(TSW%MN)</td>
<td>Trap status - monitor mode reference</td>
</tr>
<tr>
<td>B18-B35</td>
<td>Address of reference that caused trap</td>
</tr>
</tbody>
</table>

GTSTS JSYS 24

FUNCTION
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

JFN STATUS WORD

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0(GS%DPN)</td>
<td>File is open</td>
</tr>
<tr>
<td>B1(GS%RDF)</td>
<td>File is open for read access</td>
</tr>
<tr>
<td>B2(GS%WRF)</td>
<td>File is open for write access</td>
</tr>
<tr>
<td>B3(GS%XCF)</td>
<td>File is open for execute access</td>
</tr>
<tr>
<td>B4(GS%RND)</td>
<td>File is open for non-append access</td>
</tr>
<tr>
<td>B7(GS%LNG)</td>
<td>File is longer than 512 pages</td>
</tr>
<tr>
<td>B8(GS%EOF)</td>
<td>Last read was past end of file</td>
</tr>
<tr>
<td>B9(GS%ERR)</td>
<td>File may be in error</td>
</tr>
<tr>
<td>B10(GS%NAM)</td>
<td>Filespec is associated with this JFN</td>
</tr>
<tr>
<td>B11(GS%AST)</td>
<td>JFN is parse-only</td>
</tr>
<tr>
<td>B12(GS%ASG)</td>
<td>JFN is currently being assigned</td>
</tr>
<tr>
<td>B13(GS%HLT)</td>
<td>I/O errors are considered terminating</td>
</tr>
<tr>
<td></td>
<td>conditions</td>
</tr>
<tr>
<td>B17(GS%FRK)</td>
<td>JFN is restricted</td>
</tr>
<tr>
<td>B18(GS%PLN)</td>
<td>If on, file line numbers are passed</td>
</tr>
<tr>
<td></td>
<td>during input; if 0, line numbers are</td>
</tr>
<tr>
<td></td>
<td>stripped before input</td>
</tr>
<tr>
<td>B32-B35(GS%MOD)</td>
<td>Data mode of the file</td>
</tr>
</tbody>
</table>

GTYP JSYS 303

FUNCTION
Returns the terminal type number for the specified terminal line.

CALLING SEQUENCE
AC1: Terminal designator

RETURNS  +1: Always, with
AC2: Terminal type number
AC3: B0-17  # of input buffers to allocate
      B18-35  # of output buffers to allocate

HALTF  JSYS 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).

HFORK  JSYS 162

FUNCTION
Halts one or more inferior processes.

CALLING SEQUENCE
AC1: Process handle (inferior processes only)

RETURNS  +1: Always

HPTIM  JSYS 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  .HPELP  Elapsed time since system startup
      1  .HPRNT  CPU runtime for this process

HSYS  JSYS 307

FUNCTION
Initiates an orderly shutdown of timesharing.

RESTRICTIONS
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: BO(IC%DSA) Apply daylight savings according to setting of B1(IC%ADS)
     B1(IC%ADS) Apply daylight savings if BO(IC%DSA) is on
     B2(IC%UTZ) Use time zone in B12-17; if off, use local zone
     B3(IC%JUD) 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: BO and B2 On for compatibility with DDCNV
                   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%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 and month does not agree with setting of B2(IT%SNM)
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 hhmm: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

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%NDA 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%NDA was set
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

BO(IT%nda) 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%smn) 2nd number in date is month

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

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%ncn) Colon cannot be used to separate hours and minutes

B10(IT%amc) 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
Code    Symbol    Meaning
0       .INLUB    List logical names defined for this job
1       .INLSY    List logical names defined for system

JFNS    JSYS 30

FUNCTION
Returns the filespec currently associated with the JFN.

CALLING SEQUENCE
AC1: Destination designator for ASCIZ 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 VALUE
Value    Symbol    Meaning
0        .JSNDP    Do not output this field
1        .JSADF    Always output this field
2        .JSSDD    Suppress this field if system default

FORMAT CONTROL FIELDS
Field    Meaning
B0-2(JS%DEV)  Output for device field
B3-5(JS%DIR)  Output for directory field
B6-8(JS%NAM)  Output for filename field (2 is illegal)
B9-11(JS%TPY) Output for file type field (2 is illegal)
B12-14(JS%GEN) Output for generation number field
B0-14(JS%SPC) Output for all filespec fields named above
B15-17(JS%PRD) Output for protection field
B18-20(JS%ACT) Output for account field
B21-23(JS%TMP) Return $T if appropriate
B22(JS%SIZ)   Return size of file in pages
B23(JS%CDR)   Return creation date
B24(JS%LWR)   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

**B27(JS%ATR)**
Return filespec attributes if appropriate

**B28(JS%AT1)**
Return specification attribute referenced in AC4

**B29(JS%QFL)**
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 $T$

**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

**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

**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
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>.LNSJB</td>
<td>Obtain job-wide definition of logical name</td>
</tr>
<tr>
<td>1</td>
<td>.LNSSY</td>
<td>Obtain system-wide definition of logical name</td>
</tr>
</tbody>
</table>

LOGIN  JSYS 1

FUNCTION
Logs a job into the system.

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

CALLING SEQUENCE
AC1: 36-bit user number for login
AC2: Pointer to beginning of password string
AC3: 5B2!<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.

RESTRICTIONS
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

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BO</td>
<td>MO%LCP</td>
<td>Line printer is lowercase</td>
</tr>
</tbody>
</table>

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>.MOLVF</td>
<td>Load VFU from file indicated by JFN</td>
</tr>
<tr>
<td>34</td>
<td>.MOLTR</td>
<td>Load translation RAM from file indicated by</td>
</tr>
</tbody>
</table>

69
**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.

**METER% 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>.MREA</td>
<td>Read process execution accounting meter doubleword; returns EBOX busy time (number of EBOX ticks)</td>
</tr>
<tr>
<td>2</td>
<td>.MERMA</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**

```
+-----------------------------------------------+-----------------------------------------------+
| AC2                                                                 | AC3                                                                 |
| High Order Part | | Low Order Part | | Reserved |
|                 | 0 |                 | 23 | 24 | 35 |
```

**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: B0-17 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     .IPCFP  <length of message>,<destination address>
4     .IPCFD  User number of sender (RET)
5     .IPCFC  Enabled capabilities of sender (RET)
6     .IPCSD  Number of sender's connected directory (RET)
7     .IPCAS  Account string of sender (RET)
10    .IPCCLL 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%NOA  Do not allow other processes to use created PID (ignored unless IP%CPD 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

MSEND  JSYS 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
TDPS-20 Monitor Calls Quick Reference Guide

MSEND

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    .IPCFL  Flags
  1    .IPCFS  PID of sender; address of PID if IP%CFS or IP%CFR is set in word .IPCFL; 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 .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%NOA  Do not allow other processes to use created PID (ignored unless IP%CPD set)
B18   IP%CFF  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 .IPCFL
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   .IPCPF  Insufficient privileges
       16   .IPCUF  Invalid function
       67   .IPCFSN  <SYSTEM>INFO needs name
       72   .IPCFF  <SYSTEM>INFO free space exhausted
       74   .IPCBP  PID has no name or is invalid
       75   .IPCQDN  Duplicate name has been specified
       76   .IPCQNN  Unknown name has been specified
77 .IPCEH Invalid name has been specified

B30-32 IP%CFC 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 .IPCFN Process's input queue contains undeliverable packet

FORMAT OF REQUEST PACKET TO <SYSTEM>INFO

Word Symbol Meaning
0 .IPCI0 <user-defined code>,,<<SYSTEM>INFO function>
1 .IPCI1 PID to receive copy of <SYSTEM>INFO's response
2 .IPCI2 Function-specific argument

<SYSTEM>INFO FUNCTION CODES

Function Argument Meaning
.IP CIW Name Return PID associated with specified name in word .IPCI1
.IP CIG PID Return name associated with specified PID in word .IPCI1
.IP CII ASCIZ name Assign specified name to PID of process making request
.IP CIIJ ASCIZ name Same as .IPCIII function
.IP CIK PID Inform PID when PID in word .IPCI2 is deleted (WHL/DPR)
.IP CIS 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
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/OPR/MNT</td>
<td>Return status of next disk unit</td>
</tr>
<tr>
<td>1</td>
<td>MSRSU</td>
<td>WH/OPR/MNT</td>
<td>Return status of given disk unit</td>
</tr>
<tr>
<td>2</td>
<td>MSMNT</td>
<td>WH/OPR</td>
<td>Mount structure</td>
</tr>
<tr>
<td>3</td>
<td>MSDIS</td>
<td>WH/OPR</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>MSSSS</td>
<td>WH/OPR</td>
<td>Change status of structure</td>
</tr>
<tr>
<td>6</td>
<td>MSINI</td>
<td>WH/OPR</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/OPR</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>MSOFL</td>
<td>WH/OPR</td>
<td>Receive interrupt when disk comes on-line</td>
</tr>
<tr>
<td>16</td>
<td>MSIIC</td>
<td>WH/OPR</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%OFL) 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>
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 RMD3
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 .MSRSU 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

0-27 .MSRUS Same as .MSRNU
0 .MSMNT Pointer to ASCIZ string for structure name
1 .MSTAL Pointer to ASCIZ string for structure alias
2 .MSTFL Flags, <# units in structure>
BO(MS%NFH) Do not fix bad HOME blocks
B1(MS%NFB) Do not fix bad BAT blocks
B2(MS%XCL) Mount structure for exclusive use by job
B3(MS%IGN) 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
BO(MS%PS) Structure is public
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 .MGNUM Number of units in structure
3 .MGCOUNT Mount count for this structure
4 .MGFC Count for file count for this structure
5 .MGSI Pointer to ASCIZ string for structure's physical ID

0 .MSSSN Device designator, or pointer to ASCIZ structure alias string
1 .MSSTT Word containing new values for bits being changed
2 .MSMWW 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%RWSC) Read-after-write checking in swapping area
B8(MS%RWDD) Read-after-write checking in data area

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

B1(MS%NFB) 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 .MSRRE Reconstruct root directory
3 .MSWRE Write new HOME blocks
4 .MSRIS Rebuild index table

B18-35(.MSINU) # of units in structure
3-5 .MSISU 3 words of data for each unit in
structure
0 .MSICH Channel # of unit
1 .MSICT Controller # of unit
2 .MSIUN Unit # of unit
6 .MSIST Status word (reserved)
7 .MSISW Number of pages for swapping on structure
10 .MSIFE Number of pages for front-end file system
11-13 .MSIU Unit ID (3 words of ASCII)
14-16 .MSIOD Owner ID (3 words of ASCII)
17-21 .MSIFI File system ID (3 words of ASCII; reserved)
22 .MSIFB Number of pages for file BOOTSTRAP.BIN
23 .MSISN Serial number of CPU for which structure is used in booting system

.MSICF 0 .MSDEV Device designator, or pointer to ASCIZ structure alias string
1 .MSJOB Number of job (if not current job) whose mount count is to be incremented; (optional; enabled WHL/DPR)

.MSDCF 0 .MSDEV Device designator, or pointer to ASCIZ structure alias string
1 .MSJOB Number of job (if not current job) whose mount count is to be decremented; (optional; enabled WHL/DPR)

.MSGSU 0 .MSUAL Device designator, or pointer to ASCIZ structure alias string
1 .MSUFL <flag bits>,<0
BO(MS%GTA) Return users who have accessed structure
B1(MS%GTM) Return users who have incremented mount count
B2(MS%GTC) Return users who are connected to structure

.MSICF 0 .MSDEV Device designator, or pointer to ASCIZ structure alias string

.MSDCF 0 .MSDEV Device designator, or pointer to ASCIZ structure alias string

.MSOFL 0 .MSCHN Place process on software interrupt channel; if -1, deassign previously assigned channel

DATA RETURNED BY FUNCTION .MSGSU
Word Symbol Meaning
1 .MSUFL <flag bits from call>,<# of items returned>
**TOPS-20 Monitor Calls Quick Reference Guide**

**MSSTR**

\[ n+1 \]

- MSUJ 1 <flag bits for job>,<job #>
- MSUJ 2 <flag bits for job>,<job #>
- BO(MS%GTA) Job accessed structure
- BI(MS%GTM) Job incremented mount count for structure
- B2(MS%GTC) Job connected to structure

### 11-Formatted ASCII

<table>
<thead>
<tr>
<th>XX</th>
<th>CHAR 1</th>
<th>CHAR 0</th>
<th>XX</th>
<th>CHAR 3</th>
<th>CHAR 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>CHAR 5</td>
<td>CHAR 4</td>
<td>XX</td>
<td>CHAR 7</td>
<td>CHAR 6</td>
</tr>
<tr>
<td>XX</td>
<td>CHAR 9</td>
<td>CHAR 8</td>
<td>XX</td>
<td>CHAR 11</td>
<td>CHAR 10</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

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

DECnet FUNCTION CODES
Code  Symbol  Meaning/Arguments
24    .MOACN   Allow network task to enable interrupt channels for some tasks
          AC3: B0-8(MO%CDN) Connect event pending
               B9-17(MO%INA) Interrupt message available
               B18-26(MO%DAV) Data available
          Values for AC3 fields
          nn  # of channel to be enabled: 0 to 5, 23. to 35.
25    .MORLS   Return logical link status
          AC3: <flag bits>,<disconnect code> (RET)
          BO(MO%CON) Link is connected
          B1(MO%SRV) Link is a server
          B2(MO%WFC) Link waiting for connection
          B3(MO%WCC) Link waiting for connect confirmation
          B4(MO%EOM) Link has entire message to be read
          B5(MO%ABT) Link has been aborted
          B6(MO%SYN) Link has been closed normally
          B7(MO%INT) Link has interrupt message available
          B8(MO%LWC) Link has been previously connected
          Disconnect codes
          0  .DCX0  No special error
          1  .DCX1  Resource allocation failure
          2  .DCX2  Destination node does not exist
          3  .DCX3  Node shutting down
          4  .DCX4  Destination process does not exist
          5  .DCX5  Invalid name field
          6  .DCX6  Destination process queue
overflow
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 .MORHN Return ASCII host node name at other end of logical link
AC3: Pointer to string for host name (8-bit)
27 .MORTN Return unique task name associated with this end of logical link
AC3: Pointer to string for task name (8-bit)
30 .MORUS Return source task user identification supplied in connect initiate message
AC3: Pointer to string for user ID (8-bit)
31 .MORPW Return source task’s password supplied in connect initiate message
AC3: Pointer to string for password (8-bit)
32 .MORAC Return account string supplied by source task in connect initiate message
AC3: Pointer to string for account (8-bit)
33 .MORDA Return optional data supplied in connect/disconnect messages
AC3: Pointer to string for data (8-bit)
34 .MORCN Return object type used by source task to address connection
35 .MORIM 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 maxiumum)

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 TTY 
44 .MOSNH Set network host
AC3: Address of argbk
0 Count including this word
1 .SHTTY ID of TTY controlling local
job
2 .SHESC Flags, <ASCII escape char>
SH%LPM Local page mode

FRONT-END FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.MOEUF</td>
<td>Flush TOPS-20 buffers and send all data to front end</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: 0 Flush buffers and send EOF to FE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#0 Flush buffers only</td>
</tr>
<tr>
<td>4</td>
<td>.MODTE</td>
<td>Assign specified device to DTE controller on front end (enabled WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Device type (WHL/OPR)</td>
</tr>
</tbody>
</table>

MTA/MT FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.MOCLE</td>
<td>Clear any error flags from previous MTOPR</td>
</tr>
<tr>
<td>1</td>
<td>.MOREW</td>
<td>Rewind tape; if labeled, mount 1st volume in set and position at BOT</td>
</tr>
<tr>
<td>2</td>
<td>.MOSDR</td>
<td>Set direction of tape motion for reading (unlabeled only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: 0 Read forwards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Read backwards</td>
</tr>
<tr>
<td>3</td>
<td>.MOEUF</td>
<td>Write tape mark</td>
</tr>
<tr>
<td>4</td>
<td>.MOSDM</td>
<td>Set hardware data mode for tape data transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Hardware data mode</td>
</tr>
<tr>
<td>5</td>
<td>.MOSRS</td>
<td>Set record size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Record size in bytes</td>
</tr>
</tbody>
</table>
6 .MOFWR  Advance one record in read direction
7 .MOBK R  Back up one record from read direction
10 .MOEOT  Advance to EOT (unlabeled) or EOV (labeled)
11 .MORUL  Rewind and unload tape (illegal for MOUNTed tapes)
12 .MORDN  Return density
13 .MOERS  Erase tape gap (unlabeled only)
14 .MORDM  Return hardware data mode
15 .MDRRS  Return record size
16 .MDFWF  Advance to next tape mark
17 .MOBK F  Backup to last tape mark or BOT
20 .MOSPR  Set parity
       AC3: Desired parity
          0 .SUPRO Odd parity
          1 .SUPRE Even parity
21 .MORPR  Return parity
22 .MONRB  Return number of bytes remaining in current record
23 .MOFOU  Force output of partial records during sequential write
24 .MOSDN  Set density (unlabeled only)
       AC3: Desired density
25 .MOINF  Return tape information
       AC3: Address of argblk
       0 .MOICT Word count not including this word
       1 .MOITP MTA type code
       2 .MOIID 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 .MOIFIC Number of files on tape
       10 .MOISR Number of soft read errors
       11 .MOISW Number of soft write errors
       12 .MOIHR Number of hard read errors
       13 .MOIHW Number of hard write errors
       14 .MOIRF Number of frames read
       15 .MOIWF Number of frames written
26 .MORDR  Return read direction
       AC3: 0 Forwards
            1 Backwards
27 .MOSID  Set reel ID of mounted tape (enabled WHL/DPR)
       AC3: 36-bit reel ID
30 .MOIEL  Set error logging for tape
       AC3: 0 Log errors
            #0 Do not log errors
31 .MONOP  Wait for all activity to stop
32 .MOLOC  Identify 1st volume in MOUNT request or next volume for volume switch (WHL/DPR)
       AC3: Pointer to argblk
       0 .MOCNT Word count
       1 .MOMTN MT unit # to associate with MTA
       2 .MOLBT Label type

82
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>MODNS</td>
<td>Density</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MOAVL</td>
<td>Address of volume labels</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MONVL</td>
<td># of volume labels at MOAVL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MOCVN</td>
<td>Volume number in volume set</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MOVSN</td>
<td>SIXBIT file set identifier</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>MOSTA</td>
<td>Return current magtape status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AC3:</td>
<td>Address of argblk</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>MOCNT</td>
<td>Word count including this word</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MODDN</td>
<td>Density flags (RET)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1(SJ%CP2)</td>
<td>200 BPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2(SJ%CP5)</td>
<td>556 BPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3(SJ%CP8)</td>
<td>800 BPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4(SJ%C16)</td>
<td>1600 BPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B5(SJ%C62)</td>
<td>6250 BPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MODDM</td>
<td>Data mode flags (RET)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1(SJ%CMC)</td>
<td>Core dump</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2(SJ%CMG)</td>
<td>SIXBIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3(SJ%CMA)</td>
<td>ANSI ASCII</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4(SJ%CM8)</td>
<td>Industry compatible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B5(SJ%CMH)</td>
<td>High density mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MOTRK</td>
<td>Recording track flags (RET)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1(SJ%7TR)</td>
<td>7-track drive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2(SJ%9TR)</td>
<td>9-track drive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MOCST</td>
<td>Tape status flags (RET)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BO(SJ%0FS)</td>
<td>Off-line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1(SJ%MAI)</td>
<td>Maintenance mode enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B2(SJ%MRQ)</td>
<td>Maintenance mode requested</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B3(SJ%BOT)</td>
<td>Beginning of tape</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B4(SJ%REW)</td>
<td>Rewinding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B5(SJ%WLK)</td>
<td>Write locked</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MODVT</td>
<td>Device type (RET)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>MOOFL</td>
<td>Enable interrupts for on-line/off-line transition (WHL/NPR)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>MOPST</td>
<td>Set interrupt channel to indicate availability of UHL(BOV)/UTL(EOD) labels</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AC3:</td>
<td>PSI channel; -1 to clear</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>MORVL</td>
<td>Rewind current labeled tape volume</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>MOVLS</td>
<td>Switch volumes for unlabeled multi-volume set</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC3:</td>
<td>Address of argblk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Word count including this word</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Flags, &lt;function code&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VSMNV</td>
<td>Mount absolute volume #</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VSFST</td>
<td>Mount 1st volume in set</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>VSLST</td>
<td>Mount last volume in set</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>VSMRV</td>
<td>Mount relative volume #</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>VSFLS</td>
<td>Force volume switch (labeled only)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Volume number (if required)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>MONTR</td>
<td>Set translate flag (EBCDIC ==&gt; ASCII; labeled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

83
only)
AC3: 0 Clear translate flag
     #0 Set translate flag
46 .MORDL Read user header labels
AC3: Pointer to string for label
AC3: Byte pointer to label contents (must be
     76 bytes)
AC3: Label identifier code (any ASCII char)
47 .MOWUL Write user header or trailer labels (labeled
     only)
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 .TPFNC 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(MD%MSG) Suppress CTY device
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>.MONOP</td>
<td>Wait for all activity to stop</td>
</tr>
<tr>
<td>32</td>
<td>.MDLVF</td>
<td>Load line printer's VFU from file referenced in argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  JFN of file containing VFU</td>
</tr>
<tr>
<td>33</td>
<td>.MORVF</td>
<td>Read name of current VFU file in monitor's data base</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  Pointer to string for ASCIZ name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2  Number of bytes in string</td>
</tr>
<tr>
<td>34</td>
<td>.MOLTR</td>
<td>Load line printer's translation RAM from file referenced in argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  JFN of file containing translation RAM</td>
</tr>
<tr>
<td>35</td>
<td>.MORTR</td>
<td>Read name of current translation RAM file in monitor's data base</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  Pointer to string for ASCIZ name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2  Number of bytes in string</td>
</tr>
<tr>
<td>36</td>
<td>.MOSTS</td>
<td>Set status of line printer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  Software status word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO(MO%LCP)                    Printer is lowercase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B12(MO%EOF)                   Set MO%EOF when all data printed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B14(MO%SER)                   Clear software error condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2  Value for page counter register</td>
</tr>
<tr>
<td>37</td>
<td>.MORST</td>
<td>Read line printer status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: Address of argblk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1  Status word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO(MO%LCP)                    Printer is lowercase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(MO%RLD)                    FE has been reloaded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B10(MO%FER)                   Fatal hardware error occurred</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B12(MO%EOF)                   All data sent has been printed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B13(MO%IOP)                   Output in progress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B14(MO%SER)                   Software error occurred</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B15(MO%HE)                    Hardware error occurred</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B16(MO%OL)                    Printer is off-line</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B17(MO%FNX)                   Printer does not exist</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B30(MO%RPE)                   RAM parity error occurred</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B31(MO%LVU)                   Printer has optical VFU</td>
</tr>
</tbody>
</table>
B33(M0%LVF) VFU error occurred
B34(M0%LCL) Character interrupt occurred
B35(M0%LPC) Page counter register overflowed

40 .MDFLO Flush any output not yet printed

PCDP FUNCTION CODES
Code Symbol Meaning/Arguments
27 .MDPSI Enable software interrupt on nonfatal device conditions
AC3: Address of argblk
   0 Word count including this word
   1 Interrupt channel number
   2 Flags
      B0(M0%MSG) Suppress CTY device messages

37 .MORST Read card punch status
AC3: Address of argblk
   0 Word count including this word
   1 Status word
      B10(M0%FER) Fatal error condition
      B12(M0%EOF) All pending output processed
      B13(M0%IDP) Output in progress
      B14(M0%SER) Software error occurred
      B15(M0%HE) Hardware error occurred
      B16(M0%OL) Card punch is off-line
      B17(M0%FNX) Punch doesn't exist
      B32(M0%HEM) Hopper empty or stacker full
      B33(M0%SCK) Stack check
      B34(M0%PCK) Pick check
      B35(M0%RCK) Read check

PCDR FUNCTION CODES
Code Symbol Meaning/Arguments
27 .MDPSI Enable software interrupt on nonfatal device conditions
AC3: Address of argblk
   0 Word count including this word
   1 Interrupt channel number
   2 Flags
      B0(M0%MSG) Suppress CTY device messages

37 .MORST Read card reader status
AC3: Address of argblk
   0 Word count including this word
   1 Status word
      B0(M0%COL) Card reader is online
      B1(M0%RDL) FE has been reloaded
      B10(M0%FER) Fatal hardware error occurred
B12(MO%ECF) Card reader at EOF
B13(MO%ICP) Input in progress
B14(MO%SER) Software error occurred
B15(MO%HE) Hardware error occurred
B16(MO%OL) Reader is off-line
B17(MO%FNX) Reader does not exist
B31(MO%SFL) Output stacker full
B32(MO%HEM) Input hopper empty
B33(MO%CK) Stack check
B34(MO%PCK) Pick check
B35(MO%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: BO(MO%WFI) Enable waiting-for-input interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(MO%OIR) Enable output-is-ready interrupt</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B12-17(MO%SIC) Interrupt channel for PTY output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Function code</td>
</tr>
</tbody>
</table>

25 .MOPIH Determine if PTY job needs input
26 .MOBAT Set batch control bit

AC3: 0 Job not under batch
      1 Job under batch

TTY FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning/Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>.MOPIH</td>
<td>Determine if TTY job needs input</td>
</tr>
<tr>
<td>26</td>
<td>.MDSPD</td>
<td>Set terminal line speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC2: BO(MO%RMT) Remote line (WHL/DPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B1(MO%AUT) Remote autobaud line (WHL/DPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Function code</td>
</tr>
</tbody>
</table>

27 .MORSP Return terminal line speed
30 .MORLW Return terminal page width
31 .MOSLW Set terminal page width
      AC3: Page width
32 .MORLL Return terminal page length.
33 .MOSLL Set terminal page length
      AC3: Page length
34 .MOSNT Set terminal receive-system-messages code
      AC3: 0(.MOSMY) Allow messages
      1(.MOSMN) Refuse messages
35 .MORNT Return terminal receive-system-messages code
36 .MOSIG Set terminal input on inactive line code
      AC3: 0 Do not ignore input
      1 Ignore input
37 .MORBm Read 128-character break mask
40 .MOSBM Set 128-character break mask
      AC3: Address of argblk
      0 Word count not including this word
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 .MOXOF Set pause-at-end-of-page mode
   AC3: 0(.MOOFF) Disable pause-at-end-of-
   page mode
   1(.MOONX) Enable pause-at-end-of-page
   mode
44 .MORXQ Read end-of-page mode
45 .MOSLC Set terminal's line counter
   AC3: Line counter value
46 .MORLC Read terminal's line counter
47 .MOSLM Set line maximum
   AC3: Line maximum value
50 .MORLM Read line maximum
51 .MTOPS Assign terminal interrupt channels
   AC3: Address of argblk
   0 Word count including this word
   1 B0-17 Output PSI channel
      B18-35 Input PSI channel
52 .MOPCS Set terminal pause/unpause characters
   AC3: <pause character>,<unpause character>
53 .MOPCR Read terminal pause/unpause characters

MTU% JSYS 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 .MTNV  Declare volume switch error
0 .MTCNT Word count
1 .MTCOD Error code to return to user
2 .MTPTR Byte pointer to operator response
2 .MTRAL Read labels
0 .MTCNT Word count
1 .MTVL1 Byte pointer to area for VOL1 label
2 .MTVL2 Byte pointer to area for VOL2 label
3 .MTHD1 Byte pointer to area for HDR1 label
4 .MTHD2 Byte pointer to area for HDR2 label
3 .MTASL Return assignment information

88
0 .MTCNT Word count
1 .MTPHU Returned MTA # associated with MT
4 .MTCVV Clear volume ID for specified MT

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

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

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

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

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>.MUENB</td>
<td>Enable specified PID to receive packets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID</td>
</tr>
<tr>
<td>2</td>
<td>.MUDIS</td>
<td>Disable specified PID from receiving packets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID</td>
</tr>
<tr>
<td>3</td>
<td>.MUGTI</td>
<td>Return PID associated with &lt;SYSTEM&gt;INFO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID or job #</td>
</tr>
<tr>
<td>4</td>
<td>.MUCPI</td>
<td>Create private copy of &lt;SYSTEM&gt;INFO for job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(enabled IPCF)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID to be assigned to &lt;SYSTEM&gt;INFO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 PID or job # creating private copy</td>
</tr>
<tr>
<td>5</td>
<td>.MUDES</td>
<td>Delete specified PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID</td>
</tr>
<tr>
<td>6</td>
<td>.MUCRE</td>
<td>Create PID for specified process or job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Flags.,&lt;process handle or job #</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B6(IP%JWP) PID is job-wide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B7(IP%NOA) PID is not available to other processes</td>
</tr>
<tr>
<td>7</td>
<td>.MUSSQ</td>
<td>Set send/receive quotas for specified PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(enabled IPCF)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 B18-26 New send quota</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B27-35 New receive quota</td>
</tr>
<tr>
<td>10</td>
<td>.MUCHD</td>
<td>Change job # associated with specified PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(enabled WHL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 PID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 New job # or PID belonging to new job</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
MUUTIL

11 .MUFOJ 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 .MUDFI 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/OPR/IPCF)
   1 Index into system PID table
   2 PID
25 .MURSP 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/OPE)</td>
</tr>
<tr>
<td>1</td>
<td>.NDGLN</td>
<td>Get local node name</td>
</tr>
<tr>
<td>2</td>
<td>.NDSNM</td>
<td>Set local node number (WHL/OPE)</td>
</tr>
<tr>
<td>3</td>
<td>.NDGNM</td>
<td>Get local node number</td>
</tr>
<tr>
<td>4</td>
<td>.NDSL</td>
<td>Set loopback port (KS-10 only; WHL/OPE/MNT)</td>
</tr>
<tr>
<td>5</td>
<td>.NDCLP</td>
<td>Clear loopback port (KS-10 only; WHL/OPE/MNT)</td>
</tr>
<tr>
<td>6</td>
<td>.NDCLP</td>
<td>Find loopback port (KS-10 only)</td>
</tr>
<tr>
<td>7</td>
<td>.NDSL</td>
<td>Set node table (WHL/OPE)</td>
</tr>
<tr>
<td>10</td>
<td>.NDGNT</td>
<td>Get node table</td>
</tr>
<tr>
<td>11</td>
<td>.NDGNT</td>
<td>Set topology interrupt channel</td>
</tr>
<tr>
<td>12</td>
<td>.NDGNT</td>
<td>Clear topology interrupt channel</td>
</tr>
<tr>
<td>13</td>
<td>.NDGVR</td>
<td>Get NSP version number</td>
</tr>
</tbody>
</table>

91
TOPS-20 Monitor Calls Quick Reference Guide

NODE

<table>
<thead>
<tr>
<th>Word</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NDNAM</td>
</tr>
<tr>
<td>1</td>
<td>NDSTA</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NDNXT</td>
</tr>
<tr>
<td>3-4</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Word</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NDVER</td>
</tr>
<tr>
<td>1</td>
<td>NDECO</td>
</tr>
<tr>
<td>2</td>
<td>NDCST</td>
</tr>
</tbody>
</table>
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: BO(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%O0V) 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

NTMAN% 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 .NTNOD 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 .NTEVT Event
Selectors for .NTRET
   -1 .NTKNO Known items
   -2 .NTACT Active items
   -3 .NTLOP 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

ODCNV JSYS 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: BO(IC%DSA) Apply daylight savings according to
     B1(IC%ADS)
     B1(IC%ADS) Apply daylight savings if 1BO(IC%DSA)
     B2(IC%UTZ) Use time zone in B12-B17(IC%TMZ)
     B3(IC%JUD) Apply Julian day format
     B12-B17(IC%TMZ) Time zone to use if 1B2(IC%UTZ)

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

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
Converting 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-mmmy 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%NMN)
    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 delimiting; if
    B6(OT%DAM), output month day, year
    B8(OT%SLA)
    Output numeric date with slash delimiter; if
    OB7 and OB8, output day-month-year with dash
delimiter
B9(OT%NTM)
    Do not output time and ignore B10-13
B10(OT%NSC)
    Do not output seconds
B11(OT%12H)
    Output time in 12-hour format with AM or PM
B12(OT%NCO)
    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)
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(IC%ADS) Apply daylight savings on output
     B12-17(IC%TMZ) Time zone desired
     B18-35(IC%TIM) Local time in seconds since midnight
AC5: Format option flags

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

FORMAT OPTION FLAGS
B0(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%NMN) 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 1B6(OT%DAM), output month day, year
B8(OT%SLA) Output numeric date with slash delimiter; if
            OB7 and OB8, output day-month-year with dash
delimiter
B9(OT%NTM) Do not output time and ignore B10-13
B10(OT%NSC) Do not output seconds
B11(OT%12H) Output time in 12-hour format with AM or PM
B12(OT%NC0) Output time without colon between hours and
            minutes
B13(OT%TMZ) Output time with "-" and time zone (US zones
            and Greenwich Mean only)
B17(OT%SCL) Suppress columnization of date and time (omit
            leading spaces and zeros)

FUNCTION
Opens the given file.

CALLING SEQUENCE
AC1: 0,, JFN
AC2:  B0-5(OF%BSZ) Byte size (maximum of 36.; 36. default)
     B6-9(OF%MOD) Data mode in which to open file
     B18(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%NWT)  Do not wait if access disallowed; return error
B29(OF%RTD)  Enforce restricted access
B30(OF%PLN)  Disable line number checking
B31(OF%DUD)  Suppress system updating of modified pages in memory to thawed files on disk unless CLOSF or UFPGS issued
B32(OF%OFL)  Open device even if off-line
B33(OF%FDT)  Force update of .FREF (last read) in FDB 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-J 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

PDVDP%    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
RETURNS +1: Always, with data returned in the data block, and updated count in .POCT2 if needed.

FUNCTION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.POGET</td>
<td>Return all PDVAs within range specified in argblk</td>
</tr>
<tr>
<td>1</td>
<td>.POADD</td>
<td>Add PDVAs specified in data block to system’s database for process</td>
</tr>
<tr>
<td>2</td>
<td>.POREM</td>
<td>Remove PDVAs within range specified in argblk from system’s data base for process</td>
</tr>
<tr>
<td>3</td>
<td>.PONAM</td>
<td>Return ASCIZ name of program referenced in word .PVNAM of PDV</td>
</tr>
<tr>
<td>4</td>
<td>.POVER</td>
<td>Return program version number from word .PVVER of PDV</td>
</tr>
<tr>
<td>5</td>
<td>.POLOC</td>
<td>Return all PDVAs of PDVs for program referenced in AC3</td>
</tr>
</tbody>
</table>

ARGUMENT BLOCK

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.POCT1</td>
<td>Number of words in argblk</td>
</tr>
<tr>
<td>1</td>
<td>.POPHD</td>
<td>Handle of desired process</td>
</tr>
<tr>
<td>2</td>
<td>.POCT2</td>
<td># of words in data block; on return &lt;# of words available&gt;,&lt;# of words returned&gt;</td>
</tr>
<tr>
<td>3</td>
<td>.PODAT</td>
<td>Starting address of data block for returned data</td>
</tr>
<tr>
<td>4</td>
<td>.POADR</td>
<td>Starting address of memory range</td>
</tr>
<tr>
<td>5</td>
<td>.POADE</td>
<td>Ending address of memory range</td>
</tr>
</tbody>
</table>

FORMAT OF PROGRAM DATA VECTOR

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.PVCNT</td>
<td>Length of PDV including this word</td>
</tr>
<tr>
<td>1</td>
<td>.PVNAM</td>
<td>Pointer to ASCIZ program name string for this PDV</td>
</tr>
<tr>
<td>2</td>
<td>.PVSTR</td>
<td>Program starting address</td>
</tr>
<tr>
<td>3</td>
<td>.PVREE</td>
<td>Program reenter address</td>
</tr>
<tr>
<td>4</td>
<td>.PVVER</td>
<td>Program version number</td>
</tr>
<tr>
<td>5</td>
<td>.PVMEM</td>
<td>Address of block of memory containing length in Word 0 and memory map in remaining words</td>
</tr>
<tr>
<td>6</td>
<td>.PVSYM</td>
<td>Address of program symbol table</td>
</tr>
<tr>
<td>7</td>
<td>.PVTM</td>
<td>Time at which program was compiled</td>
</tr>
<tr>
<td>10</td>
<td>.PVCVR</td>
<td>Version number of compiler</td>
</tr>
<tr>
<td>11</td>
<td>.PVLTM</td>
<td>Time at which program was loaded</td>
</tr>
<tr>
<td>12</td>
<td>.PVLVR</td>
<td>Version number of LINK</td>
</tr>
<tr>
<td>13</td>
<td>.PVMON</td>
<td>Address of monitor data block (not used currently)</td>
</tr>
<tr>
<td>14</td>
<td>.PVPRG</td>
<td>Address of program data block (not used currently)</td>
</tr>
<tr>
<td>15</td>
<td>.PVCST</td>
<td>Address of customer-defined data block</td>
</tr>
</tbody>
</table>
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's 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%NCH)  Pages will not be cached
      B3(LK%AOL)  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:  JFN,<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)
<process handle>,<process page #> (Case IV)
JFN,<file page #> (Case V)

AC3:  
BO(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 1BO(PM%CNT)

RETURNS  +1: Always

PMCTL  JSYS 56C

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>
</table>
| 0    | .MCRCE | Return status of cache memory
|      |        | 0 .MCCST  If 1B35(MC%CEN), cache is enabled            |
| 1    | .MCSCE | Set status of cache memory
|      |        | 0 .MCCST  Enable cache if 1B35(MC%CEN)                 |
| 2    | .MCRPS | Return status of specified page
|      |        | 0 .MCPPN  - count,<physical page #>                    |
|      |        | 1 .MCPST  Returned page status
|      |        | 0 .MCPSA  Page available                               |
|      |        | 1 .MCPSS  Page in transition                           |
|      |        | 2 .MCPSO  Page off-line (nonexistent)                   |
|      |        | 3 .MCPSSE  Page off-line due to error                   |
| 3    | .MCSPS | Set status of specified page                           |

100
TOPS-20 Monitor Calls Quick Reference Guide

PMCTL

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

4  .MCRME  Return information about MOS memory errors
   0  .PMMTP  <1B81<count>B17>,<controller #>
   1  .PMMRG  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  JSYS 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  JSYS 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  .PRARD  Return arguments in argblk
2  .PRAST  Set arguments from argblk

ARGUMENT BLOCK

101
TOPS-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></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>B7-12</td>
<td>Type code</td>
</tr>
<tr>
<td></td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>02</td>
</tr>
<tr>
<td></td>
<td>04</td>
</tr>
<tr>
<td></td>
<td>17</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%EMD 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%DOM No match was found for string
B4 RC%AMB String given was ambiguous
B5 RC%NMD 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 (iBn 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%STP  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
      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

RCVDK%  JSYS 575

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

RESTRICTIONS
Requires enabled WHEEL or OPERATOR capability.
CALLING SEQUENCE
AC1: Address of argb1k
AC2: Length of argb1k

RETURNS +1: Always

ARGUMENT BLOCK (RET)
Word Symbol Contents
0 .RCFCJ <GETOK% function code>,<job # of requestor>
1 .RCUNO User number
2 .RCCDR Connected directory
3 .RCRNQ Request number
4 .RCNUA BO-17 # arguments passed to RCVOK% block
   B18-35 # user arguments in user block
5 .RCARA Address of user arguments
6 .RCCAPCapabilities enabled
7 .RCTER Controlling terminal number; or -1 for
   detached job
10 .RCRJB Requested job #
11 User arguments
11+D ..

RDTTY 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>
   O, <# of bytes in string> to break on EDL only
0, <# of bytes in string> to break on CTRL/Z or ESC
B0(RD%BRK) 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
B9(RD%BEG) 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

RFCOC  JSYS 112

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

CALLING SEQUENCE
AC1: File designator

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

RFORK

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

108
TOPS-20 Monitor Calls Quick Reference Guide

RFSTS

FUNCTION
Returns the status of the specified process.

CALLING SEQUENCE
AC1: O,,<process handle> (short form)
     flags,,<process handle> (long form)
BO    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
BO   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  .RFLNT  Process dismissed by HFORK or
                  HALTF or never started
      3  .RFFTP  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  .RPFSW  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%EX0  Process is execute-only
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

<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 and time file was last written</td>
</tr>
<tr>
<td>1</td>
<td>.RSCRV</td>
<td>Internal date and time file was created</td>
</tr>
<tr>
<td>2</td>
<td>.RSREF</td>
<td>Internal date and time file was last referenced</td>
</tr>
<tr>
<td>3</td>
<td>.RSERE</td>
<td>System date and time of last write by monitor</td>
</tr>
<tr>
<td>4</td>
<td>.RSTDT</td>
<td>Tape-write date and time for archived or migrated files</td>
</tr>
<tr>
<td>5</td>
<td>.RSNET</td>
<td>Online expiration date and time</td>
</tr>
<tr>
<td>6</td>
<td>.RSFET</td>
<td>Offline expiration date and time</td>
</tr>
</tbody>
</table>

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-J in AC2; 0 if EOF

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
1st pointer
B23(P1%PEX) Page exists in 1st pointer
B27(P1%CPY) Copy-on-write in 1st pointer

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 0 if rescan buffer empty

FUNCTION CODES
Code Symbol Meaning
0 .RSINI Make rescan buffer available for input
1 .RSCNT Return count of characters remaining in rescan buffer

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>
TOPS-20 Monitor Calls Quick Reference Guide

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(SM%RD) Read access allowed
B3(SM%WR) Write access allowed
B4(SM%EX) Execute access allowed
B5(PA%PEX) Section exists
B6(SM%IND) Section created using
indirect pointer

RTFRK JSYS 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: JSYS instruction that caused process
suspension

RTIW JSYS 173

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

AC1: BO(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
**RUNTM** _JSYS 15_

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)

**RWM** _JSYS 135_

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)

**RWSET** _JSYS 176_

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

RETURNS  +1: Always

**SACTF** _JSYS 62_

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

RESTRICTIONS
In non-zero sections, OWGBPs must specify 7-bit bytes.
CALLING SEQUENCE
AC1:  JFN
AC2:  <5B21<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>
         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
Code Symbol   Meaning
  0 .SCRET   Return designator of given process’s
             controlling terminal in AC2
  1 .SCSET   Change given process’s (and inferiors)
             controlling terminal to terminal designated
in AC2 (SC%SCT)
Reset given process’s (and inferiors) controlling terminal to job’s controlling terminal (SC%SCT)

**SCVEC** JSYS 301

**FUNCTION**
Sets the entry vector and the UU0 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 UU0 simulation
AC3: <UU0 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 UU0s</td>
</tr>
<tr>
<td>1</td>
<td>SVINE</td>
<td>Initial entry for setup and first UU0</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 UU0 call</td>
</tr>
<tr>
<td>4</td>
<td>SVRPC</td>
<td>Address to receive return PC word on UU0 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
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 .SDDMS Address in which to store RMS call
4 .SDRPC Address in which to store return PC word

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

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>.SJDEN</td>
<td>Set default magtape density</td>
</tr>
<tr>
<td>1</td>
<td>.SJPAR</td>
<td>Set default for magtape parity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SJPRO Odd parity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SJPRE Even parity</td>
</tr>
<tr>
<td>2</td>
<td>.SJDM</td>
<td>Set default for magtape data mode</td>
</tr>
<tr>
<td>3</td>
<td>.SJRS</td>
<td>Set default for magtape record size (in bytes)</td>
</tr>
<tr>
<td>4</td>
<td>.SJDFS</td>
<td>Set spooling mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SJSPI Immediate mode spooling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SJSPD Deferred mode spooling</td>
</tr>
<tr>
<td>5</td>
<td>.SJSRM</td>
<td>Set remark for current job session; pointer to remark in AC3</td>
</tr>
<tr>
<td>6</td>
<td>.SJ2TO</td>
<td>Indicate if job is at EXEC or program level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 Job is at EXEC level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 Job is at program level</td>
</tr>
<tr>
<td>7</td>
<td>.SJDFR</td>
<td>Set job default retrieval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SJRFA OPENF of off-line disk file should fail (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SJRSA OPENF of off-line disk file should wait for restoral</td>
</tr>
<tr>
<td>10</td>
<td>.SJBAT</td>
<td>Set batch flags and batch stream number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BO-1(0B%WTO) Write to operator capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .OBALL WTO &amp; WTOR allowed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .OBNWR No WTO allowed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 .OBN0M No message allowed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B10(0B%BSS) 0B%BSN contains batch stream #</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B11-17(0B%BSS) Batch stream #</td>
</tr>
<tr>
<td>11</td>
<td>.SJLLO</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
SEVEC  JSYS 204

FUNCTION
Sets the entry vector of the specified process.

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

CALLING SEQUENCE
AC1:  Process handle
AC2:  <entry vector length>,<entry vector address>; or 0 to remove entry vector

RETURNS  +1:  Always

SFACS  JSYS 160

FUNCTION
Sets the ACs of the specified process.

CALLING SEQUENCE
AC1:  Process handle
AC2:  Address of 20 word block containing new AC values for process

RETURNS  +1:  Always

SFBSZ  JSYS 46

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

CALLING SEQUENCE
AC1:  JFN
AC2:  Byte size, right-justified

RETURNS  +1:  Failure, error code in AC1
          +2:  success
SFCOC  JSYS 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  JSYS 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  JSYS 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 argbk
AC3:  Length of argbk
RETURNS  +1:  Always

ARGUMENT BLOCK
Word  Symbol  Meaning
0  .RSWRT  Internal date/time file was last written
          (enabled WHL/OPR for >current)
1  .RSCRV  Internal date and time file was created
          (enabled WHL/OPR for >current)
2  .RSREF  Internal date and time file was last read
          (enabled WHL/OPR for >current)
3  .RSCRE  System date and time of last write by monitor
          (enabled WHL/OPR)
4  .RSTDT  Tape-write date and time of archived or
          migrated files (enabled WHL/OPR)
5  .RSNET  On-line expiration date and time (date/time
          or interval)
6  .RSFET  Offline expiration date and time (date/time
          or interval)

SFUST  JSYS 551

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
Code  Symbol  Meaning
0  .SFAUT  Set name of author of file
1  .SFLWR  Set name of user who last wrote file (enabled
          WHL/OPR)

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

**SIN** JSYS 52

**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-J) 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-J) on which to terminate input (if +\(n\) in AC3)

**RETURNS** +1: Always, with
AC1: Updated byte pointer
AC2: Updated byte pointer
AC3: 0 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;
IDX10 returned and unread bytes discarded

SIR JSYS 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

SIRCM JSYS 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 JSYS 36

FUNCTION
Returns the length of an existing file.

CALLING SEQUENCE
AC1: JFN
RETURNS +1: Failure, error code in AC1
+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
BO-17 Percentage of CPU resources (1 - 99%) guaranteed to job; 0 for no request
B18 System job flag (JP%SYS); 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

SKED% 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

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.SKRBC</td>
<td>Read bias control knob setting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SACNT Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SAKB Bias control knob setting (RET)</td>
</tr>
<tr>
<td>2</td>
<td>.SKSBC</td>
<td>Set bias control setting (WHL/OPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .SACNT Word count including this word</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 .SAKB Bias control setting (1-20)</td>
</tr>
</tbody>
</table>

126
3 .SKRCS
Read class parameters
0 .SACNT Word count including this word
1 .SAOIS Class of job (RET)
2 .SAOIR Share of CPU allocated to class (RET; 0.0<=n.n<=1.0)
3 .SAUSE Amount of CPU used by class (RET; 0.0<=n.n<=1.0)
4 .SA1ML 1 min load avg for class (RET)
5 .SA5ML 5 min load avg for class (RET)
6 .SA1ML 15 min load avg for class (RET)

4 .SKSCS
Set class parameters (WHL/OPR)
0 .SACNT Word count including this word
1 .SAOIS Class of job
2 .SAOIR Share of CPU allocated to class (0.0<=n.n<=1.0)

5 .SKICS
Start or stop class scheduler (WHL/OPR)
0 .SACNT Word count including this word
1 .SACTL Control flags
   BO(SK%ACT) Class by accounts
   B1(SK%WDF) Withhold windfall
   B2(SK%STP) Class scheduler off

6 .SKSCJ
Set job class (WHL/OPR for other than calling job)
0 .SACNT Word count including this word
1 .SAOJB Job #; -1 for calling job
2 .SAOCL Class of job
3 .SAWA Windfall allocation

7 .SKRJP
Read class parameters for a job
0 .SACNT Word count including this word
1 .SAUSH Job's share of CPU (RET; 0.0<=n.n<=1.0)
2 .SAUUS Job's current CPU use (RET; 0.0<=n.n<=1.0)

10 .SKBCR
Read class setting for batch jobs
0 .SACNT Word count including this word
1 .SABCL Batch class; -1 if none (RET)

11 .SKBCS
Set batch class (WHL/OPR)
0 .SACNT Word count including this word
1 .SABCL Batch class; -1 for none

12 .SKBBG
Run all batch jobs in "dregs" queue; illegal if class scheduling in use (WHL/OPR)
0 .SACNT Word count including this word
1 .SADRG 0 don't run in dregs queue
          #0 run in dregs queue

13 .SKDDC
Reserved

14 .SKRCV
Read status
0 .SACNT Word count including this word
1 .SACTL Flags
   BO(SK%ACT) Class by accounts
   B1(SK%WDF) Withhold windfall
   B2(SK%STP) Class scheduler off
   B3(SK%DRG) Batch jobs being run in dregs queue

127
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)
     <# (1-37) of contiguous sections to map> (Cases I-III)
     B2(SM%RD) Allow read access (Cases I-III)
     B3(SM%WR) Allow write access (Cases I-III)
     B4(SM%EX) Allow execute access (Cases I-III)
     B6(SM%IND) Map using indirect section pointer (Case II-III)

RETURNS
+1: Always

SMON JSYS 6

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

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

CALLING SEQUENCE
AC1: Function code
AC2: New value for function
<table>
<thead>
<tr>
<th>Function Codes</th>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
<th>AC2</th>
<th>AC1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>.SFFAC</td>
<td>Allow FACT file entries</td>
<td>1(SF%FAC) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>.SFCDE</td>
<td>CHECKD found errors</td>
<td>1(SF%CDE) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>.SFCDR</td>
<td>CHECKD is running</td>
<td>1(SF%CDR) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>.SFMST</td>
<td>Manual start in progress</td>
<td>1(SF%MST) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>.SFRMT</td>
<td>Allow remote LOGINS</td>
<td>1(SF%RMT) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>.SFPTY</td>
<td>Allow PTY LOGINS</td>
<td>1(SF%PTY) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>.SFCTY</td>
<td>Allow CTY LOGINS</td>
<td>1(SF%CTY) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>.SFOPR</td>
<td>Operator in attendance</td>
<td>1(SF%OPR) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>.SFLCL</td>
<td>Allow local LOGINS</td>
<td>1(SF%LCL) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>.SFBTE</td>
<td>Bit table errors found on startup</td>
<td>1(SF%BTE) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>.SFCRD</td>
<td>Users can change directory parameters</td>
<td>1(SF%CRD) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>.SFNVT</td>
<td>Allow ARPA local LOGINS (ARPA)</td>
<td>1(SF%NVT) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>.SFUSG</td>
<td>Allow USAGE file entries</td>
<td>1(SF%USG) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>.SFFLO</td>
<td>Set full disk latency optimization (requires Kl10-E revision level 2 and RH20 board M8555 revision level D)</td>
<td>1(SF%FLO) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>.SFMTA</td>
<td>Enable MOUNTR magtape allocation</td>
<td>1(SF%MTA) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>.SFMSG</td>
<td>Set system message level 0</td>
<td>1(SF%MSG) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>.SFMS1</td>
<td>Set system message level 1</td>
<td>1(SF%MS1) to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>.SFXTN</td>
<td>Turn ARPANET on (ARPA)</td>
<td>1 to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>.SFNDU</td>
<td>Reinitialize ARPANET if down (ARPA)</td>
<td>1 to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>.SFNHI</td>
<td>Initialize ARPANET host table (ARPA)</td>
<td>1 to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>.SFTMZ</td>
<td>Set local time zone</td>
<td>time zone</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>.SFLHN</td>
<td>Set local ARPANET host number (ARPA)</td>
<td>ARPANET host number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>.SFAVR</td>
<td>Enable account validation</td>
<td>1 to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>.SFSTS</td>
<td>Enable status reporting</td>
<td>1 to set; 0 to clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>.SFSOK</td>
<td>Set GETOK% defaults</td>
<td>flags,,&lt;GETOK% function code&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BO(SF%EOK) 0 to disable access checking
        1 to enable access checking
B1(SF%DOK) 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 .SFRLU 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%UUT) 1 to unload unrecognizable tapes
           0 to treat unrecognizable tapes as unlabeled
61 .SFWSR 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: BO 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 OPERATCR 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>.SNPIB</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>.SNPUL</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

SOBE JSYS 103

SOBF JSYS 175

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

SOUT JSYS 53

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-J) 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
FUNCTION

SOUTR  JSYS 532

Writes a variable-length record to 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.

CALLING SEQUENCE

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

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

FUNCTION

SPACS  JSYS 60

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

FUNCTION

SPJFN  JSYS 207

Sets the primary JFNs (.PRIIN and .PRIOUT) 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
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

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>.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>1</td>
<td>.SPLSD</td>
<td>Set directory of spooled device (enabled WHL/DPR)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.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>.SPLDV Designator of spooled device</td>
</tr>
</tbody>
</table>
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 (JP%SYS); 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

SAVE  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%UCA) 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.

RESTRICITIONS
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
Compared two ASCIZ strings.

RESTRICITIONS
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  JYS 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  JYS 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(ST%DIM) Set deferred terminal interrupt mask
given in AC3
        B18-35(ST%PRH) 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
STO

STO  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

**STTPY** _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>.SWART</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>.SWRA</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 LUUO block address for non-zero sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: LUUO address; 0 to clear</td>
</tr>
<tr>
<td>3</td>
<td>.SWRLT</td>
<td>Read LUUO block address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AC3: LUUO address; 0 if none set (RET)</td>
</tr>
</tbody>
</table>
LUU0 BLOCK FORMAT
Offset 0 12 13 17 18 26 27 30 31 35

.ARPFL(0) PC flags | O | opcode | AC | O |
.AROP(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
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%AMB) 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 1B2

COMMAND TABLE FORMAT
Word Contents
0  # of remaining words, <max # of remaining words>
1...n address of argblk, <available to user>

ARGUMENT BLOCK
Word Contents
0 If 0(BO-6) and 1B7(CM%FW), <flags>B18-35 and string begins in next word; if -0(BO-6) or 0B7, string starts in this word
B34(CM%NDR) Do not recognize this string
Start of string if $0(B0-6)$ and $1B7(CM\%FW)$ in word 0

**TEXTI** **JSYS 524**

**FUNCTION**
Reads input from a terminal or a file.

**CALLING SEQUENCE**
AC1: Address of argblk

**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 argblk

**ARGUMENT BLOCK**

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.RDCWB</td>
<td>Word count not including this word</td>
</tr>
<tr>
<td>1</td>
<td>.RDFLG</td>
<td>Flags</td>
</tr>
<tr>
<td></td>
<td>B0(RD%BRK)</td>
<td>Break on CTRL/Z or ESC</td>
</tr>
<tr>
<td></td>
<td>B1(RD%TOP)</td>
<td>Break on CTRL/G, CTRL/K, CTRL/L, CTRL/Z, ESC, CR, LF</td>
</tr>
<tr>
<td></td>
<td>B2(RD%PUN)</td>
<td>Break on punctuation</td>
</tr>
<tr>
<td></td>
<td>B3(RD%BEL)</td>
<td>Break on EOL (CRLF or LF only)</td>
</tr>
<tr>
<td></td>
<td>B4(RD%CRF)</td>
<td>Suppress CR and return LF only</td>
</tr>
<tr>
<td></td>
<td>B5(RD%RND)</td>
<td>Return if user tries to delete past beginning of buffer UFNs in word .RDIOJ</td>
</tr>
<tr>
<td></td>
<td>B6(RD%JFN)</td>
<td>Return if input buffer empty</td>
</tr>
<tr>
<td></td>
<td>B7(RD%RIE)</td>
<td>Not used</td>
</tr>
<tr>
<td></td>
<td>B8(RD%BBG)</td>
<td>Return when .RDBKL pointer is reached</td>
</tr>
<tr>
<td></td>
<td>B9(RD%BE)</td>
<td>Convert lowercase input to UPPERCASE</td>
</tr>
<tr>
<td></td>
<td>B10(RD%RAI)</td>
<td>Suppress CTRL/U indication</td>
</tr>
<tr>
<td></td>
<td>B11(RD%SUI)</td>
<td>Break character terminated input (RET)</td>
</tr>
<tr>
<td></td>
<td>B12(RD%BTM)</td>
<td>Returned because user tried to delete past beginning of buffer (RET)</td>
</tr>
<tr>
<td></td>
<td>B13(RD%BF)</td>
<td>Backup limit for editing reached (RET)</td>
</tr>
<tr>
<td></td>
<td>B14(RD%BLR)</td>
<td>Byte pointer to string; or &lt;input JFN&gt;, &lt;output JFN&gt;</td>
</tr>
<tr>
<td></td>
<td>.RDIOJ</td>
<td>Byte pointer to destination string buffer</td>
</tr>
<tr>
<td></td>
<td>.RDDBP</td>
<td>Number of bytes in destination string</td>
</tr>
<tr>
<td></td>
<td>.RDDBC</td>
<td>Byte pointer to beginning of destination buffer</td>
</tr>
<tr>
<td></td>
<td>.RDBFP</td>
<td>Byte pointer to beginning of CTRL/R buffer</td>
</tr>
<tr>
<td></td>
<td>.RDRTY</td>
<td>Address of 4-word break character mask block</td>
</tr>
<tr>
<td></td>
<td>.RDBRK</td>
<td>Byte pointer to backup limit in destination buffer</td>
</tr>
<tr>
<td></td>
<td>.RDBKL</td>
<td>Byte pointer to backup limit in destination buffer</td>
</tr>
</tbody>
</table>
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
Code Symbol  Function
0  .TFSET  Set JSYS traps for given process (illegal for execute-only processes)
1  .TFRAL  Remove all JSYS traps for given process (illegal for execute-only processes)
2  .TFRTP  Remove JSYS traps indicated in monitor call bit table for given process (illegal for execute-only processes)
3  .TFSPS  Set interrupts on given software channel
4  .TFRPS  Return interrupt channel in left half of AC2
5  .TFTST  Test if caller is to be intercepted when it attempts to execute monitor calls;
          On return
          AC2: -1 intercept; 0 no intercept
6  .TFRES  Remove intercepts for all inferiors and clear assigned software channels
7  .TFUUD  Set JSYS traps for TOPS-10 UUOs for given process (illegal for execute-only processes)
10  .TFSJU  Set JSYS traps for both TOPS-10 UUOs indicated in monitor call bit table (illegal for execute-only processes)
11  .TFRUU  Remove JSYS traps for TOPS-10 UUOs

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>

144
TOPS-20 Monitor Calls Quick Reference Guide

TLINK JSYS 216

FUNCTION
Controls terminal linking.

REstrictions
Some functions require enabled WHEEL or OPERATOR capability.

calling sequence
AC1: B0(TL%CRO)
   Clear link from remote to object
designator
B1(TL%COR)
   Clear link from object to remote
designator
B2(TL%EOR)
   Establish link from object to remote
designator
B3(TL%ERO)
   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

TMOn JSYS 7

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
Code Symbol Function
0 .SFFAC FACT file entries are allowed
1 .SCFDE CHECKD found errors
2 .SFCDR CHECKD is running
3 .SFMDST Manual start is in progress
4 .SFRMT Remote LOGINS are allowed
5 .SFPTY PTY LOGINS are allowed
6 .SFCTY CTY LOGINS are allowed
7 .SFOPR Operator is in attendance
10 .SFLCL Local LOGINs are allowed
11 .SBTE Bit table errors found on startup
12 .SFCDR Users can change nonprivileged directory parameters
13 .SFNVT ARPANET terminal LOGINs are allowed
21 .SFUSG USAGE file entries are allowed
22 .SFFLO Disk latency optimization using RH20 backup register is enabled
23 .SFMTA MOUNTR magtape 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 .SFTMZ Local time zone
50 .SFLHN ARPANET local host number
51 .SFAVR Account validation is running
52 .SFSTS Status reporting is enabled
53 .SFSOK GETOK% defaults

AC2: flags,<GETOK% function code>
   BO(SF%EOK) 0 Access checking is disabled
   1 Access checking is enabled
   B1(SF%DOK) 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 .SFRDU Update last access read time for directories
56 .SFACY Maximum offline expiration period in days for archive files
57 .SFRTW File-retrieval requests should fail
60 .SFTDF Tape mount controls
   BO(MT%UUT) 1 unload unrecognizable tapes
   0 treat unrecognizable tapes as unlabeled
61 .SFWSR 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>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>4</td>
<td>.USLGD</td>
<td>Terminate checkpoint entry for job and write entry to system's data file</td>
</tr>
<tr>
<td>5</td>
<td>.USSEN</td>
<td>Terminate current session, write entry to system's data file, and initialize new checkpoint entry for job</td>
</tr>
<tr>
<td>6</td>
<td>.USCKI</td>
<td>Set checkpoint time interval</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>12</td>
<td>.USRAS</td>
<td>Read accounting shifts</td>
</tr>
</tbody>
</table>

AC2: Pointer to argblk of format: 0 <# table entries>, <max # entries> 1-0 B0-6 (US%DDW) Days-of-week entry in effect (0=Monday) B7-17 Not used, must be zero B16-35 Time in seconds since midnight for accounting shift change

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 IDT flag set
**TOPS-20 Monitor Calls Quick Reference Guide**

**UTES**  
**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**

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.UTSET</td>
<td>Start testing code</td>
</tr>
<tr>
<td>1</td>
<td>.UTCLR</td>
<td>Stop testing code and update bit map in argblk</td>
</tr>
</tbody>
</table>

**ARGUMENT BLOCK**

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.UTADR</td>
<td>Address of beginning of code section to be tested</td>
</tr>
<tr>
<td>1</td>
<td>.UTLEN</td>
<td>Length of code section to be tested</td>
</tr>
<tr>
<td>2</td>
<td>.UTMAP</td>
<td>Start of bit map representing instructions to be tested in code section</td>
</tr>
</tbody>
</table>

**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>  
    BO(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

149
RETURNS +1: Always, with updated pointer in AC2

FUNCTION
Dismisses the current process indefinitely and does not return.

WAIT JSYS 306

FUNCTION
Causes the current process to wait for an inferior process to terminate.

WFORk JSYS 163

CALLING SEQUENCE
AC1: Inferior process handle

RETURNS +1: Always, when specified processes terminates

WILD% JSYS 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: BO(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>BO(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>
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% JSYS 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: BO-5 Flags
B6-35 Address of entry vector

XGTPW% JSYS 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 On return
Flags
BO(PF%USR) Page failure on user-mode reference
B1(PF%WTF) Page failure on write reference
2 0 On return
Address that referenced page
3 0 On return
MUU0 opcode and AC
4 0 On return
30-bit effective address of MUU0
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

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

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**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2</td>
<td>RM%RD</td>
<td>Read access allowed</td>
</tr>
<tr>
<td>B3</td>
<td>RM%WR</td>
<td>Write access allowed</td>
</tr>
<tr>
<td>B4</td>
<td>RM%EX</td>
<td>Execute access allowed</td>
</tr>
<tr>
<td>B5</td>
<td>RM%PEX</td>
<td>Page exists</td>
</tr>
<tr>
<td>B9</td>
<td>RM%CPY</td>
<td>Copy-on-write access allowed</td>
</tr>
</tbody>
</table>

**FUNCTION**

Starts the specified process in a non-zero section of memory.

**CALLING SEQUENCE**

AC1: Flags,,<process handle>
     BO(SF%CON) Continue process that has halted

AC2: <PC flags>,0

AC3: Address to set PC to (ignored if SC%CON on)

**RETURNS** +1: Always

**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
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%EV)  Extended entry vector; if on, entry vector points to 2-word extended PC and extended format UUO word
       B6-35  Address of entry vector

RETURNS  +1:  Always

XSVEC%  JSYS 6C7

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
### CONTROL CHARACTER OUTPUT CONTROL (CCOC) WORD

<table>
<thead>
<tr>
<th>ASCII Code</th>
<th>Wake-up Class</th>
<th>CCOC Word</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>SPACE</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>P</td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>P</td>
<td>&quot;</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></td>
</tr>
<tr>
<td>46</td>
<td>P</td>
<td>&amp;</td>
<td></td>
</tr>
<tr>
<td>47</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></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>60-71</td>
<td>A</td>
<td>O-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>
TDPS-20 Monitor Calls Quick Reference Guide
CCDC Word

| P | 75 | = |
| P | 76 | > |
| P | 77 | ? |
| @ | 100 | |
| A | 101-132 | UPPERCASE LETTERS A-Z |
| [ | 133 |
| \ | 134 | ] |
| ^ | 135 | |
| _ | 136 | |
| accent grave | 140 |
| A | 141-172 | LOWERCASE LETTERS a-z |
| { | 173(1) |
| | 174(1) | } |
| | 175(1) | |
| ~ | 176(1) |
| DELETE (RUBOUT) | 177 |

| A | Alphanumeric character | 0(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>RSX2OF protocol</td>
</tr>
<tr>
<td>1</td>
<td>.VNM CB</td>
<td>MCB DECnet protocol</td>
</tr>
<tr>
<td>2</td>
<td>.VND6O</td>
<td>DN60 (IBMCOM) protocol</td>
</tr>
<tr>
<td></td>
<td>.VND DC</td>
<td>DDCMP (DECnet) protocol</td>
</tr>
<tr>
<td>3</td>
<td>.VNMDP</td>
<td>MOP (DDCMP 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>spooled 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>spooled 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>.DVCDR</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>.DVTY</td>
<td>yes</td>
</tr>
<tr>
<td>PTTY:</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>spooled 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>.DVSRV</td>
<td>no</td>
</tr>
</tbody>
</table>

Device designator = <600000(.DVDES)+type>,,<unit number>
<600000(.DVDES)+type>,,-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>DP%RD</td>
<td>Directory access controlled by individual file access</td>
</tr>
<tr>
<td>10</td>
<td>DP%CN</td>
<td>Connecting to directory and changing protection/account allowed</td>
</tr>
<tr>
<td>4</td>
<td>DP%CF</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>FP%RD</td>
<td>Read access</td>
</tr>
<tr>
<td>20</td>
<td>FP%WR</td>
<td>Write access</td>
</tr>
<tr>
<td>10</td>
<td>FP%EX</td>
<td>Execute access</td>
</tr>
<tr>
<td>4</td>
<td>FP%APP</td>
<td>Append access</td>
</tr>
<tr>
<td>2</td>
<td>FP%DIR</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>1</td>
<td>.FBCTL</td>
<td><strong>Reserved for DIGITAL</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0-28: 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></td>
<td></td>
<td>B0(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%NX): 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%NOD): 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%SDR): Directory has subdirectories</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B11(FB%ARC): 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%FCF): File class field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>O(.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: 5000000,&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>.FBGEN</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>.FBBYV</td>
<td>File I/O flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B0-5(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%MOD): 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>.FBCRV</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>.FREF</td>
<td>Date/time of last non-write access</td>
</tr>
<tr>
<td>16</td>
<td>.FBCNT</td>
<td>&lt;# of file writes&gt;,&lt;# of file references&gt;</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>.FBBBBT</td>
<td>Flags,&lt;# file pages when deleted&gt;</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>

158
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 .FBGNL  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 saveset #>,<1st tape file #>
35 .FBTP2  Tape ID for second archive or collection run.
36 .FBSS2  <2nd tape saveset #>,<2nd tape file #>

FORK (PROCESS) HANDLES

<table>
<thead>
<tr>
<th>Value</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>400000</td>
<td>.FHSLF</td>
<td>Current process</td>
</tr>
<tr>
<td>400000+n</td>
<td></td>
<td>Process n, inferior to current process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(relative fork handle)</td>
</tr>
<tr>
<td>-1</td>
<td>.FHSUP</td>
<td>Superior process</td>
</tr>
<tr>
<td>-2</td>
<td>.FHTOP</td>
<td>Top-level process</td>
</tr>
<tr>
<td>-3</td>
<td>.FHSAI</td>
<td>Current process and all inferiors</td>
</tr>
<tr>
<td>-4</td>
<td>.FHINF</td>
<td>All inferiors of current process</td>
</tr>
<tr>
<td>-5</td>
<td>.FHJOB</td>
<td>All processes in job</td>
</tr>
</tbody>
</table>

FLOATING-POINT FORMAT CONTROL

<table>
<thead>
<tr>
<th>Bit</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0-1</td>
<td>FL%SGN</td>
<td>Sign control for 1st field; 1st character position used for minus for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>negative numbers; for positive numbers, 1st character</td>
</tr>
<tr>
<td></td>
<td></td>
<td>position defined according to:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 .FLDIG 1st character is digit</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
Floating-Point Format Control

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2-3</td>
<td>FLJUS</td>
</tr>
<tr>
<td>0</td>
<td>.FLLSP Right justify with leading spaces</td>
</tr>
<tr>
<td>1</td>
<td>.FLLZR Right justify with leading 0's</td>
</tr>
<tr>
<td>2</td>
<td>.FLLAS Right justify with leading asterisks</td>
</tr>
<tr>
<td>3</td>
<td>.FLTSP Left justify with trailing spaces</td>
</tr>
<tr>
<td>B4</td>
<td>FLONE</td>
</tr>
<tr>
<td>B5</td>
<td>FLDOL</td>
</tr>
<tr>
<td>B6</td>
<td>FLPNT</td>
</tr>
<tr>
<td>0-8</td>
<td>FLEXP</td>
</tr>
<tr>
<td>0</td>
<td>.FLEXN No exponent field</td>
</tr>
<tr>
<td>1</td>
<td>.FLEXE Output E as 1st character of exponent field</td>
</tr>
<tr>
<td>2</td>
<td>.FLEXD Output D as 1st character of exponent field</td>
</tr>
<tr>
<td>3</td>
<td>.FLEXM Output *10^ as 1st characters of exponent field</td>
</tr>
<tr>
<td>B9-10</td>
<td>FLESG</td>
</tr>
<tr>
<td>0</td>
<td>.FLDGE 1st character after exponent prefix is digit</td>
</tr>
<tr>
<td>1</td>
<td>.FLPLE 1st character after prefix is plus sign</td>
</tr>
<tr>
<td>2</td>
<td>.FLSPE 1st character after prefix is space</td>
</tr>
<tr>
<td>3</td>
<td>.FLDGT 1st character after exponent prefix is digit</td>
</tr>
<tr>
<td>B11</td>
<td>FLDOVL</td>
</tr>
<tr>
<td>B13-17</td>
<td>FLRND</td>
</tr>
<tr>
<td>B18-23</td>
<td>FLFST</td>
</tr>
<tr>
<td>B24-29</td>
<td>FLSEND</td>
</tr>
<tr>
<td>B30-35</td>
<td>FLTHD</td>
</tr>
</tbody>
</table>

**I/O IDENTIFIERS**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>LH,RH</td>
<td>Job File Number (file handle)</td>
</tr>
<tr>
<td>.JFN</td>
<td>Primary input designator</td>
</tr>
<tr>
<td>.JDN</td>
<td>Primary output designator</td>
</tr>
<tr>
<td>.ULID</td>
<td>Null designator</td>
</tr>
<tr>
<td>.UTDES</td>
<td>Universal terminal designator</td>
</tr>
<tr>
<td>.CTRM</td>
<td>Process's controlling terminal</td>
</tr>
<tr>
<td>.DVDES</td>
<td>Universal device designator</td>
</tr>
<tr>
<td>600000,xxxx</td>
<td>Implicit byte pointer</td>
</tr>
<tr>
<td>777777,,address</td>
<td>Universal default</td>
</tr>
</tbody>
</table>
### 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%MFF</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></td>
<td></td>
<td>B18 not used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B19 TT%IGN Ignore other TT%WAK bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B20 TT%WKF Formatting control character</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B21 TT%WKU Non-formatting control character</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B22 TT%WKP Punctuation character</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B23 TT%WKA 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></td>
<td></td>
<td>00 .TTBIN No translation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01 .TTASC Translate both echo and output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 .TTATO Translate output only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 .TTATE Translate echo only</td>
</tr>
<tr>
<td>B30</td>
<td>TT%UOC</td>
<td>Upper case output control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 Do not indicate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Indicate by ‘X</td>
</tr>
<tr>
<td>B31</td>
<td>TT%LIC</td>
<td>Lower case input control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 No conversion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Convert lower to upper</td>
</tr>
<tr>
<td>B32-33</td>
<td>TT%DUM</td>
<td>Duplex mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 .TTFDX Full duplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01 Reserved for DIGITAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 .TTHDX Character half duplex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 .TTLDX Line half duplex</td>
</tr>
<tr>
<td>B34</td>
<td>TT%PGM</td>
<td>Pause-on-command mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 Disable pause-on-command mode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 Enable pause-on-command mode</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%GTB</td>
<td>Process can examine monitor tables with GETAB</td>
</tr>
<tr>
<td>B3</td>
<td>SC%LOG</td>
<td>Process can execute protected log functions</td>
</tr>
<tr>
<td>B6</td>
<td>SC%SCT</td>
<td>Process can change source of terminal interrupts for other processes</td>
</tr>
</tbody>
</table>
B-9-17 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  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 %ENO  User has ENQ/DEQ capability
B24  SC %NWZ  User has NET WIZARD (ARPAnet) capability
B25  SC %NAS  User has ARPA NET 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>.SUDDM6</td>
<td>SIXBIT mode</td>
</tr>
<tr>
<td>3</td>
<td>.SUDDM4</td>
<td>ANSI ASCII mode</td>
</tr>
<tr>
<td>4</td>
<td>.SUDDM8</td>
<td>Industry compatible mode</td>
</tr>
<tr>
<td>5</td>
<td>.SUDDM9</td>
<td>High-density mode (TU70, TU72 only)</td>
</tr>
</tbody>
</table>
TOPS-20 Monitor Calls Quick Reference Guide
Magtape Label States

MA GT APE 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>

MA GT APE 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>

MA GT APE RECORD SIZES

- Data Mode   Maximum Record Size
  - System-default: --
  - Dump: 8192 bytes
  - SIXBIT: 49152 bytes
  - ANSI ASCII: 40960 bytes
  - Industry compatible: 32768 bytes
  - High density: 8192 bytes

MA GT APE 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>.SJFN2</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>MO%FER</td>
<td>Fatal error condition</td>
</tr>
<tr>
<td>B12</td>
<td>MO%EDF</td>
<td>All pending output has been processed</td>
</tr>
<tr>
<td>B13</td>
<td>MO%IOP</td>
<td>Output 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%DL</td>
<td>Card-punch off-line</td>
</tr>
<tr>
<td>B17</td>
<td>MO%FNX</td>
<td>Card punch doesn’t exist</td>
</tr>
<tr>
<td>B32</td>
<td>MO%HEM</td>
<td>Stacker full or hooper empty</td>
</tr>
<tr>
<td>B33</td>
<td>MO%SCK</td>
<td>Stack check</td>
</tr>
</tbody>
</table>
### PHYSICAL CARD READER (PCDR:) STATUS BITS

<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%EDF</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%SCK</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>8</td>
<td>HT (^I)</td>
<td>Skips to beginning of every 8th column on same line</td>
</tr>
<tr>
<td></td>
<td></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>
### 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>MO%LCP</td>
<td>Lower case printer</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>All data sent to printer has been printed</td>
</tr>
<tr>
<td>B13</td>
<td>MO%IDP</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>B30</td>
<td>MO%RPE</td>
<td>RAM parity error</td>
</tr>
<tr>
<td>B31</td>
<td>MO%LVU</td>
<td>Optical VFU</td>
</tr>
<tr>
<td>B33</td>
<td>MO%LVF</td>
<td>VFU error</td>
</tr>
<tr>
<td>B34</td>
<td>MO%LCI</td>
<td>Character interrupt</td>
</tr>
<tr>
<td>B35</td>
<td>MO%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>B18</td>
<td>MT%ILW</td>
<td>Drive is write protected</td>
</tr>
<tr>
<td>B19</td>
<td>MT%DVE</td>
<td>Device error (hung or data late)</td>
</tr>
<tr>
<td>B20</td>
<td>MT%DAE</td>
<td>Data error</td>
</tr>
<tr>
<td>B21</td>
<td>MT%SER</td>
<td>Suppress automatic error recovery procedures</td>
</tr>
<tr>
<td>B22</td>
<td>MT%EOF</td>
<td>Device EOF (file) mark</td>
</tr>
<tr>
<td>B23</td>
<td>MT%IRL</td>
<td>Incorrect record length</td>
</tr>
<tr>
<td>B24</td>
<td>MT%BOT</td>
<td>Beginning of tape</td>
</tr>
<tr>
<td>B25</td>
<td>MT%EOT</td>
<td>End of tape</td>
</tr>
<tr>
<td>B26</td>
<td>MT%EVP</td>
<td>Even parity</td>
</tr>
<tr>
<td>B29-31</td>
<td>MT%CCT</td>
<td>Character counter if MT%IRL on</td>
</tr>
<tr>
<td>B32</td>
<td>MT%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>.GRNR</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>.GSIMG</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></td>
<td>Assignable by user program</td>
</tr>
<tr>
<td>6</td>
<td>.ICAOV</td>
<td>Arithmetic overflow (includes NODIV)</td>
</tr>
<tr>
<td>7</td>
<td>.ICFOV</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;INFO</td>
</tr>
<tr>
<td>2</td>
<td>.SPQSR</td>
<td>PID of QUASAR</td>
</tr>
<tr>
<td>3</td>
<td>.SPMDA</td>
<td>PID of QSRMDA</td>
</tr>
<tr>
<td>4</td>
<td>.SPOP</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>BLDTD</td>
<td></td>
<td>Date and time system was generated</td>
</tr>
<tr>
<td>DEBUGSW</td>
<td>0</td>
<td>State of operator coverage</td>
</tr>
<tr>
<td></td>
<td>1</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; or -2 if reserved for device allocator</td>
</tr>
<tr>
<td>DRMERR</td>
<td>0</td>
<td>Information on drum errors</td>
</tr>
<tr>
<td></td>
<td>1 to n</td>
<td>Number of recoverable errors</td>
</tr>
<tr>
<td>DSKERR</td>
<td>0</td>
<td>Information on disk errors</td>
</tr>
<tr>
<td></td>
<td>1 to n</td>
<td>Number of recoverable disk errors</td>
</tr>
<tr>
<td>DWNTIM</td>
<td>0</td>
<td>Downtime information</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>Date/time of next scheduled system shutdown</td>
</tr>
<tr>
<td>HQLAV</td>
<td></td>
<td>High queue load averages</td>
</tr>
<tr>
<td>IMPLT1</td>
<td>i(P2)</td>
<td>ARPANET: 1 fullword for each link; -1 if control link; or internal connection index for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETAWD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETBAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETBTC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETBUF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETFSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETLSK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETSTS</td>
</tr>
<tr>
<td></td>
<td>B18-19</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>B20-27</td>
<td>Host number</td>
</tr>
<tr>
<td></td>
<td>B28-35</td>
<td>Link number</td>
</tr>
<tr>
<td>IMPLT2</td>
<td>i(P2)</td>
<td>ARPANET: 1 fullword for each link; Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B10-17 Byte size of buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Address of input buffer</td>
</tr>
<tr>
<td>IMPLT3</td>
<td>i(P2)</td>
<td>ARPANET: 1 fullword for each link; Address of output buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Message saved for retransmission</td>
</tr>
<tr>
<td>IMPLT4</td>
<td>i(P2)</td>
<td>ARPANET: 1 fullword for each link; Address of current buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B18-35 Message allocation in bits</td>
</tr>
<tr>
<td>JBONT</td>
<td>Job #</td>
<td>Owning job for CRJOB-created jobs</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>Job #</td>
<td>BO-17</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>JOBPNM</td>
<td>Job #</td>
<td>B18-35</td>
</tr>
<tr>
<td>JOBRT</td>
<td>Job #</td>
<td>SIXBIT</td>
</tr>
<tr>
<td>JOBTY</td>
<td>Job #</td>
<td>CPU time used by job (negative if no such job)</td>
</tr>
<tr>
<td>LOGDES</td>
<td></td>
<td>BO-17</td>
</tr>
<tr>
<td>LOGDES</td>
<td></td>
<td>B18-35</td>
</tr>
<tr>
<td>LOGDES</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>LOGDES</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>LOGDES</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>LQ lava</td>
<td></td>
<td>Low queue load averages</td>
</tr>
<tr>
<td>NETHST</td>
<td>i(P2)</td>
<td>ARPANET: 1 full word for each internal connection; -1 if no foreign host connection</td>
</tr>
<tr>
<td>NETAWD</td>
<td>i(P2)</td>
<td>ARPANET: 1 full word for each internal connection</td>
</tr>
<tr>
<td>NETBAL</td>
<td>i(P2)</td>
<td>ARPANET: number of bits allocated to each internal connection</td>
</tr>
<tr>
<td>NETBTC</td>
<td>i(P2)</td>
<td>ARPANET: byte count statistics</td>
</tr>
<tr>
<td>NETBUF</td>
<td>i(P2)</td>
<td>ARPANET: 1 fullword for each internal connection</td>
</tr>
<tr>
<td>NETFSK</td>
<td>i(P2)</td>
<td>ARPANET: foreign socket number (32 bits) for each internal connection</td>
</tr>
<tr>
<td>NETLSK</td>
<td>i(P2)</td>
<td>ARPANET: local socket number for each internal connection</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>ARPANET: operational status table</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>&gt;0</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>-1</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>NETRDY</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>BO-15</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>B16-17</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>B18-21</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>B22-31</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>NCPGS</td>
<td></td>
<td>Number of pages of real (physical) user core available in system (1 word)</td>
</tr>
<tr>
<td>NSWPNGS</td>
<td></td>
<td>Default swapping pages</td>
</tr>
<tr>
<td>PTYPAR</td>
<td>Pseudo-TTY parameter information</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>B0-17 Number of PTYs in system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B18-35 TTY number of first PTY</td>
<td></td>
</tr>
<tr>
<td>QTIMES</td>
<td>Accumulated runtime of jobs on n scheduler queues</td>
<td></td>
</tr>
<tr>
<td>0 to n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNAME</td>
<td>SIXBIT name of system program; 0 if entry unused</td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNBLKS</td>
<td>Number of samples in working set size</td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td>integral</td>
<td></td>
</tr>
<tr>
<td>SPFLTS</td>
<td>Total number of page faults of system</td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td>program</td>
<td></td>
</tr>
<tr>
<td>SSIZE</td>
<td>Time integral of working set size</td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STIMES</td>
<td>Total runtime of system program</td>
<td></td>
</tr>
<tr>
<td>(P3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYMTAB</td>
<td>SIXBIT table names of all GETAB tables</td>
<td></td>
</tr>
<tr>
<td>SYSTAT</td>
<td>Monitor statistics</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Time with no runnable jobs</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Waiting time with 1 or more runnable jobs</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Time spent in scheduler</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Time spent processing pager traps</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Number of drum reads</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Number of drum writes</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Number of disk reads</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Number of disk writes</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Number of terminal wakeups</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Number of terminal interrupts</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Time integral of number of processes in balance set</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Time integral of number of runnable processes</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Exponential 1-minute average of number of runnable processes</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Exponential 5-minute average of number of runnable processes</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Exponential 15-minute average of number of runnable processes</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Time integral of number of processes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>waiting for disk</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Time integral of number of processes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>waiting for drum</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Number of terminal input characters</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Number of terminal output characters</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Number of system core management cycles</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Time spent doing postpurging</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Number of forced balance set process</td>
<td></td>
</tr>
<tr>
<td></td>
<td>removals</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Time integral of number of processes in swap wait</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Scheduler overhead time in high precision units</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Idle time in high precision units</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Lost time in high precision units</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>User time</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Time integral of number of processes on high queue</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Time integral of number of processes on low queue</td>
<td></td>
</tr>
</tbody>
</table>

169
TOPS-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
TTYJOB line # BO-17 Job # for which this is controlling terminal;
                  -1 for unassigned line;
                  -2 for line currently being assigned; or
                  job # to which line is assigned
B18-35 -1 if no process is waiting for input from terminal
        # -1 if some process is waiting for input

(Pn) specifies a set of parallel tables where n is a unique identifier of the set
i specifies an index into a table derived from B24-35 of NETAWD
j specifies an index into a table derived from BO-17 of IMPLT1

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>
<tr>
<td></td>
<td>Description</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------</td>
<td>--------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TTY model 37</td>
<td>.TT37</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TI/EXECUPORT</td>
<td>.TTEXE</td>
<td></td>
</tr>
<tr>
<td>4-7</td>
<td>Reserved for customer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Default</td>
<td>.TTDEF</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Ideal</td>
<td>.TTIDL</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>VT05</td>
<td>.TTV05</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>VT50</td>
<td>.TTV50</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>LA30</td>
<td>.TTL30</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>GT40</td>
<td>.TTG40</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>LA36</td>
<td>.TTL36</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>VT52</td>
<td>.TTV52</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>VT100</td>
<td>.TT100</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>LA38</td>
<td>.TTL38</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>LA120</td>
<td>.TT120</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>VT125</td>
<td>.TT125</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>VK100</td>
<td>.TTK10</td>
<td></td>
</tr>
</tbody>
</table>

Width 72, page length 66. Lowercase; padding after TAB and FF; page width 72, page length 66.

Lowercase; padding after CR only; page width 80, page length 66.

Reserved for customer.

Lowercase; full padding; page width 72, page length 66.

Mechanical FF and TAB; lowercase; no padding; no specified width or length.

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.

UPPERCASE only; full padding; page width 80, page length 66.

Lowercase; no padding; page width 80, page length 30.

Lowercase; no padding; page width 132, page length 66.

Mechanical TAB; lowercase; no padding; page width 80, page length 24.

Mechanical TAB; lowercase; no padding; page width 80, page length 24; cursor control.

Mechanical TAB; lowercase; no padding; page width 80, page length 24; cursor control; graphics capabilities.

Mechanical TAB; lowercase; no padding; page width 80, page length 24; cursor control; color graphics capabilities.
### Terminal Characteristics

<table>
<thead>
<tr>
<th>Character/Condition</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRL/@ or BREAK</td>
<td>0</td>
</tr>
<tr>
<td>CTRL/A</td>
<td>1</td>
</tr>
<tr>
<td>CTRL/B</td>
<td>2</td>
</tr>
<tr>
<td>CTRL/C</td>
<td>3</td>
</tr>
<tr>
<td>CTRL/D</td>
<td>4</td>
</tr>
<tr>
<td>CTRL/E</td>
<td>5</td>
</tr>
<tr>
<td>CTRL/F</td>
<td>6</td>
</tr>
<tr>
<td>CTRL/G</td>
<td>7</td>
</tr>
<tr>
<td>CTRL/H</td>
<td>8</td>
</tr>
<tr>
<td>CTRL/I (TAB)</td>
<td>9</td>
</tr>
<tr>
<td>CTRL/J (LF)</td>
<td>10</td>
</tr>
<tr>
<td>CTRL/K (vertical TAB)</td>
<td>11</td>
</tr>
<tr>
<td>CTRL/L (FF)</td>
<td>12</td>
</tr>
<tr>
<td>CTRL/M (CR)</td>
<td>13</td>
</tr>
<tr>
<td>CTRL/N</td>
<td>14</td>
</tr>
<tr>
<td>CTRL/O</td>
<td>15</td>
</tr>
<tr>
<td>CTRL/P</td>
<td>16</td>
</tr>
<tr>
<td>CTRL/Q</td>
<td>17</td>
</tr>
<tr>
<td>CTRL/R</td>
<td>18</td>
</tr>
<tr>
<td>CTRL/S</td>
<td>19</td>
</tr>
<tr>
<td>CTRL/T</td>
<td>20</td>
</tr>
<tr>
<td>CTRL/U</td>
<td>21</td>
</tr>
<tr>
<td>CTRL/V</td>
<td>22</td>
</tr>
<tr>
<td>CTRL/W</td>
<td>23</td>
</tr>
<tr>
<td>CTRL/X</td>
<td>24</td>
</tr>
<tr>
<td>CTRL/Y</td>
<td>25</td>
</tr>
<tr>
<td>CTRL/Z</td>
<td>26</td>
</tr>
<tr>
<td>ESCAPE (altmode)</td>
<td>27</td>
</tr>
<tr>
<td>DELETE (RUBOUT)</td>
<td>28</td>
</tr>
<tr>
<td>SPACE</td>
<td>29</td>
</tr>
<tr>
<td>Dataset carrier off</td>
<td>30</td>
</tr>
<tr>
<td>Typein</td>
<td>31</td>
</tr>
<tr>
<td>Typeout</td>
<td>32</td>
</tr>
<tr>
<td>Reserved for DIGITAL</td>
<td>33-35</td>
</tr>
</tbody>
</table>
### TIME ZONES

<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>HDT</td>
<td>700012</td>
</tr>
<tr>
<td>ALASKA-HAWAII STANDARD TIME</td>
<td>HST</td>
<td>500012</td>
</tr>
<tr>
<td>BERING DAYLIGHT TIME</td>
<td>BDT</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>
## 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>GJFX1</td>
</tr>
<tr>
<td>600056</td>
<td>GJFX2</td>
<td>600057</td>
<td>GJFX3</td>
<td>600060</td>
<td>GJFX4</td>
</tr>
<tr>
<td>600061</td>
<td>GJFX5</td>
<td>600062</td>
<td>GJFX6</td>
<td>600063</td>
<td>GJFX7</td>
</tr>
<tr>
<td>600064</td>
<td>GJFX8</td>
<td>600065</td>
<td>GJFX9</td>
<td>600066</td>
<td>GJFX10</td>
</tr>
<tr>
<td>600067</td>
<td>GJFX11</td>
<td>600070</td>
<td>GJFX12</td>
<td>600071</td>
<td>GJFX13</td>
</tr>
<tr>
<td>600072</td>
<td>GJFX14</td>
<td>600073</td>
<td>GJFX15</td>
<td>600074</td>
<td>GJFX16</td>
</tr>
<tr>
<td>600075</td>
<td>GJFX17</td>
<td>600076</td>
<td>GJFX18</td>
<td>600077</td>
<td>GJFX19</td>
</tr>
<tr>
<td>600100</td>
<td>GJFX20</td>
<td>600101</td>
<td>GJFX21</td>
<td>600102</td>
<td>GJFX22</td>
</tr>
<tr>
<td>600103</td>
<td>GJFX23</td>
<td>600104</td>
<td>GJFX24</td>
<td>600107</td>
<td>GJFX27</td>
</tr>
<tr>
<td>600110</td>
<td>GJFX28</td>
<td>600112</td>
<td>GJFX30</td>
<td>600113</td>
<td>GJFX31</td>
</tr>
<tr>
<td>600114</td>
<td>GJFX32</td>
<td>600115</td>
<td>GJFX33</td>
<td>600116</td>
<td>GJFX34</td>
</tr>
<tr>
<td>600117</td>
<td>GJFX35</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>RJFNX1</td>
</tr>
<tr>
<td>600166</td>
<td>RJFNX2</td>
<td>600167</td>
<td>RJFNX3</td>
<td>600170</td>
<td>DELFX1</td>
</tr>
<tr>
<td>600175</td>
<td>SFTPX1</td>
<td>600176</td>
<td>SFTPX2</td>
<td>600177</td>
<td>SFTPX3</td>
</tr>
<tr>
<td>600200</td>
<td>CNIDX1</td>
<td>600204</td>
<td>CNIDX5</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>GTAXB1</td>
<td>600270</td>
<td>GTAXB2</td>
</tr>
<tr>
<td>600271</td>
<td>GTAXB3</td>
<td>600273</td>
<td>RUNTX1</td>
<td>600275</td>
<td>STADX1</td>
</tr>
<tr>
<td>600276</td>
<td>STADX2</td>
<td>600300</td>
<td>ASNDX1</td>
<td>600301</td>
<td>ASNDX2</td>
</tr>
<tr>
<td>600302</td>
<td>ASNDX3</td>
<td>600320</td>
<td>ATACX1</td>
<td>600321</td>
<td>ATACX2</td>
</tr>
<tr>
<td>600322</td>
<td>ATACX3</td>
<td>600323</td>
<td>ATACX4</td>
<td>600324</td>
<td>ATACX5</td>
</tr>
<tr>
<td>600332</td>
<td>STDVX1</td>
<td>600335</td>
<td>DEVX1</td>
<td>600336</td>
<td>DEVX2</td>
</tr>
<tr>
<td>600337</td>
<td>DEVX3</td>
<td>600350</td>
<td>TERMX1</td>
<td>600351</td>
<td>TLNXX1</td>
</tr>
<tr>
<td>600352</td>
<td>ATIX1</td>
<td>600353</td>
<td>ATIX2</td>
<td>600356</td>
<td>TLNXX2</td>
</tr>
<tr>
<td>600357</td>
<td>TLNXX3</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>600366</td>
<td>KFRXX2</td>
<td>600370</td>
<td>HFRHX1</td>
<td>600371</td>
<td>GFRXX1</td>
</tr>
<tr>
<td>600373</td>
<td>GETX1</td>
<td>600374</td>
<td>GETX2</td>
<td>600375</td>
<td>TFRXX1</td>
</tr>
<tr>
<td>600376</td>
<td>TFRXX2</td>
<td>600377</td>
<td>SFRVX1</td>
<td>600407</td>
<td>NOUTX1</td>
</tr>
<tr>
<td>600410</td>
<td>NOUTX2</td>
<td>600411</td>
<td>TFRXX3</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>GFDBX1</td>
</tr>
<tr>
<td>600425</td>
<td>GFDBX2</td>
<td>600426</td>
<td>GFDBX3</td>
<td>600430</td>
<td>CFDBX1</td>
</tr>
<tr>
<td>600431</td>
<td>CFDBX2</td>
<td>600432</td>
<td>CFDBX3</td>
<td>600433</td>
<td>CFDBX4</td>
</tr>
<tr>
<td>600440</td>
<td>DUMPX1</td>
<td>600441</td>
<td>DUMPX2</td>
<td>600442</td>
<td>DUMPX3</td>
</tr>
<tr>
<td>600443</td>
<td>DUMPX4</td>
<td>600450</td>
<td>RNAMX1</td>
<td>600451</td>
<td>RNAMX2</td>
</tr>
<tr>
<td>600452</td>
<td>RNAMX3</td>
<td>600453</td>
<td>RNAMX4</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>DTDNX1</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>600516</td>
<td>SMONX1</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>FFUFX1</td>
<td>600545</td>
<td>FFUFX2</td>
<td>600546</td>
<td>FFUFX3</td>
</tr>
<tr>
<td>600555</td>
<td>DSMX1</td>
<td>600570</td>
<td>SIRX1</td>
<td>600580</td>
<td>SSAVX1</td>
</tr>
<tr>
<td>600601</td>
<td>SSAVX2</td>
<td>600610</td>
<td>SEVEX1</td>
<td>600614</td>
<td>WHELX1</td>
</tr>
<tr>
<td>600615</td>
<td>CAPX1</td>
<td>600617</td>
<td>PEEKX2</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>FLINX3</td>
<td>600653</td>
<td>FLINX4</td>
<td>600660</td>
<td>FLOTX1</td>
</tr>
<tr>
<td>600661</td>
<td>FLOTX2</td>
<td>600705</td>
<td>GTHSX2</td>
<td>600707</td>
<td>GTHSX3</td>
</tr>
<tr>
<td>600704</td>
<td>GTHSX1</td>
<td>600711</td>
<td>ATNX2</td>
<td>600712</td>
<td>ATNX3</td>
</tr>
<tr>
<td>600710</td>
<td>ATNX1</td>
<td>600714</td>
<td>ATNX5</td>
<td>600715</td>
<td>ATNX6</td>
</tr>
<tr>
<td>600713</td>
<td>ATNX4</td>
<td>600717</td>
<td>ATNX8</td>
<td>600720</td>
<td>ATNX9</td>
</tr>
<tr>
<td>600716</td>
<td>ATNX7</td>
<td>600722</td>
<td>ATNX11</td>
<td>600723</td>
<td>ATNX12</td>
</tr>
<tr>
<td>600721</td>
<td>ATNX10</td>
<td>600727</td>
<td>CVHST1</td>
<td>600730</td>
<td>CVSXX1</td>
</tr>
<tr>
<td>600724</td>
<td>ATNX13</td>
<td>600732</td>
<td>SNDIX1</td>
<td>600733</td>
<td>SNDIX2</td>
</tr>
<tr>
<td>600731</td>
<td>CVSKX2</td>
<td>600735</td>
<td>SNDIX4</td>
<td>600736</td>
<td>SNDIX5</td>
</tr>
<tr>
<td>600734</td>
<td>SNDIX3</td>
<td>600740</td>
<td>ASNSX1</td>
<td>600741</td>
<td>ASNSX2</td>
</tr>
<tr>
<td>600737</td>
<td>NTWZX1</td>
<td>600743</td>
<td>SQX2</td>
<td>600746</td>
<td>GTNCX1</td>
</tr>
<tr>
<td>600742</td>
<td>SQX1</td>
<td>600750</td>
<td>RNAMX5</td>
<td>600751</td>
<td>RNAMX6</td>
</tr>
<tr>
<td>600747</td>
<td>GTNCX2</td>
<td>600753</td>
<td>RNAMX8</td>
<td>600754</td>
<td>RNAMX9</td>
</tr>
<tr>
<td>600752</td>
<td>RNAMX7</td>
<td>600756</td>
<td>RNMX11</td>
<td>600757</td>
<td>RNMX12</td>
</tr>
<tr>
<td>600755</td>
<td>RNMX10</td>
<td>600770</td>
<td>ILINS1</td>
<td>600771</td>
<td>ILINS2</td>
</tr>
<tr>
<td>600760</td>
<td>GJFX36</td>
<td>601000</td>
<td>CRLNX1</td>
<td>601001</td>
<td>INLNX1</td>
</tr>
<tr>
<td>600772</td>
<td>ILINS3</td>
<td>601010</td>
<td>RDTX1</td>
<td>601011</td>
<td>GFKXSX1</td>
</tr>
<tr>
<td>601002</td>
<td>LNSTX1</td>
<td>601014</td>
<td>GTJIX2</td>
<td>601015</td>
<td>GTJIX3</td>
</tr>
<tr>
<td>601013</td>
<td>GTJIX1</td>
<td>601017</td>
<td>IPCFX2</td>
<td>601020</td>
<td>IPCFX3</td>
</tr>
<tr>
<td>601016</td>
<td>IPCFX1</td>
<td>601022</td>
<td>IPCFX5</td>
<td>601023</td>
<td>IPCFX6</td>
</tr>
<tr>
<td>601021</td>
<td>IPCFX4</td>
<td>601025</td>
<td>IPCFX8</td>
<td>601026</td>
<td>IPCFX9</td>
</tr>
<tr>
<td>601024</td>
<td>IPCFX7</td>
<td>601030</td>
<td>IPCF11</td>
<td>601031</td>
<td>IPCF12</td>
</tr>
<tr>
<td>601027</td>
<td>IPCF10</td>
<td>601033</td>
<td>IPCF14</td>
<td>601034</td>
<td>IPCF15</td>
</tr>
<tr>
<td>601032</td>
<td>IPCF13</td>
<td>601036</td>
<td>IPCF17</td>
<td>601037</td>
<td>IPCF18</td>
</tr>
<tr>
<td>601035</td>
<td>IPCF16</td>
<td>601041</td>
<td>IPCF20</td>
<td>601042</td>
<td>IPCF21</td>
</tr>
<tr>
<td>601040</td>
<td>IPCF19</td>
<td>601044</td>
<td>IPCF23</td>
<td>601045</td>
<td>IPCF24</td>
</tr>
<tr>
<td>601043</td>
<td>IPCF22</td>
<td>601047</td>
<td>IPCF26</td>
<td>601050</td>
<td>IPCF27</td>
</tr>
<tr>
<td>601046</td>
<td>IPCF25</td>
<td>601052</td>
<td>IPCF29</td>
<td>601053</td>
<td>IPCF30</td>
</tr>
<tr>
<td>601051</td>
<td>IPCF28</td>
<td>601055</td>
<td>ENQX1</td>
<td>601056</td>
<td>ENQX2</td>
</tr>
<tr>
<td>601054</td>
<td>GNJFX1</td>
<td>601060</td>
<td>ENQX4</td>
<td>601061</td>
<td>ENQX5</td>
</tr>
<tr>
<td>601057</td>
<td>ENQX3</td>
<td>601063</td>
<td>ENQX7</td>
<td>601064</td>
<td>ENQX8</td>
</tr>
<tr>
<td>601062</td>
<td>ENQX6</td>
<td>601066</td>
<td>ENQX10</td>
<td>601067</td>
<td>ENQX11</td>
</tr>
</tbody>
</table>

175
<table>
<thead>
<tr>
<th>Code</th>
<th>Code</th>
<th>Code</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>601070</td>
<td>ENQX12</td>
<td>601071</td>
<td>ENQX13</td>
</tr>
<tr>
<td>601073</td>
<td>ENQX15</td>
<td>601074</td>
<td>ENQX16</td>
</tr>
<tr>
<td>601076</td>
<td>ENQX18</td>
<td>601077</td>
<td>ENQX19</td>
</tr>
<tr>
<td>601101</td>
<td>ENQX21</td>
<td>601102</td>
<td>IPCF31</td>
</tr>
<tr>
<td>601104</td>
<td>PMAPX3</td>
<td>601105</td>
<td>PMAPX4</td>
</tr>
<tr>
<td>601107</td>
<td>PMAPX6</td>
<td>601110</td>
<td>SNOPX1</td>
</tr>
<tr>
<td>601112</td>
<td>SNOPX3</td>
<td>601113</td>
<td>SNOPX4</td>
</tr>
<tr>
<td>601115</td>
<td>SNOPX6</td>
<td>601116</td>
<td>SNOPX7</td>
</tr>
<tr>
<td>601120</td>
<td>SNOPX9</td>
<td>601121</td>
<td>SNOP10</td>
</tr>
<tr>
<td>601123</td>
<td>SNOP12</td>
<td>601124</td>
<td>SNOP13</td>
</tr>
<tr>
<td>601126</td>
<td>SNOP15</td>
<td>601127</td>
<td>SNOP16</td>
</tr>
<tr>
<td>601131</td>
<td>SNOP17</td>
<td>601132</td>
<td>OPNX23</td>
</tr>
<tr>
<td>601134</td>
<td>CRLNX2</td>
<td>601135</td>
<td>INLNX2</td>
</tr>
<tr>
<td>601137</td>
<td>ALCX1</td>
<td>601140</td>
<td>ALCX2</td>
</tr>
<tr>
<td>601142</td>
<td>ALCX4</td>
<td>601143</td>
<td>ALCX5</td>
</tr>
<tr>
<td>601145</td>
<td>SPLX2</td>
<td>601146</td>
<td>SPLX3</td>
</tr>
<tr>
<td>601150</td>
<td>SPLX5</td>
<td>601151</td>
<td>CLSX3</td>
</tr>
<tr>
<td>601153</td>
<td>ALCX6</td>
<td>601154</td>
<td>CKAX1</td>
</tr>
<tr>
<td>601156</td>
<td>CKAX3</td>
<td>601157</td>
<td>TIMX1</td>
</tr>
<tr>
<td>601161</td>
<td>TIMX3</td>
<td>601162</td>
<td>TIMX4</td>
</tr>
<tr>
<td>601164</td>
<td>GJFX38</td>
<td>601165</td>
<td>GJFX39</td>
</tr>
<tr>
<td>601167</td>
<td>CRDIX9</td>
<td>601170</td>
<td>CRDI10</td>
</tr>
<tr>
<td>601172</td>
<td>DELDX2</td>
<td>601173</td>
<td>GACTX3</td>
</tr>
<tr>
<td>601175</td>
<td>DIAGX2</td>
<td>601176</td>
<td>DIAGX3</td>
</tr>
<tr>
<td>601200</td>
<td>DIAGX5</td>
<td>601201</td>
<td>DIAGX6</td>
</tr>
<tr>
<td>601203</td>
<td>DIAGX8</td>
<td>601204</td>
<td>DIAGX9</td>
</tr>
<tr>
<td>601206</td>
<td>SYEX1</td>
<td>601207</td>
<td>SYEX2</td>
</tr>
<tr>
<td>601211</td>
<td>IDX7</td>
<td>601212</td>
<td>IDX8</td>
</tr>
<tr>
<td>601214</td>
<td>DUMPX5</td>
<td>601215</td>
<td>DUMPX6</td>
</tr>
<tr>
<td>601217</td>
<td>CLSX4</td>
<td>601220</td>
<td>MTOX2</td>
</tr>
<tr>
<td>601222</td>
<td>MTOX4</td>
<td>601223</td>
<td>MTOX6</td>
</tr>
<tr>
<td>601225</td>
<td>GJFX40</td>
<td>601226</td>
<td>MTOX7</td>
</tr>
<tr>
<td>601230</td>
<td>LOUTX4</td>
<td>601231</td>
<td>CAPX2</td>
</tr>
<tr>
<td>601233</td>
<td>SAVX4</td>
<td>601234</td>
<td>TDELX1</td>
</tr>
<tr>
<td>601236</td>
<td>TADDX2</td>
<td>601237</td>
<td>TLUXK1</td>
</tr>
<tr>
<td>601244</td>
<td>SJBX1</td>
<td>601245</td>
<td>SJBX2</td>
</tr>
<tr>
<td>601247</td>
<td>TMONX1</td>
<td>601250</td>
<td>SMONX2</td>
</tr>
<tr>
<td>601252</td>
<td>SJBX5</td>
<td>601253</td>
<td>SJBX6</td>
</tr>
<tr>
<td>601255</td>
<td>ILINS4</td>
<td>601256</td>
<td>ILINS5</td>
</tr>
<tr>
<td>601260</td>
<td>COMNX2</td>
<td>601261</td>
<td>COMNX3</td>
</tr>
<tr>
<td>601264</td>
<td>PRAX2</td>
<td>601265</td>
<td>COMNX5</td>
</tr>
<tr>
<td>601271</td>
<td>CKAX4</td>
<td>601272</td>
<td>GACXX1</td>
</tr>
<tr>
<td>601274</td>
<td>MTOX8</td>
<td>601275</td>
<td>DRKX1</td>
</tr>
<tr>
<td>601277</td>
<td>GJFX41</td>
<td>601300</td>
<td>GJFX42</td>
</tr>
<tr>
<td>601302</td>
<td>TEMEX2</td>
<td>601303</td>
<td>DELFX2</td>
</tr>
<tr>
<td>601305</td>
<td>DELFX4</td>
<td>601306</td>
<td>DELFX5</td>
</tr>
<tr>
<td>601310</td>
<td>DELFX7</td>
<td>601311</td>
<td>DELFX8</td>
</tr>
<tr>
<td>601313</td>
<td>DIRX1</td>
<td>601314</td>
<td>DIRX2</td>
</tr>
<tr>
<td>601316</td>
<td>UFPGX1</td>
<td>601317</td>
<td>LNGFX1</td>
</tr>
<tr>
<td>601321</td>
<td>COMNX8</td>
<td>601322</td>
<td>MTOX9</td>
</tr>
<tr>
<td>601324</td>
<td>MTOX11</td>
<td>601325</td>
<td>MTOX12</td>
</tr>
<tr>
<td>601327</td>
<td>MTOX14</td>
<td>601330</td>
<td>SAVX1</td>
</tr>
<tr>
<td>601332</td>
<td>MTOX16</td>
<td>601333</td>
<td>LPINX1</td>
</tr>
<tr>
<td>601335</td>
<td>LPINX3</td>
<td>601336</td>
<td>MTOX17</td>
</tr>
<tr>
<td>601340</td>
<td>DESX9</td>
<td>601341</td>
<td>ACESX1</td>
</tr>
<tr>
<td>601344</td>
<td>DSKOX2</td>
<td>601345</td>
<td>MSTRX1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>601347</td>
<td>MSTRX3</td>
<td>601350</td>
<td>MSTRX4</td>
</tr>
<tr>
<td>601352</td>
<td>MSTRX6</td>
<td>601353</td>
<td>MSTRX7</td>
</tr>
<tr>
<td>601355</td>
<td>MSTRX9</td>
<td>601356</td>
<td>MSTRX10</td>
</tr>
<tr>
<td>601360</td>
<td>MSTRX12</td>
<td>601361</td>
<td>MSTRX13</td>
</tr>
<tr>
<td>601363</td>
<td>MSTRX15</td>
<td>601364</td>
<td>MSTRX16</td>
</tr>
<tr>
<td>601367</td>
<td>DSKX03</td>
<td>601371</td>
<td>GFUX1</td>
</tr>
<tr>
<td>601373</td>
<td>SFUX1</td>
<td>601374</td>
<td>SFUX2</td>
</tr>
<tr>
<td>601377</td>
<td>RCDX2</td>
<td>601400</td>
<td>RCDX3</td>
</tr>
<tr>
<td>601402</td>
<td>RCUSX1</td>
<td>601403</td>
<td>TEDLX2</td>
</tr>
<tr>
<td>601405</td>
<td>LSTRX1</td>
<td>601406</td>
<td>SWJFX1</td>
</tr>
<tr>
<td>601410</td>
<td>DPNX26</td>
<td>601411</td>
<td>DELFX9</td>
</tr>
<tr>
<td>601413</td>
<td>COMNX9</td>
<td>601414</td>
<td>STYPX1</td>
</tr>
<tr>
<td>601416</td>
<td>DSKOX3</td>
<td>601417</td>
<td>DESX10</td>
</tr>
<tr>
<td>601421</td>
<td>MSTRX17</td>
<td>601422</td>
<td>MSTRX18</td>
</tr>
<tr>
<td>601424</td>
<td>MSTRX20</td>
<td>601425</td>
<td>MSTRX21</td>
</tr>
<tr>
<td>601427</td>
<td>CRDI11</td>
<td>601430</td>
<td>MSTRX23</td>
</tr>
<tr>
<td>601432</td>
<td>ACESX4</td>
<td>601433</td>
<td>ACESX5</td>
</tr>
<tr>
<td>601436</td>
<td>STRX01</td>
<td>601437</td>
<td>STRX02</td>
</tr>
<tr>
<td>601441</td>
<td>IDX12</td>
<td>601442</td>
<td>STRX03</td>
</tr>
<tr>
<td>601444</td>
<td>PPNX1</td>
<td>601445</td>
<td>PPNX2</td>
</tr>
<tr>
<td>601451</td>
<td>CRDI12</td>
<td>601452</td>
<td>GFUX3</td>
</tr>
<tr>
<td>601454</td>
<td>RNMX13</td>
<td>601455</td>
<td>SJBX8</td>
</tr>
<tr>
<td>601460</td>
<td>WILDX1</td>
<td>601461</td>
<td>MSTRX41</td>
</tr>
<tr>
<td>601475</td>
<td>LCBDBP</td>
<td>601477</td>
<td>LCNDXD</td>
</tr>
<tr>
<td>601502</td>
<td>ATACX6</td>
<td>601503</td>
<td>ATACX7</td>
</tr>
<tr>
<td>601537</td>
<td>TIMX8</td>
<td>601535</td>
<td>TIMX6</td>
</tr>
<tr>
<td>601550</td>
<td>SCTX1</td>
<td>601551</td>
<td>SCTX2</td>
</tr>
<tr>
<td>601553</td>
<td>SCTX4</td>
<td>601554</td>
<td>PDVX01</td>
</tr>
<tr>
<td>601556</td>
<td>PDVX03</td>
<td>601557</td>
<td>GETX4</td>
</tr>
<tr>
<td>601700</td>
<td>SFUX4</td>
<td>601701</td>
<td>SFUX5</td>
</tr>
<tr>
<td>601703</td>
<td>GETX3</td>
<td>601706</td>
<td>CAPX3</td>
</tr>
<tr>
<td>601715</td>
<td>ARGX04</td>
<td>601716</td>
<td>ARGX05</td>
</tr>
<tr>
<td>601720</td>
<td>ARGX07</td>
<td>601721</td>
<td>ARGX08</td>
</tr>
<tr>
<td>601723</td>
<td>ARGX10</td>
<td>601725</td>
<td>ARGX12</td>
</tr>
<tr>
<td>601727</td>
<td>MONX01</td>
<td>601730</td>
<td>MONX02</td>
</tr>
<tr>
<td>601732</td>
<td>MONX04</td>
<td>601733</td>
<td>ARGX14</td>
</tr>
<tr>
<td>601741</td>
<td>ARGX16</td>
<td>601742</td>
<td>ARGX17</td>
</tr>
<tr>
<td>601744</td>
<td>DEVX5</td>
<td>601747</td>
<td>STRX06</td>
</tr>
<tr>
<td>601751</td>
<td>MSTRX25</td>
<td>601752</td>
<td>MSTRX26</td>
</tr>
<tr>
<td>601754</td>
<td>GJFX43</td>
<td>601755</td>
<td>MTDX19</td>
</tr>
<tr>
<td>601757</td>
<td>MSTRX27</td>
<td>601760</td>
<td>MSTRX28</td>
</tr>
<tr>
<td>601763</td>
<td>DSKX05</td>
<td>601764</td>
<td>DSKX06</td>
</tr>
<tr>
<td>601766</td>
<td>DSKX08</td>
<td>601767</td>
<td>COMX10</td>
</tr>
<tr>
<td>601771</td>
<td>LOCKX1</td>
<td>601772</td>
<td>LOCKX2</td>
</tr>
<tr>
<td>601775</td>
<td>ILLX02</td>
<td>601776</td>
<td>ILLX03</td>
</tr>
<tr>
<td>602000</td>
<td>MSTRX31</td>
<td>602001</td>
<td>MSTRX32</td>
</tr>
<tr>
<td>602003</td>
<td>STDI1X</td>
<td>602004</td>
<td>CNDX17</td>
</tr>
<tr>
<td>602006</td>
<td>PMCLX2</td>
<td>602007</td>
<td>PMCLX3</td>
</tr>
<tr>
<td>602011</td>
<td>DLFX11</td>
<td>602012</td>
<td>GJFX44</td>
</tr>
<tr>
<td>602014</td>
<td>UTSTX2</td>
<td>602015</td>
<td>UTSTX3</td>
</tr>
<tr>
<td>602017</td>
<td>BDTOX2</td>
<td>602020</td>
<td>DCNX1</td>
</tr>
<tr>
<td>602022</td>
<td>DCNX3</td>
<td>602023</td>
<td>DCNX4</td>
</tr>
<tr>
<td>602025</td>
<td>DCNX8</td>
<td>602026</td>
<td>DCNX11</td>
</tr>
<tr>
<td>602030</td>
<td>TTYX01</td>
<td>602031</td>
<td>BDTOX3</td>
</tr>
<tr>
<td>602033</td>
<td>ARGX19</td>
<td>602035</td>
<td>COMX11</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>602037</td>
<td>COMX13</td>
<td>602040</td>
<td>COMX14</td>
</tr>
<tr>
<td>602042</td>
<td>COMX16</td>
<td>602043</td>
<td>COMX17</td>
</tr>
<tr>
<td>602045</td>
<td>NPXNSW</td>
<td>602046</td>
<td>NPXNOM</td>
</tr>
<tr>
<td>602050</td>
<td>NPXINW</td>
<td>602051</td>
<td>NPXNC</td>
</tr>
<tr>
<td>602053</td>
<td>NPXIDT</td>
<td>602054</td>
<td>NPXNQS</td>
</tr>
<tr>
<td>602056</td>
<td>NPXNMD</td>
<td>602057</td>
<td>NPXCMA</td>
</tr>
<tr>
<td>602061</td>
<td>GJFX46</td>
<td>602062</td>
<td>GJFX47</td>
</tr>
<tr>
<td>602064</td>
<td>GJFX48</td>
<td>602065</td>
<td>GJFX49</td>
</tr>
<tr>
<td>602100</td>
<td>DELF10</td>
<td>602101</td>
<td>CRDI13</td>
</tr>
<tr>
<td>602103</td>
<td>CRDI15</td>
<td>602104</td>
<td>CRDI16</td>
</tr>
<tr>
<td>602106</td>
<td>ENACX2</td>
<td>602107</td>
<td>ENACX3</td>
</tr>
<tr>
<td>602111</td>
<td>VACCXO</td>
<td>602112</td>
<td>VACCX1</td>
</tr>
<tr>
<td>602114</td>
<td>BOTX04</td>
<td>602116</td>
<td>USGX02</td>
</tr>
<tr>
<td>602120</td>
<td>ENQX23</td>
<td>602121</td>
<td>ENQX22</td>
</tr>
<tr>
<td>602123</td>
<td>ABRX1</td>
<td>602124</td>
<td>USGX03</td>
</tr>
<tr>
<td>602126</td>
<td>VACCX2</td>
<td>602127</td>
<td>CRDI18</td>
</tr>
<tr>
<td>602132</td>
<td>BOTX05</td>
<td>602133</td>
<td>CRDI20</td>
</tr>
<tr>
<td>602135</td>
<td>COMX19</td>
<td>602136</td>
<td>CRDI21</td>
</tr>
<tr>
<td>602140</td>
<td>CRDI22</td>
<td>602141</td>
<td>CRDI23</td>
</tr>
<tr>
<td>602143</td>
<td>STRX08</td>
<td>602144</td>
<td>CRDI24</td>
</tr>
<tr>
<td>602170</td>
<td>FRKHX8</td>
<td>602171</td>
<td>ARGX20</td>
</tr>
<tr>
<td>602173</td>
<td>ARGX22</td>
<td>602177</td>
<td>ARGX23</td>
</tr>
<tr>
<td>602201</td>
<td>MSTRX35</td>
<td>602202</td>
<td>DCNX13</td>
</tr>
<tr>
<td>602204</td>
<td>DCNX15</td>
<td>602205</td>
<td>GJFX50</td>
</tr>
<tr>
<td>602207</td>
<td>NODX02</td>
<td>602210</td>
<td>NDDX03</td>
</tr>
<tr>
<td>602212</td>
<td>COMX20</td>
<td>602220</td>
<td>GOKER1</td>
</tr>
<tr>
<td>602222</td>
<td>STRX09</td>
<td>602223</td>
<td>MSTRX36</td>
</tr>
<tr>
<td>602225</td>
<td>MSTRX40</td>
<td>602227</td>
<td>IOX13</td>
</tr>
<tr>
<td>602231</td>
<td>IDX15</td>
<td>602233</td>
<td>IOX17</td>
</tr>
<tr>
<td>602235</td>
<td>IDX21</td>
<td>602236</td>
<td>IOX22</td>
</tr>
<tr>
<td>602240</td>
<td>IDX24</td>
<td>602241</td>
<td>IOX25</td>
</tr>
<tr>
<td>602243</td>
<td>IDX26</td>
<td>602245</td>
<td>IOX30</td>
</tr>
<tr>
<td>602267</td>
<td>SKDX1</td>
<td>602275</td>
<td>DEVF6</td>
</tr>
<tr>
<td>602269</td>
<td>ARCFX2</td>
<td>602313</td>
<td>ARCFX3</td>
</tr>
<tr>
<td>602270</td>
<td>ARCFX5</td>
<td>602316</td>
<td>ARCFX6</td>
</tr>
<tr>
<td>602271</td>
<td>ARCFX8</td>
<td>602321</td>
<td>ARCFX9</td>
</tr>
<tr>
<td>602273</td>
<td>ARCX11</td>
<td>602324</td>
<td>ARCX12</td>
</tr>
<tr>
<td>602276</td>
<td>OPNX30</td>
<td>602327</td>
<td>OPNX31</td>
</tr>
<tr>
<td>602331</td>
<td>DELX12</td>
<td>602332</td>
<td>ARCX14</td>
</tr>
<tr>
<td>602334</td>
<td>ARCX16</td>
<td>602335</td>
<td>ARCX17</td>
</tr>
<tr>
<td>602337</td>
<td>ARCX19</td>
<td>602340</td>
<td>ARCGX26</td>
</tr>
<tr>
<td>602342</td>
<td>DIRX5</td>
<td>602343</td>
<td>IOX31</td>
</tr>
<tr>
<td>602352</td>
<td>METRX1</td>
<td>602353</td>
<td>NSPX00</td>
</tr>
<tr>
<td>602355</td>
<td>NSPX02</td>
<td>602356</td>
<td>NSPX03</td>
</tr>
<tr>
<td>602360</td>
<td>NSPX05</td>
<td>602361</td>
<td>NSPX06</td>
</tr>
<tr>
<td>602363</td>
<td>NSPX08</td>
<td>602364</td>
<td>NSPX09</td>
</tr>
<tr>
<td>602366</td>
<td>NSPX11</td>
<td>602367</td>
<td>NSPX12</td>
</tr>
<tr>
<td>602371</td>
<td>NSPX14</td>
<td>602372</td>
<td>NSPX15</td>
</tr>
<tr>
<td>602374</td>
<td>NSPX17</td>
<td>602375</td>
<td>NSPX18</td>
</tr>
<tr>
<td>602377</td>
<td>NSPX20</td>
<td>602400</td>
<td>NSPX21</td>
</tr>
<tr>
<td>602406</td>
<td>DIAG11</td>
<td>602407</td>
<td>DIAG12</td>
</tr>
<tr>
<td>602411</td>
<td>NSPX23</td>
<td>602412</td>
<td>ARGX28</td>
</tr>
<tr>
<td>602414</td>
<td>ARGX29</td>
<td>602415</td>
<td>ARGX30</td>
</tr>
<tr>
<td>602417</td>
<td>DEVF7</td>
<td>602420</td>
<td>GJFX52</td>
</tr>
<tr>
<td>602422</td>
<td>IOX32</td>
<td>602423</td>
<td>IOX33</td>
</tr>
<tr>
<td>602425</td>
<td>SIRX2</td>
<td>602426</td>
<td>RIRX1</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602431</td>
<td>SMAPX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602432</td>
<td>TTMSX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602433</td>
<td>MONX06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602434</td>
<td>BD TX06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602435</td>
<td>BD TX07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602436</td>
<td>BD TX08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602437</td>
<td>BD TX09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602440</td>
<td>BD TX10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602441</td>
<td>BD TX11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602442</td>
<td>BD TX12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602443</td>
<td>BD TX13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602444</td>
<td>BD TX14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602445</td>
<td>BD TX15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602446</td>
<td>BD TX16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602447</td>
<td>BD TX17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602450</td>
<td>BD TX18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602451</td>
<td>NTMX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602452</td>
<td>CMDX21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602453</td>
<td>DELX13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602454</td>
<td>ANTX01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602455</td>
<td>TTYX02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602456</td>
<td>NSPX24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602457</td>
<td>NSPX25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602460</td>
<td>NSPX26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602461</td>
<td>GJFX53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602462</td>
<td>IDX34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602463</td>
<td>IDX35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602464</td>
<td>PMAPX8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602465</td>
<td>SMAPX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602466</td>
<td>BD TX19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602470</td>
<td>BD TX20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602471</td>
<td>ILLX05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602472</td>
<td>XSEVX1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602473</td>
<td>XSEVX2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602474</td>
<td>XSEVX3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>602475</td>
<td>ABRKX2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 [ABERK]</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>ALCX1:</td>
<td>601137</td>
<td>Invalid function [ALLOC]</td>
</tr>
<tr>
<td>ALCX2:</td>
<td>601140</td>
<td>WHEEL or OPERATOR capability required [ALLOC]</td>
</tr>
<tr>
<td>ALCX3:</td>
<td>601141</td>
<td>Device is not assignable [ALLOC]</td>
</tr>
<tr>
<td>ALCX4:</td>
<td>601142</td>
<td>Invalid job number [ALLOC]</td>
</tr>
<tr>
<td>ALCX5:</td>
<td>601143</td>
<td>Device already assigned to another job [ALLOC]</td>
</tr>
<tr>
<td>ALCX6:</td>
<td>601153</td>
<td>Device assigned to user job, but will be given to allocator when released [ALLOC]</td>
</tr>
<tr>
<td>ANTXO1:</td>
<td>602454</td>
<td>No more network terminals available [MTOPR]</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 .ARSS [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
[ARCF]

ARCX13: 602325 Archive system request not completed
[ARCF]

ARCX14: 602332 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 [ABBRK; BOOD; DSKAS;
GTHST%; 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 [CRLNM; 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 LUUO 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 [DELD; 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]

181
ATACX2: 600321 Job already attached [ATCH]
ATACX3: 600322 Incorrect user number [ATCH]
ATACX4: 600323 Invalid password [ATCH]
ATACX5: 600324 This job has no controlling terminal [ATCH]
ATACX6: 601502 Terminal is already attached to a job [ATCH]
ATACX7: 601503 Illegal terminal number [ATCH]
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]
BKJFX1: 600454 Illegal to back up terminal pointer twice [BKJFN]
BOTX01: 602016 Invalid DTE-20 number [BOOT]
BOTX02: 602017 Invalid byte size [BOOT]
BOTX03: 602031 Invalid protocol version number [BOOT]
BOTX04: 602114 Byte count is not positive [BOOT]
BOTX05: 602132 Protocol initialization failed [BOOT]
BOTX06: 602434 GTJFN failed for dump file [BOOT]
BOTX07: 602435 OPENF failed for dump file [BOOT]
BOTX08: 602436 Dump failed [BOOT]
BOTX09: 602437 To -10 error on dump [BOOT]
BOTX10: 602440 To -11 error on dump [BOOT]
BOTX11: 602441 Failed to assign page on dump [BOOT]
BOTX12: 602442 Reload failed [BOOT]
BOTX13: 602443 -11 didn’t power down [BOOT]
BOTX14: 602444 -11 didn’t power up [BOOT]
BOTX15: 602445 ROM did not ACK the -10 [BOOT]
BOTX16: 602446 -11 boot program did not make it to -11 [BOOT]
BOTX17: 602447 -11 took more than 1 minute to reload; will cause retry [BOG]
BOTX18: 602450 Unknown BOOT error [BOOT]
BOTX19: 602467 Overdue TO-11 transfer aborted [BOOT]
BOTX20: 602470 Overdue TO-10 transfer aborted [BOOT]
CACTX1: 600045 Invalid account identifier [CACTE]
CACTX2: 600046 Job is not logged in [CACTE; MSTR]
CAPX1: 600615 WHEEL or OPERATOR capability required [ARCF; ACCES: BOOT; GIVOK%; MSFRK; MTU%];
NTMAN%; PEEK; RCVDK%; 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]

CFRKX3: 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]

CLSX1: 600160 File is not open [CLOSF]

CLSX2: 600161 File cannot be closed by this process [CLOSF]

CLSX3: 601151 File still mapped [CLOSF]

CLSX4: 601217 Device still active [CLOSF]

CNDX1: 600200 Invalid password [ACCES]

CNDX5: 600204 Job is not logged in [ACCES]

CNDX7: 602004 The CNDIR JSYS has been replaced by ACCES

COMNX1: 601257 Invalid COMND 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: 602035 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]

CRDI10: 601170 Maximum directory number exceeded; index table needs expanding [CRDIR]

CRDI11: 601427 Invalid terminating bracket on directory [CRDIR]

CRDI12: 601451 Structure is not mounted [CRDIR]

CRDI13: 602101 Request exceeds superior directory working quota [CRDIR]

CRDI14: 602102 Request exceeds superior directory permanent quota [CRDIR]

CRDI15: 602103 Request exceeds superior directory subdirectory quota [CRDIR]

183
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]
CRJBX1:  600020  Invalid parameter or function bit combination [CRJOB]
CRJBX2:  600021  Illegal for created job to enter MINI-EXEC [CRJOB]
CRJBX4:  600023  Terminal is not available [CRJOB]
CRJBX5:  600024  Unknown name for LGGIN [CRJOB]
CRJBX6:  600025  Insufficient system resources [CRJOB]
CRLNX1:  601000  Logical name is not defined [CRLNM]
CRLNX2:  601136  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 [ODCNV; ODTIM; SFTAD]
DATEX7:  602310  Julian day is out of range [IDCNV]
DBRXX1:  601275  No interrupts in progress [DEBRK]
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
DCNX13: 602202 Node not accessible
DCNX14: 602203 Previous interrupt message outstanding
DCNX15: 602204 No interrupt message available
DECRSV: 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]
DELFX1: 600170 Delete access required [DELF; DELNF]
DELFX2: 601303 File cannot be expunged because it is currently open [DELF; DELF]
DELFX3: 601304 System scratch area depleted; file not deleted [DELF]
DELFX4: 601305 Directory symbol table could not be rebuilt [DELF; DELF]
DELFX5: 601306 Directory symbol table needs rebuilding [DELF; DELF]
DELFX6: 601307 Internal format of directory is incorrect [DELF; DELF; DIRST; GFUST; GTDAL; PPNST; VACCT]
DELFX7: 601310 FDB formatted incorrectly; file not deleted [DELF; DELF]
DELFX8: 601311 FDB not found; file not deleted [DELF; DELF]
DELFX9: 601411 File is not a directory file [DELF]
DELX11: 602330 File has archive status, delete is not permitted
DELX12: 602331 File has no pointer to offline storage [DELF]
DELX13: 602453 File is marked "Never Delete" [DELF; DELNF]
DE SX1: 600150 Invalid source/destination designator [BIN; BKJFN; BOUT; CFIBF; CFOBF; CHFDB; CLOSF; DELF; DELNF; DEVST; DFIN; DFOUT; DIBE; DIRST; DOBE; DUMPO; DVCHR; ERSTR; FFFFMP; FFUPF; FLIN; FLOUT; GACTF; GDSTS; GFUST; GNJFN; GTFDB; GTTYP; JFNS; MTOPR; MTU%; NIN; NOUT; OPENF; PBIN; PBOUT; PMAP; PPNST; PSOUT; RCDIR; RFBSZ; RFPOS; RFPTKR; RFSTAD; RIN; RLJFN; RNMF; ROUT; RPACS; SACTF; SCTTY; SDSTS; SFBSZ; SFCOC; SFMOD; SFPTKR; SFTAD; SFUST; SIBE; SIN; SINR; SIZEF; SOBE; SOUT; SOUTR; SPACS; STPAR; STPPN; STSTS; STTYP; SWJFN; TLINK]
DE SX2: 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; DUMPI; DUMPO; FLIN;
FLOUT; GACTF; GDSTS; GFUST; GNJFN; JFNS;
MTOPR; NIN; NOUT; PBIN; PBOUT; PPNST;
PSOUT; RFBSZ; RFPTR; RIN; ROUT; SDSTS;
SFBSZ; SFCOC; SFPTR; SFUST; SIN; SIZEF;
SOUT; SPACS; STI; STO; STPPN; STSTS;
STTP; SWJFN]

DEsx3:  600152  JFN is not assigned [BIN; BKJFN; BOUT;
CFIBF; CFOBF; CHFDB; CLOSF; DELF; DELNF;
DEVST; DFIN; Dfout; DIBE; DIRST; DOBE;
DUMPI; DUMPO; DVCHR; FFFF; FFUF; FLIN;
FLOUT; GACTF; GDSTS; GFUST; GNJFN; GTFDB;
JFNS; MTOPR; NIN; NOUT; OPENF; PBOUT;
PMAP; PPNST; PSOUT; RCDIR; RFBSZ; RFPOS;
RFPTR; RFTAD; RIN; RLJFN; RNAMF; ROUT;
RPACS; SACTF; SDSTS; SFBSZ; SFCOC; SFMOD;
SFPTR; SFUST; SIBE; SIN; SINR;
SIZEF; SOBE; SOUT; SOUTR; SPACS; SPJFN;
STPAR; STPPN; STSTS; SWJFN; UFPGS; WILD%]

DEsx4:  600153  Invalid use of terminal designator or
string pointer [CHFDB; CLOSF; DELF; DELNF;
DUMPI; DUMPO; DVCHR; FFFF; FFUF; GACTF;
GDSTS; GFUST; GNJFN; GTFDB; JFNS; MTOPR;
OPENF; RCDIR; RFBSZ; RFPTR; RIN; RLJFN;
RNAMF; ROUT; RPACS; SACTF; SDSTS; SFBSZ;
SF PTR; SFUST; SIZEF; SPACS; STPPN; STSTS;
SWJFN; UFPGS]

DEsx5:  600154  File is not open [BIN; BKJFN; BOUT; CFIBF;
CFOB; DEQ; DFIN; Dfout; DIBE; DIRST;
DOBE; DUMPI; DUMPO; ENQ; FFFF; FLIN;
FLOUT; GDSTS; GFUST; MTOPR; NIN; NOUT;
PBIN; PBOUT; PMAP; PPNST; PSOUT; RFBSZ;
RFPOS; RFPTR; RIN; ROUT; RPACS; SDSTS;
SFBSZ; SFCOC; SFMOD; SIBE; SIN; SINR;
SOBE; SOUT; SOUTR; 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;
FFUF; GACTF; GFUST; GTFDB; OPENF; PMAP;
RCDIR; RFTAD; RNAMF; SFUST; STPPN;
UFPGS]

DEsx8:  600157  File is not on disk [DEQ; ENQ; ENQC;
GFUST; RCDIR; RPACS; SFBSZ; SFPTR; SFUST;
SPACS; STPPN; UFPGS]

DEsx9:  601340  Invalid operation for this device [DELF;
GTJFN; MTOPR; MTU%; SDSTS]

DEsx10: 601417 Structure is dismounted [GFUST; RCDIR;
SFUST; STPPN]

DEsx11: 602410 Invalid operation for this label type
[MTOPR]

DEVX1:  600335  Invalid device designator [ALLOC; ASND;
DEVST; DVCHR; GDSKC; RELD]

DEVX2:  600336  Device already assigned to another job
[ASND; CFIBF; CFOBF; DIBE; DOBE; MTOPR;
RELD; RFPOS; SCTTY; SFCOC; SFMOD; SIBE;
SOBE; STI; STO; STPAR]
TOPS-20 Monitor Calls Quick Reference Guide
TOPS-20 JSYS Error Mnemonics

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]
DIAGX1: 601174  Invalid function [DIAG]
DIAGX2: 601175  Device is not assigned [DIAG]
DIAGX3: 601176  Argument block too small [DIAG]
DIAGX4: 601177  Invalid device type [DIAG]
DIAGX5: 601200  WHEEL, OPERATOR, or MAINTENANCE capability required [DIAG]
DIAGX6: 601201  Invalid channel command list [DIAG]
DIAGX7: 601202  Illegal to do I/O across page boundary [DIAG]
DIAGX8: 601203  No such device [DIAG]
DIAGX9: 601204  Unit does not exist [DIAG]
DILFX1: 600464  Invalid date format [IDTIM; IDTNC]
DIRX1:  601313  Invalid directory number [DIRST; GTDAL, PPNST; VACCT]
DIRX2:  601314  Insufficient system resources [DIRST; GFUST; PPNST]
DIRX3:  601315  Internal format of directory is incorrect [DIRST; GFUST; PPNST; VACCT]
DIRX5:  602342  Directory too large
DLFX10: 602010  Cannot delete directory; file still mapped [DLFL]
DLFX11: 602011  Cannot delete directory file in this manner [DLFL]
DSK0X1: 601343  Channel number too large [DSKOP]
DSK0X2: 601344  Unit number too large [DSKOP]
DSK0X3: 601416  Invalid structure number [DSKOP]
DSK0X4: 601420  Invalid address type specified [DSKOP]
DSK0X5: 601533  Invalid word count
DSK0X6: 601534  Invalid buffer address
DSKKX0: 601365  Invalid structure number [DSKAS]
DSKKX03: 601367  Bit table has not been initialized [DSKAS]
DSKKX05: 601763  Disk assignments and deassignments are currently prohibited [DSKAS]
DSKKX06: 601764  Invalid disk address [DSKAS]
DSKKX07: 601765  Address cannot be deassigned because it has not been assigned [DSKAS]
DSKKX08: 601766  Address cannot be assigned because it is already assigned [DSKAS]
DSMX1:  600555  File(s) not closed [ASND]
DUMPX1: 600440  Command list error [DUMP1; DUMPO]
DUMPX2: 600441  JFN is not open in dump mode [DUMP1; DUMPO]
DUMPX3: 600442  Address error (too big or crosses end of memory) [DUMP1; DUMPO]
DUMPX4: 600443  Access error (cannot read or write data in memory) [DUMP1; DUMPO]
DUMPX5: 601214  No-wait dump mode not supported for this device [DUMP1; DUMPO]
DUMPX6: 601215  Dump mode not supported for this device
[DUMP; DUMPO]

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 DFN for
<SYSTEM>ACCOUNTS-TABLE.BIN

ENQX10: 601066 Invalid argument block length [DEQ; ENQ; ENQC]

ENQX11: 601067 Invalid software interrupt channel number
[DEQ; ENQ; ENQC]

ENQX12: 601070 Invalid number of resources requested
[ENQ; ENQC]

ENQX13: 601071 Indirect or indexed byte pointer not allowed [DEQ; ENQ; ENQC]

ENQX14: 601072 Invalid byte size [DEQ; ENQ; ENQC]

ENQX15: 601073 ENQ/DEQ capability required [DEQ; ENQ; ENQC]

ENQX16: 601074 WHEEL or OPERATOR capability required
[DEQ; ENQ; ENQC]

ENQX17: 601075 Invalid JFN [DEQ; ENQ; ENQC]

ENQX18: 601076 Quota exceeded [DEQ; ENQ; ENQC]

ENQX19: 601077 String too long [DEQ; ENQ; ENQC]

ENQX1: 601055 Invalid function [DEQ; ENQ; ENQC]

ENQX20: 601100 Locked JFN cannot be closed [CLDSF; DEQ;
ENQ; ENQC]

ENQX21: 601101 Job is not logged in [DEQ; ENQ; ENQC]

ENQX22: 602121 Invalid mask block length [ENQ]

ENQX23: 602120 Mismatched mask block lengths [ENQ]

ENQX24: 601056 Level number too small [DEQ; ENQ; ENQC]

ENQX3: 601057 Request and lock level numbers do not match [DEQ; ENQ; ENQC]

ENQX4: 601060 Number of pool and lock resources do not match [DEQ; ENQ; ENQC]

ENQX5: 601061 Lock already requested [ENQ; ENQC]

ENQX6: 601062 Requested locks are not all locked [DEQ;
ENQ; ENQC]

ENQX7: 601063 No ENQ on this lock [DEQ; ENQC]

ENQX8: 601064 Invalid access change requested [ENQ; ENQC]

ENQX9: 601065 Invalid number of blocks specified [DEQ;
ENQ; ENQC]

FFUFX1: 600544 File is not open [FFUFP]

FFUFX2: 600545 File is not on multiple-directory device
[FFUFP]

FFUFX3: 600546 No used page found [FFUFP]

FLINX1: 600650 First character is not blank or numeric
[DFIN; FLIN]

FLINX2: 600651 Number too small [DFIN; FLIN]

FLINX3: 600652 Number too large [DFIN; FLIN]

FLINX4: 600653 Invalid format [DFIN; FLIN]

FLOTX1: 600660 Column overflow in field 1 or 2 [DFOUT;
FLOUT]

FLOTX2: 600661 Column overflow in field 3 [DFOUT; FLOUT]

FLOTX3: 600662 Invalid format specified [DFOUT; FLOUT]
FRKHX1:  600250  Invalid process handle [ADBRK; AIC; CLZFF; DIC; DIR; EIR; EPCAP; ERSTR; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; GTRPW; HFORK; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRCM; RMAP; RPACS; RPCAP; RTIW; RUNTM; RWI; RWM; SAVE; SCTTY; SCVEC; SETER; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SKPIR; SPACS; SPJFN; SPLFK; SSAVE; STIW; UTFRK; WFORK; XGVEC%; XMRAP%; XSRFK%; XSIR%; XSVEC%]

FRKHX2:  600251  Illegal to manipulate a superior process [ADBRK; AIC; CLZFF; DIC; DIR; EIR; EPCAP; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; GTRPW; HFORK; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRCM; RPACS; RTIW; RWM; SAVE; SCTTY; SCVEC; SETER; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SKPIR; SPACS; SPJFN; SSAVE; STIW; UTFRK; WFORK; XGVEC%; XSRFK%; XSIR%; XSVEC%]

FRKHX3:  600252  Invalid use of multiple process handle [ADBRK; AIC; CLZFF; DIC; DIR; EIR; FFORK; GCVEC; GET; GEVEC; GFRKH; GFRKS; GPJFN; GTRPI; GTRPW; GTRPW; HFORK; IIC; KFORK; MSFRK; PMAP; RCM; RFACS; RFORK; RFRKH; RFSTS; RIR; RIRCM; RPACS; RTIW; RWM; SAVE; SCVEC; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SKPIR; SPACS; SPJFN; SSAVE; STIW; UTFRK; XGVEC%; XSRFK%; XSIR%; XSVEC%]

FRKHX4:  600253  Process is running [RFACS; SCVEC; SETER; SFACS; SFORK]

FRKHX5:  600254  Process has not been started [SFORK; XSRFK%]

FRKHX6:  600255  All relative process handles in use [CFORK; GFRKH; GFRKS; RFRKH]

FRKHX7:  601312  Process page cannot exceed 777 [PMAP]

FRKHX8:  602170  Illegal to manipulate an execute-only process [ADBRK; AIC; CFORK; DIC; DIR; EIR; GET; IIC; PMAP; RFACS; SAVE; SCVEC; SDVEC; SETER; SEVEC; SFACS; SFORK; SFRKV; SIR; SIRCM; SPACS; STIW; TFORK; UTFRK; XSRFK%; XSIR%; XSVEC%]

GACCCX1:  601272  Invalid job number [GACCT]

GACCCX2:  601273  No such job [GACCT]

GACCCX3:  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:  600373  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 multisection
GETX5: 601560 EXE file directory entry specifies a
section-crossing [GET]
GFDBX1: 600424 Invalid displacement [GTFDB]
GFDBX2: 600425 Invalid number of words [GTFDB]
GFDBX3: 600426 List access required [GTFDB]
GFKX1: 601011 Area too small to hold process structure
[GFRKS]
GFRXX1: 600371 Invalid process handle [GFRKH]
GFUX1: 601371 Invalid function [GFUST]
GFUX2: 601372 Insufficient system resources [GFUST]
GFUX3: 601452 File expunged [GFUST]
GFUX4: 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; PPNST]
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 on-line [GTJFN]
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GJFX31</td>
<td>600113 Invalid wildcard designator [CRLNM; GTJFN]</td>
</tr>
<tr>
<td>GJFX32</td>
<td>600114 No files match this specification [GTJFN]</td>
</tr>
<tr>
<td>GJFX33</td>
<td>600115 Filename was not specified [GTJFN]</td>
</tr>
<tr>
<td>GJFX34</td>
<td>600116 Invalid character &quot;?&quot; in file specification [GTJFN]</td>
</tr>
<tr>
<td>GJFX35</td>
<td>600117 Directory access privileges required [GTJFN]</td>
</tr>
<tr>
<td>GJFX36</td>
<td>600760 Internal format of directory is incorrect [GTJFN]</td>
</tr>
<tr>
<td>GJFX37</td>
<td>601133 Input deleted [GTJFN]</td>
</tr>
<tr>
<td>GJFX38</td>
<td>601164 File not found because output-only device was specified [GTJFN]</td>
</tr>
<tr>
<td>GJFX39</td>
<td>601165 Logical name loop detected [GTJFN]</td>
</tr>
<tr>
<td>GJFX40</td>
<td>601225 Undefined attribute in file specification [GTJFN]</td>
</tr>
<tr>
<td>GJFX41</td>
<td>601277 File name must not exceed 6 characters [GTJFN]</td>
</tr>
<tr>
<td>GJFX42</td>
<td>601300 File type must not exceed 3 characters [GTJFN]</td>
</tr>
<tr>
<td>GJFX43</td>
<td>601754 More than one ;T specification is not allowed [GTJFN]</td>
</tr>
<tr>
<td>GJFX44</td>
<td>602012 Account string does not match [GTJFN]</td>
</tr>
<tr>
<td>GJFX45</td>
<td>602060 Illegal to request multiple specifications for the same attribute [GTJFN]</td>
</tr>
<tr>
<td>GJFX46</td>
<td>602061 Attribute value is required [GTJFN]</td>
</tr>
<tr>
<td>GJFX47</td>
<td>602062 Attribute does not take a value [GTJFN]</td>
</tr>
<tr>
<td>GJFX48</td>
<td>602064 GTJFN input buffer is empty [GTJFN]</td>
</tr>
<tr>
<td>GJFX49</td>
<td>602065 Invalid attribute for this device [GTJFN]</td>
</tr>
<tr>
<td>GJFX50</td>
<td>602205 Invalid argument for attribute [GTJFN]</td>
</tr>
<tr>
<td>GJFX51</td>
<td>602211 Byte count too small [GTJFN]</td>
</tr>
<tr>
<td>GJFX52</td>
<td>602420 End of tape encountered while searching for file</td>
</tr>
<tr>
<td>GJFX53</td>
<td>602461 Tape label filename specification exceeds 17 characters [GTJFN]</td>
</tr>
<tr>
<td>GNJFX1</td>
<td>601054 No more files in this specification [GNJFN]</td>
</tr>
<tr>
<td>GOKER1</td>
<td>602220 Illegal function [GETOK%]</td>
</tr>
<tr>
<td>GOKER2</td>
<td>602221 Request denied by Access Control Facility [GETOK%]</td>
</tr>
<tr>
<td>GOKER3</td>
<td>602421 JSYS not executed within ACU fork [GIVOK%; RCVOK%]</td>
</tr>
<tr>
<td>GTABX1</td>
<td>600267 Invalid table number [GETAB]</td>
</tr>
<tr>
<td>GTABX2</td>
<td>600270 Invalid table index [GETAB]</td>
</tr>
<tr>
<td>GTABX3</td>
<td>600271 GETAB privileges required [GETAB]</td>
</tr>
<tr>
<td>GTDIX1</td>
<td>600640 WHEEL or OPERATOR capability required [GTDIR; TTMSG]</td>
</tr>
<tr>
<td>GTDIX2</td>
<td>600641 Invalid directory number [GTDIR]</td>
</tr>
<tr>
<td>GTHSX1</td>
<td>600704 Unknown host number [GTHST%]</td>
</tr>
<tr>
<td>GTHSX2</td>
<td>600705 No number for that host name [GTHST%]</td>
</tr>
<tr>
<td>GTHSX3</td>
<td>600707 No string for that host number [GTHST%]</td>
</tr>
<tr>
<td>GTJIX1</td>
<td>601013 Invalid index [GETJI; GTHST%; GTNCP%]</td>
</tr>
<tr>
<td>GTJIX2</td>
<td>601014 Invalid terminal line number [GETJI]</td>
</tr>
<tr>
<td>GTJIX3</td>
<td>601015 Invalid job number [GETJI]</td>
</tr>
<tr>
<td>GTJIX4</td>
<td>601254 No such job [GETJI]</td>
</tr>
<tr>
<td>GTNCX1</td>
<td>600746 Invalid network JFN [GTNCP%]</td>
</tr>
<tr>
<td>GTNCX2</td>
<td>600747 Invalid or inactive NVT [GTNCP%]</td>
</tr>
</tbody>
</table>

191
HFRKX1: 600370 Illegal to halt self with HFKRX [HFRKRX]
HPTX1: 600670 Undefined clock number [HPTIM]
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 operator code
ILINS2: 600771 Undefined JSYS [ASNSQ; ATNVN; CVHST;
CVKST; FLHST; GTHST; GTNCP; RCVNM; RELSQ; SINDM]
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; SIRN; DUMP]
IOX2: 600216 File is not open for writing [BOUT; PBOUT;
PSOUT; SOUSTR; 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; SIRN; MTOR]
IOX5: 600221 Device or data error [BIN; BOUT; DUMP;
DUMP; MTOPR; PBIN; PBOUT; PSOUT; RIN;
ROUT; SIN; SIRN; SOUT; SOUSTR]
IOX6: 600222 Illegal to write beyond absolute end of
file [PBOUT; PSOUT; ROUT; SOUT; SOUSTR;
BOUT]
IOX7: 601211 Insufficient system resources (Job Storage
Block full) [SIN; SIRN; SOUT; SOUSTR]
IOX8: 601212 Monitor internal error [MTUR%; SIN; SIRN;
SOUT; SOUSTR]
IOX9: 601216 Function legal for sequential write only
[SOUTR]
IOX10: 601240 Record is longer than user requested
[SINR]
IOX11: 601440 Quota exceeded [BOUT; CLOSF; CLZFF; DEVST;
DFOUT; DIRST; DUMP; ERSTR; FOUT; GTJFN;
JFNS; NOUT; PBOUT; PMAP; PPNST; PSOUT;
RDTTY; ROUT; SAVE; SOUT; SOUSTR; SSAVE;
TEXT; 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]
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]
IDX26: 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; CLOSF; CLZFF; DEVST; DFOUT; DIRST; DUMPO; ERSTR; FLOUT; GTJFN; JFNS; NOUT; PBOUT; PMAP; PPNST; PSOUT; RDTTY; ROUT; SAVE; SOUT; SOUR; SSAVE; TEXTI; UFPGS]
IDX35: 602463 Unable to allocate disk - structure damaged [BOUT; CLOSF; CLZFF; DEVST; DFOUT; DIRST; DUMPO; ERSTR; FLOUT; GTJFN; JFNS; NOUT; PBOUT; PMAP; PPNST; PSOUT; RDTTY; ROUT; SAVE; SOUT; SOUR; SSAVE; TEXTI; UFPGS]
IPCF10: 601027 WHEEL capability required [MUTIL]
IPCF11: 601030 WHEEL or IPCF capability required [MRECV; MSEND; MUTIL]
IPCF12: 601031 No free PID's available [MSEND; MUTIL]
IPCF13: 601032 PID quota exceeded [MSEND; MUTIL]
IPCF14: 601033 No PID's available to this job [MRECV; MSEND; MUTIL]
IPCF15: 601034 No PID's available to this process [MRECV; MSEND; MUTIL]
IPCF16: 601035 Receive and message data modes do not match [MRECV; MSEND]
IPCF17: 601036 Argument block too small [MUTIL]
IPCF18: 601037 Invalid MUTIL SYS function [MUTIL]
IPCF19: 601040 No PID for [SYSTEM]INFO [MSEND; 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 [MRECV; MSEND; MUTIL]
IPCF25: 601046 PID does not belong to this job [MRECV; MSEND; MUTIL]
IPCF26: 601047 PID does not belong to this process [MRECV; MSEND; MUTIL]
IPCF27: 601050 PID is not defined [MRECV; MSEND; MUTIL]
IPCF28: 601051 PID not accessible by this process [MRECV; MSEND; MUTIL]
IPCF29: 601052 PID already being used by another process [MRECV; MSEND; MUTIL]
IPCF30: 601053 Job is not logged in [MUTIL]
IPCF31: 601102 Invalid page number [MRECV; MSEND]
IPCF32: 601103 Page is not private [MRECV; MSEND; MUTIL]
IPCF33: 601130 Invalid index into system PID table [MUTIL]
IPCF34: 601320 Cannot receive into an existing page [MRECVR]
IPCF35: 602125 Invalid IPCF quota [MUTIL]
IPCFX1: 601016 Length of packet descriptor block cannot be less than 4 [MRECVR; MSEND]
IPCFX2: 601017 No message for this PID [MRECVR; MUTIL]
IPCFX3: 601020 Data too long for user's buffer [MRECVR; MUTIL]
IPCFX4: 601021 Receiver's PID invalid [MRECVR; MSEND; MUTIL]
IPCFX5: 601022 Receiver's PID disabled [MRECVR; 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]
KDUX01: 602206 KMC not running [BOOT]
KFRX01: 600365 Illegal to kill top level process [KFORK]
KFRX02: 600366 Illegal to kill self [KFORK]
LCBDDB: 601475 Bad byte pointer passed to LCS
LCNDND: 601477 LCS No such node
LGXX1: 600010 Invalid account identifier [LOGIN]
LGXX2: 600011 Directory is "files-only" and cannot be logged into [ACCES; LOGIN]
LGXX3: 600012 Internal format of directory is incorrect [LOGIN]
LGXX4: 600013 Invalid password [LOGIN]
LGXX5: 600014 Job is already logged in [LOGIN]
LGXX6: 601337 No more job slots available for logging in [LOGIN]
LNST01: 601317 Page table does not exist and file not open for write [PMAP; UFPGS]
LNST02: 601002 No such logical name [LNMST]
LNST02: 601136 Invalid function [LNMST]
LOCKX1: 601771 Illegal to lock other than a private page [PLOCK]
LOCKX2: 601772 Requested page unavailable [PLOCK]
LOUT01: 600035 Illegal to specify job number when logging out own job [LGOUT]
LOUT02: 600036 Invalid job number [LGOUT; MSTR]
LOUT03: 601227 WHEEL or OPERATOR capability required [LGOUT]
LOUT04: 601230 LOG capability required [LGOUT]
LOUT05: 601753 Illegal to log out job O [LGOUT]
LPWX1: 601333 Invalid unit number [LPNI]
LPWX2: 601334 WHEEL or OPERATOR capability required [LPNI]
LPWX3: 601335 Illegal to load RAM or VFU while device is OPEN [LPNI]
LSTR01: 601405 Process has not encountered any errors [GETER]
LTLBLX: 602347 Too many user labels
METR01: 602352 METER% not implemented for this processor [METER%]
MONXO1: 601727 Insufficient system resources [GETOK%; MSTR]
MONXO2: 601730 Insufficient system resources (JSB full) [PVDOH%; VACCT]
MONXO3: 601731 Monitor internal error [CRDIR]
MONXO4: 601732 Insufficient system resources (swapping space full)
MONXO5: 602032 Insufficient system resources (no resident free space) [GETOK%; MSTR]
MONXO6: 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]
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSTX35:</td>
<td>602201 Too many units in structure [MSTR]</td>
</tr>
<tr>
<td>MSTX36:</td>
<td>602223 Illegal while JFNs assigned [MSTR]</td>
</tr>
<tr>
<td>MSTX37:</td>
<td>602224 Illegal while connected to structure [MSTR]</td>
</tr>
<tr>
<td>MSTX40:</td>
<td>602225 Invalid PSI channel number given [MSTR]</td>
</tr>
<tr>
<td>MSTX41:</td>
<td>601461 Channel does not exist [MSTR]</td>
</tr>
<tr>
<td>MSTX42:</td>
<td>601462 Controller does not exist [MSTR]</td>
</tr>
<tr>
<td>MTOX10:</td>
<td>601323 VFU or RAM file cannot be OPNEned [MTOPR]</td>
</tr>
<tr>
<td>MTOX11:</td>
<td>601324 Data too large for buffers [MTOPR]</td>
</tr>
<tr>
<td>MTOX12:</td>
<td>601325 Input error or not all data read [MTOPR]</td>
</tr>
<tr>
<td>MTOX13:</td>
<td>601326 Argument block too small [MTOPR]</td>
</tr>
<tr>
<td>MTOX14:</td>
<td>601327 Invalid software interrupt channel number [MTOPR]</td>
</tr>
<tr>
<td>MTOX15:</td>
<td>601331 Device does not have Direct Access [programmable] VFU [MTOPR]</td>
</tr>
<tr>
<td>MTOX16:</td>
<td>601332 VFU or Translation RAM file must be on disk [MTOPR]</td>
</tr>
<tr>
<td>MTOX17:</td>
<td>601336 Device is not on line [MTOPR]</td>
</tr>
<tr>
<td>MTOX18:</td>
<td>601407 Invalid software interrupt channel number [MTOPR]</td>
</tr>
<tr>
<td>MTOX19:</td>
<td>601755 Invalid terminal line width [MTOPR]</td>
</tr>
<tr>
<td>MTOX20:</td>
<td>601210 Invalid function [MTOPR]</td>
</tr>
<tr>
<td>MTOX21:</td>
<td>601756 Invalid terminal line length [MTOPR]</td>
</tr>
<tr>
<td>MTOX22:</td>
<td>601220 Record size was not set before I/O was done [MTOPR]</td>
</tr>
<tr>
<td>MTOX3:</td>
<td>601221 Function not legal in dump mode [MTOPR]</td>
</tr>
<tr>
<td>MTOX4:</td>
<td>601222 Invalid record size [MTOPR]</td>
</tr>
<tr>
<td>MTOX5:</td>
<td>601213 Invalid hardware data mode for magtape [MTOPR]</td>
</tr>
<tr>
<td>MTOX6:</td>
<td>601223 Invalid magtape density [MTOPR]</td>
</tr>
<tr>
<td>MTOX7:</td>
<td>601226 WHEEL or OPERATOR capability required [MTOPR]</td>
</tr>
<tr>
<td>MTOX8:</td>
<td>601274 Argument block too long [MTOPR]</td>
</tr>
<tr>
<td>MTOX9:</td>
<td>601322 Output still pending [MTOPR]</td>
</tr>
<tr>
<td>NODX02:</td>
<td>602207 Line not turned off [NODE]</td>
</tr>
<tr>
<td>NODX03:</td>
<td>602210 Another line already looped [NODE]</td>
</tr>
<tr>
<td>NOUTX1:</td>
<td>600407 Radix is not in range 2 to 36 [NOUT]</td>
</tr>
<tr>
<td>NOUTX2:</td>
<td>600410 Column overflow [NOUT]</td>
</tr>
<tr>
<td>NPX2C2L:</td>
<td>602413 Two colons required on node name [COMND]</td>
</tr>
<tr>
<td>NPXAMB:</td>
<td>602044 Ambiguous [COMND]</td>
</tr>
<tr>
<td>N PXCA:</td>
<td>602057 Comma not given [COMND]</td>
</tr>
<tr>
<td>NPXICN:</td>
<td>602052 Invalid character in number [COMND]</td>
</tr>
<tr>
<td>NPXIDT:</td>
<td>602053 Invalid device terminator [COMND]</td>
</tr>
<tr>
<td>NPXINW:</td>
<td>602050 Invalid guide word [COMND]</td>
</tr>
<tr>
<td>NPXNC:</td>
<td>602051 Not confirmed [COMND]</td>
</tr>
<tr>
<td>NPXNMD:</td>
<td>602056 Does not match directory or user name [COMND]</td>
</tr>
<tr>
<td>NPXNMT:</td>
<td>602055 Does not match token [COMND]</td>
</tr>
<tr>
<td>NPXNOM:</td>
<td>602046 Does not match switch or keyword [COMND]</td>
</tr>
<tr>
<td>NPXNQS:</td>
<td>602054 Not a quoted string - quote missing at beginning or end [COMND]</td>
</tr>
<tr>
<td>NPXNSW:</td>
<td>602045 Not a switch - does not begin with slash [COMND]</td>
</tr>
<tr>
<td>NPXNUL:</td>
<td>602047 Null switch or keyword given [COMND]</td>
</tr>
<tr>
<td>NSPXOO:</td>
<td>602353 Connection not accepted</td>
</tr>
<tr>
<td>NSPXO1:</td>
<td>602354 Resource allocation failure</td>
</tr>
<tr>
<td>NSPX02: 602355</td>
<td>Destination node does not exist</td>
</tr>
<tr>
<td>NSPX03: 602356</td>
<td>Node shutting down</td>
</tr>
<tr>
<td>NSPX04: 602357</td>
<td>Destination process does not exist</td>
</tr>
<tr>
<td>NSPX05: 602360</td>
<td>Invalid process name</td>
</tr>
<tr>
<td>NSPX06: 602361</td>
<td>Destination process queue overflow</td>
</tr>
<tr>
<td>NSPX07: 602362</td>
<td>Unspecified error</td>
</tr>
<tr>
<td>NSPX08: 602363</td>
<td>Connection aborted by third party</td>
</tr>
<tr>
<td>NSPX09: 602364</td>
<td>Link aborted by process</td>
</tr>
<tr>
<td>NSPX10: 602365</td>
<td>NSP Failure - Flow control violation</td>
</tr>
<tr>
<td>NSPX11: 602366</td>
<td>Too many connections to node</td>
</tr>
<tr>
<td>NSPX12: 602367</td>
<td>Too many connections to destination process</td>
</tr>
<tr>
<td>NSPX13: 602370</td>
<td>Access denied due to unacceptable user name or password</td>
</tr>
<tr>
<td>NSPX14: 602371</td>
<td>NSP failure - invalid SERVICES field</td>
</tr>
<tr>
<td>NSPX15: 602372</td>
<td>Invalid account</td>
</tr>
<tr>
<td>NSPX16: 602373</td>
<td>NSP failure - invalid SEGSIZ field</td>
</tr>
<tr>
<td>NSPX17: 602374</td>
<td>Process aborted, timed out, or cancelled request</td>
</tr>
<tr>
<td>NSPX18: 602375</td>
<td>No path to destination node</td>
</tr>
<tr>
<td>NSPX19: 602376</td>
<td>NSP failure - flow control failure</td>
</tr>
<tr>
<td>NSPX20: 602377</td>
<td>NSP failure - invalid DSTADDR</td>
</tr>
<tr>
<td>NSPX21: 602400</td>
<td>Disconnect confirmation</td>
</tr>
<tr>
<td>NSPX22: 602401</td>
<td>NSP failure - image data field too long</td>
</tr>
<tr>
<td>NSPX23: 602411</td>
<td>Invalid NSP reason code</td>
</tr>
<tr>
<td>NSPX24: 602456</td>
<td>Node name not assigned to a network node</td>
</tr>
<tr>
<td>NSPX25: 602457</td>
<td>Illegal DECnet node number [NODE]</td>
</tr>
<tr>
<td>NSPX26: 602460</td>
<td>Table of topology watchers is full [NODE]</td>
</tr>
<tr>
<td>NTMX1: 602451</td>
<td>Network Management unable to complete request [NTMAN%]</td>
</tr>
</tbody>
</table>

| NTWZX1: 600737 | NET WIZARD capability required [ASNSQ] |
| ODTNX1: 600462 | Time zone must be USA or Greenwich [ODTNCC] |
| OPNX1: 600120 | File is already open [GNJFN; MTU%; OPENF; RLUFN; RNAME] |
| OPNX2: 600121 | File does not exist [GET; OPENF] |
| OPNX3: 600122 | Read access required [OPENF] |
| OPNX4: 600123 | Write access required [OPENF] |
| OPNX5: 600124 | Execute access required [OPENF] |
| OPNX6: 600125 | Append access required [OPENF] |
| OPNX7: 600126 | Device already assigned to another job [OPENF] |
| OPNX8: 600127 | Device is not on line [MTU%; OPENF] |
| OPNX9: 600130 | Invalid simultaneous access [OPENF; VACCT] |
| OPNX10: 600131 | Entire file structure full [OPENF] |
| OPNX11: 600133 | List access required [OPENF] |
| OPNX12: 600134 | Invalid access requested [OPENF] |
| OPNX13: 600135 | Invalid mode requested [OPENF] |
| OPNX14: 600136 | Read/write access required [OPENF] |
| OPNX15: 600137 | File has bad index block [OPENF; VACCT] |
| OPNX16: 600140 | No room in job for long file page table [OPENF] |
| OPNX17: 600141 | Unit Record Devices are not available [OPENF] |
| OPNX18: 600142 | IMP is not up |
| OPNX19: 600143 | Host is not up |
| OPNX20: 600144 | Connection refused |
| OPNX21: 600144 | Connection refused |
TPS-20 Monitor Calls Quick Reference Guide
TPS-20 JSYS Error Mnemonics

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 [ARCFL]
OPNX31: 602327 File is off-line [ARCFL; OPENF]
PDVX01: 601554 Address in .POADE must be as large as address in .POADR [PVDOFP]
PDVX02: 601555 Addresses in .PODAT block must be in strict ascending order [PVDOFP]
PDVX03: 601556 Address in .POADR must be a program data vector address [PVDOFP]
PKEEKX2: 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 [PMCTL]
PMCLX2: 602006 Requested physical page is unavailable [PMCTL]
PMCLX3: 602007 Requested physical page contains errors [PMCTL]
PMCLX4: 602165 No more error information [PMCTL]
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]
RCDIX1: 601376 Insufficient system resources [RCDIR]
RCDIX2: 601377 Invalid directory specification [ACCSES; RCDIR]
RCDIX3: 601400 Invalid structure name [RCDIR]
RCDIX4: 601401 Monitor internal error [RCDIR; RCUSR]
RCUSX1: 601402 Insufficient system resources [RCUSR]
RDTX1: 601010 Invalid string pointer [RDTTY; TEXTI; WILD%]
RIRX1: 602426 RIR JSYS incompatible with previous XSR [RIR]
RJFNX1: 600165 File is not closed [RLJFN]
RJFNX2: 600166 JFN is being used to accumulate filename [RLJFN]
RJFNX3: 600167 JFN is not accessible by this process [RLJFN]
RNAMX1: 600450 Files are not on same device [RNAMF]
RNAMX2: 600451 Destination file expunged [RNAMF]
| RNAMX3: 600452 | Write or owner access to destination file required [RNAMF] |
| RNAMX4: 600453 | Quota exceeded in destination of rename [RNAMF] |
| RNAMX5: 600750 | Destination file is not closed [RNAMF] |
| RNAMX6: 600751 | Destination file has bad page table [RNAMF] |
| RNAMX7: 600752 | Source file expunged [RNAMF] |
| RNAMX8: 600753 | Write or owner access to source file required [RNAMF] |
| RNAMX9: 600754 | Source file is nonexistent [RNAMF] |
| RNMX10: 600755 | Source file is not closed [RNAMF] |
| RNMX11: 600756 | Source file has bad page table [RNAMF] |
| RNMX12: 600757 | Illegal to rename to self [RNAMF] |
| RNMX13: 601454 | Insufficient system resources [RNAMF] |
| RSCNX1: 600361 | Overflowed rescan buffer, input string truncated [RSCAN] |
| RSCNX2: 600362 | Invalid function code [RSCAN] |
| RUNTX1: 600273 | Invalid process handle -3 or -4 [RUNTM] |
| SACTX1: 600530 | File is not on multiple-directory device [SACTF] |
| SACTX2: 600531 | Insufficient system resources (Job Storage Block full) [SACTF] |
| SACTX3: 600532 | Directory requires numeric account [SACTF] |
| SACTX4: 600533 | Write or owner access required [SACTF] |
| SAVX1: 601330 | Illegal to save files on this device [SAVE] |
| SCTX1: 601550 | Invalid function code [SCTTY] |
| SCTX2: 601551 | Terminal already in use as controlling terminal [SCTTY] |
| SCTX3: 601552 | Illegal to redefine the job’s controlling terminal [SCTTY] |
| SCTX4: 601553 | SC%SCT capability required [SCTTY] |
| SEVEX1: 600610 | Entry vector length is not less than 1000 [SEVEC; XSVEC%] |
| SFBSX1: 600210 | Illegal to change byte size for this opening of file [SFBSZ] |
| SFBSX2: 600211 | Invalid byte size [OPENF; SFBSZ] |
| SFPTX1: 600175 | File is not open [SFPR] |
| SFPTX2: 600176 | Illegal to reset pointer for this file [BKJFN; SFPR] |
| SFPTX3: 600177 | Invalid byte number [BKJFN; SFPR] |
| SFRVX1: 600377 | Invalid position in entry vector [SFRKV] |
| SFUSX1: 601373 | Invalid function [SFUST] |
| SFUSX2: 601374 | Insufficient system resources [SFUST] |
| SFUSX4: 601700 | File expunged [SFUST] |
| SFUSX5: 601701 | Write or owner access required [SFUST] |
| SFUSX6: 601702 | No such user name [SFUST] |
| SIRX1: 600570 | Table address is not greater than 20 [SIR; XSIR%] |
| SIRX2: 602425 | SIR JSYS invoked from non-zero section [SIR] |
| SJBX1: 601244 | Invalid function [SETJB] |
| SJBX2: 601245 | Invalid magtape density [SETJB] |
| SJBX3: 601246 | Invalid magtape data mode [SETJB] |
| SJBX4: 601251 | Invalid job number [SETJB] |
SJBX5: 601252 Job is not logged in [SETJB]
SJBX6: 601253 WHEEL or OPERATOR capability required [SETJB]
SJBX7: 602077 Remark exceeds 39 characters [SETJB]
SJBX8: 601455 Illegal to perform this function [SETJB]
SJPRI1: 601276 Job is not logged in [SJPRI]
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: 600260 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 [SPOOL]
SPLX2: 601145 Argument block too small [SPOOL]
SPLX3: 601146 Invalid device designator [SPOOL]
SPLX4: 601147 WHEEL or OPERATOR capability required [SPOOL]
SPLX5: 601150 Illegal to specify 0 as generation number
for first file [SPOOL]

601450 No directory to write spooled files into [SPOOL]

600742 Special network queue handle out of range [RCVIM; SNDIM]

600743 Special network queue not assigned [RCVIM; SNDIM]

600600 Illegal to save files on this device [GET; SSAVE]

600601 Page count (left half of table entry) must be negative [SSAVE]

601232 Insufficient system resources (Job Storage Block full) [SSAVE]

601233 Directory area of EXE file is more than one page [SSAVE]

601500 Number of PDVs grew during save [SSAVE]

600275 WHEEL or OPERATOR capability required [STAD]

600276 Invalid date or time [SFTAD; STAD]

602003 The STDIR JSYS has been replaced by RCDIR and RCUSR

600332 No such device [MSTR; PPNST; STDEV]

601436 Structure is not mounted [ACCES; DIRST; MSTR; PPNST; RCDIR; VACC]

601437 Insufficient system resources [ACCES; MSTR; STPPN]

601442 No such directory name [ACCES; STPPN]

601443 Ambiguous directory specification [ACCES; STPPN]

601747 No such user number [PPNST]

602142 Invalid user number [RCUSR]

602143 Invalid user name [RCUSR]

602222 Prior structure mount required [ACCES; GNJFN; GTJFN]

601414 Invalid terminal type [STTYP]

601406 Illegal to swap same JFN [SWJFN]

602242 Illegal to swap ATS JFN

601206 Unreasonable SPEAR block size [SYERR]

601207 No buffer space available for SPEAR [SYERR]

601235 Table is full [TBADD]

601236 Entry is already in table [TBADD]

601234 Table is empty [TBDEL]

601403 Invalid table entry location [TBDEL]

600350 Invalid terminal code [ATI; DTI]

600375 Invalid function code [TFORK]

600376 Unassigned process handle or not immediate inferior [TFORK]

600411 Process not frozen [TFORK]

600465 Invalid time format [IDTIM; IDTNC]

600460 Time cannot be greater than 24 hours [HSYS; IDCNV; ODCNV; ODTIM]

601302 Downtime cannot be more than 7 days in the future [HSYS]

601541 No system date and time [TIMER]

601157 Invalid function [TIMER]
<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMX2</td>
<td>601160 Invalid process handle [TIMER]</td>
</tr>
<tr>
<td>TIMX3</td>
<td>601161 Time limit already set [TIMER]</td>
</tr>
<tr>
<td>TIMX4</td>
<td>601162 Illegal to clear time limit [TIMER]</td>
</tr>
<tr>
<td>TIMX5</td>
<td>601404 Invalid software interrupt channel number [TIMER]</td>
</tr>
<tr>
<td>TIMX6</td>
<td>601535 Time has already passed [TIMER]</td>
</tr>
<tr>
<td>TIMX7</td>
<td>601536 No space available for a clock [TIMER]</td>
</tr>
<tr>
<td>TIMX8</td>
<td>601537 User clock allocation exceeded [TIMER]</td>
</tr>
<tr>
<td>TIMX9</td>
<td>601540 No such clock entry found [TIMER]</td>
</tr>
<tr>
<td>TLNXX1</td>
<td>600351 Illegal to set remote to object before object to remote [TLINK]</td>
</tr>
<tr>
<td>TLNXX2</td>
<td>600356 Link was not received within 15 seconds [TLINK]</td>
</tr>
<tr>
<td>TLNXX3</td>
<td>600357 Links full [TLINK]</td>
</tr>
<tr>
<td>TLUKX1</td>
<td>601237 Internal format of table is incorrect [TBLUK]</td>
</tr>
<tr>
<td>TMONX1</td>
<td>601247 Invalid TMON function [TMON]</td>
</tr>
<tr>
<td>TTMSX1</td>
<td>602432 Could not send message within timeout interval [TTMSX1]</td>
</tr>
<tr>
<td>TTYXO1</td>
<td>602030 Line is not active [BKJFN; CFIBF; CFOBF; DIBE; DOBE; GTTYP; MTPR; OPENF; RFCOC; RFMOD; RFPOS; SFCOC; SFMOD; SFPOS; SIBE; SOBE; STI; STO; STPAR; STTYP; TLINK]</td>
</tr>
<tr>
<td>TTYXO2</td>
<td>602455 Illegal character specified</td>
</tr>
<tr>
<td>TTYX1</td>
<td>600360 Device is not a terminal [STI; STD]</td>
</tr>
<tr>
<td>UFPGX1</td>
<td>601316 File is not opened for write [UFPGS]</td>
</tr>
<tr>
<td>USGX01</td>
<td>602113 Invalid USAGE entry type code [USAGE]</td>
</tr>
<tr>
<td>USGX02</td>
<td>602116 Item not found in argument list [USAGE]</td>
</tr>
<tr>
<td>USGX03</td>
<td>602124 Default item not allowed [USAGE]</td>
</tr>
<tr>
<td>UTSTX1</td>
<td>602013 Invalid function code [UTEST]</td>
</tr>
<tr>
<td>UTSTX2</td>
<td>602014 Area of code too large to test [UTEST]</td>
</tr>
<tr>
<td>UTSTX3</td>
<td>602015 UTEST facility in use by another process [UTEST]</td>
</tr>
<tr>
<td>VACCX0</td>
<td>602111 Invalid account [CACCT; SACTF; VACCT]</td>
</tr>
<tr>
<td>VACCX1</td>
<td>602112 Account string exceeds 39 characters [CACCT; SACTF; VACCT; COMND]</td>
</tr>
<tr>
<td>VACCX2</td>
<td>602126 Account has expired [SACTF; VACCT]</td>
</tr>
<tr>
<td>WHEHLX1</td>
<td>600614 WHEEL or OPERATOR capability required [DELF; DSKAS; DSKOP; MDDT%; SUPRI; SPRIM; STI]</td>
</tr>
<tr>
<td>WILDX1</td>
<td>601460 Second JFN cannot be wild [WILD%]</td>
</tr>
<tr>
<td>XSEVX1</td>
<td>602472 Illegal entry vector type [XSSEV%; XGSEV%]</td>
</tr>
<tr>
<td>XSEVX2</td>
<td>602473 Invalid entry vector length [XSSEV%]</td>
</tr>
<tr>
<td>XSEVX3</td>
<td>602474 Cannot get extended values with this monitor call [GCVEC; GDVEC]</td>
</tr>
<tr>
<td>XSIRX1</td>
<td>602424 Channel table crosses section boundary [XSIR%]</td>
</tr>
<tr>
<td>XSIRX2</td>
<td>602427 Level table crosses section boundary [XSIR%]</td>
</tr>
<tr>
<td>ZONEX1</td>
<td>600461 Time zone out of range [IDCNV; ODCNV; ODTNC]</td>
</tr>
</tbody>
</table>
**POINTER FORMATS**

**One Word Global Byte Pointer**

<table>
<thead>
<tr>
<th>P,S</th>
<th>SEC</th>
<th>ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5,6</td>
<td>17,18</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</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>53</td>
</tr>
<tr>
<td>7</td>
<td>44</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>66</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>73</td>
</tr>
<tr>
<td>18</td>
<td>44</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>76</td>
</tr>
</tbody>
</table>

**Global Format Indirect Word**

<table>
<thead>
<tr>
<th>I</th>
<th>X</th>
<th>SEC</th>
<th>ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>12</td>
<td>5,6</td>
<td>17,18</td>
</tr>
</tbody>
</table>

---

203
TOPS-20 Monitor Calls Quick Reference Guide
Pointer Formats

Two-word Local Byte Pointer

| 0 | 5|6 | 11| 12| 13| 17|18 | 35 |
|---|---|---|---|---|---|---|---|
| P | S | 1 | Reserved | Available to User |

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th></th>
<th>Reserved</th>
<th>I</th>
<th>X</th>
<th>ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1</td>
<td>2</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>17</td>
</tr>
</tbody>
</table>

Local Format Indirect Word

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th></th>
<th>Reserved</th>
<th>I</th>
<th>X</th>
<th>ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1</td>
<td>2</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>17</td>
</tr>
</tbody>
</table>

One-word Local Byte Pointer

| 0 | 5|6 | 11| 12| 13| 14| 17|18 | 35 |
|---|---|---|---|---|---|---|---|---|
| P | S | O | I | X | ADDR |

Two-word Global Byte Pointer

| 0 | 5|6 | 11| 12| 13| 17|18 | 35 |
|---|---|---|---|---|---|---|---|
| P | S | 1 | Reserved | Available to User |

<table>
<thead>
<tr>
<th>0</th>
<th>I</th>
<th>X</th>
<th>SEC</th>
<th>ADDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
PDP-10 INSTRUCTION SET

**Arithmetic Testing Instructions**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADBJP</td>
<td>252</td>
<td>(AC) + 1, 1 --&gt; (AC); If AC &gt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADBJN</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>CAIN</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>CAME</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>CAMGE</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>JUML</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>SKIPLE</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>SKIPN</td>
<td>336</td>
<td>If (E) ≠ 0: skip; If AC ≠ 0: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>SKIPG</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>ADJLE</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>ADJGE</td>
<td>345</td>
<td>(AC) + 1 --&gt; (AC); If (AC) &gt;= 0: E --&gt; (PC)</td>
</tr>
<tr>
<td>ADJN</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>
TOPS-20 Monitor Calls Quick Reference Guide
PDP-10 Instruction Set

AOS  350  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC)
AOSL 351  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
        If (E) < 0: skip
AOSE 352  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
        If (E) = 0: skip
AOSLE 353  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
        If (E) ≤ 0: skip
AOSA  354  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
AOSGE 355  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
        If (E) ≥ 0: skip
AOSN  356  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
        If (E) ≠ 0: skip
AOSG  357  (E) + 1 -->(E); If (AC) ∉ O: (E) -->(AC);
        If (E) > 0: skip
SDJ  360  (AC) - 1 -->(AC)
SDJL 361  (AC) - 1 -->(AC); If (AC) < O: E -->(PC)
SDJE  362  (AC) - 1 -->(AC); If (AC) = O: E -->(PC)
SDJLE 363  (AC) - 1 -->(AC); If (AC) ≤ O: E -->(PC)
SDJA  364  (AC) - 1 -->(AC); E -->(PC)
SDJGE 365  (AC) - 1 -->(AC); If (AC) ≥ O: E -->(PC)
SDJN  366  (AC) - 1 -->(AC); If (AC) ≠ O: E -->(PC)
SDJG  367  (AC) - 1 -->(AC); If (AC) > O: E -->(PC)
SOS  370  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC)
SOSL  371  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) < 0: skip
SOSE  372  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) = 0: skip
SOSLE 373  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) ≤ 0: skip
SOSA  374  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) ≥ 0: skip
SOSGE 375  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) ≥ 0: skip
SOSN  376  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) ≠ 0: skip
SOSG  377  (E) - 1 -->(E); If AC ∉ O: (E) -->(AC);
        If (E) > 0: 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)
SETMN 416  (E) -->(E) [no-op]
SETMB 417  (E) -->(AC)(E)
SETA  424  (AC) -->(AC) [no-op]
SETAI 425  (AC) -->(AC) [no-op]
SETAM 426  (AC) -->(E)

206
SETAB 427  (AC) --> (E)
SETQ  474  7777777777777777 --> (AC)
SETOI 475  7777777777777777 --> (AC)
SETOM 476  7777777777777777 --> (E)
SETOB 477  7777777777777777 --> (AC)(E)
SETCA 450  ¬(AC) --> (AC)
SETCAI 451  ¬(AC) --> (AC)
SETCAM 452  ¬(AC) --> (E)
SETCAB 453  ¬(AC) --> (AC)(E)
SETCM  460  ¬(E) --> (AC)
SETCGI 461  ¬[O,E] --> (AC)
SETCMII 462  ¬(E) --> (E)
SETCMII 463  ¬(E) --> (AC)(E)

AND  404  (AC) & (E) --> (AC)
ANDI  405  (AC) & 0,E --> (AC)
ANDM  406  (AC) & (E) --> (E)
ANDB  407  (AC) & (E) --> (AC)(E)
ANDCA 410  ¬(AC) & (E) --> (AC)
ANDCAI 411  ¬(AC) & 0,E --> (AC)
ANDCAB 412  ¬(AC) & (E) --> (E)
ANDCM 420  (AC) & ¬(E) --> (AC)
ANDCMI 421  (AC) & ¬[O,E] --> (AC)
ANDCCM 422  (AC) & (E) --> (E)
ANDCMII 420  (AC) & ¬(E) --> (AC)(E)
ANDCMII 420  (AC) & ¬(E) --> (AC)(E)

IOR  434  (AC) !(E) --> (AC)
IDRI  435  (AC) !(0,E) --> (AC)
IDRM  436  (AC) !(E) --> (E)
IDRB  437  (AC) !(E) --> (AC)(E)

ORCA  454  ¬(AC) !(E) --> (AC)
ORCAI  455  ¬(AC) !(0,E) --> (AC)
ORCAM  456  ¬(AC) !(E) --> (E)
ORCAB  457  ¬(AC) !(E) --> (AC)(E)
ORCM  464  (AC) !(E) --> (AC)
ORCMI  465  (AC) !(0,E] --> (AC)
ORCMII  466  (AC) !(E) --> (E)
ORCMII  466  (AC) !(E) --> (AC)(E)
ORCB  470  ¬(AC) !(E) --> (AC)
ORCBI  471  ¬(AC) !(0,E] --> (AC)
ORCBM  472  ¬(AC) !(E) --> (E)
ORCBB  473  ¬(AC) !(E) --> (AC)(E)

XOR  430  (AC) ^! (E) --> (AC)
XORI  431  (AC) ^! 0,E --> (AC)
XORM  432  (AC) ^! (E) --> (E)
XORB  433  (AC) ^! (E) --> (AC)(E)

EQV  445  ¬[(AC) ^! (E)] --> (AC)
EQVI  445  ¬[(AC) ^! 0,E] --> (AC)
TOPS-20 Monitor Calls Quick Reference Guide
PDP-10 Instruction Set

EQVM 446 \( \neg[(AC) \land (E)] \rightarrow (E) \)
EQVB 447 \( \neg[(AC) \land (E)] \rightarrow (AC)(E) \)

**Byte Instructions**

IBP 133 Linear operations on pointer in E or E,E+1
AC=0
if P-S >= 0: P-S --> P
if P-S < 0: Y+1 --> Y; 36-S --> P

ADJBP 133 Array operations on pointer in E or E,E+1
AC ≠ 0
Let A = REMAINDER \((36-P)/S\)
If S > 36-A: 1 --> NO DIVIDE
If S = 0: (E) --> (AC) or
(E,E+1) --> (AC,AC+1)
If 0 < S < 36-A: make copy C of (E) or
(E,E+1)
Compute: \((AC)+((36-P)/S) = Q \times \text{BYTES/WORD} + R\)
1 <= R <= \text{BYTES/WORD} = \((36-P)/S\) + (P/S)
Y(C) + Q --> Y(C)
36 - (R \times S) - A --> P(C)
C --> (AC) or (AC,AC+1)

LDB 135 BYTE IN ((E)) --> (AC)
DPB 137 BYTE IN (AC) --> BYTE IN ((E))
ILDDB 134 IBP and LDB
IDPB 136 IBP and DPB

**Fixed-point Arithmetic Instructions**

ADD 270 (AC) + (E) --> (AC)
ADDI 271 (AC) + O,E --> (AC)
ADDM 272 (AC) + (E) --> (E)
ADDB 273 (AC) + (E) --> (AC)(E)

SUB 274 (AC) - (E) --> (AC)
SUBI 275 (AC) - O,E --> (AC)
SUBM 276 (AC) - (E) --> (E)
SUBB 277 (AC) - (E) --> (AC)(E)

IMUL 220 (AC) * (E) --> (AC) \[1\]
IMULI 221 (AC) * O,E --> (AC) \[1\]
IMULM 222 (AC) * (E) --> (E) \[1\]
IMULB 223 (AC) * (E) --> (AC)(E) \[1\]

MUL 224 (AC) * (E) --> (AC,AC+1)
MULI 225 (AC) * O,E --> (AC,AC+1)
MULM 226 (AC) * (E) --> (E) \[2\]
MULB 227 (AC) * (E) --> (AC,AC+1)(E)

IDIV 230 (AC) / (E) --> (AC); REMAINDER --> (AC+1)
\[1\] High order word of product is discarded.
\[2\] LOW order word of product is discarded.
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDIVI</td>
<td>231</td>
<td>(AC) / O,E → (AC); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>IDIVM</td>
<td>232</td>
<td>(AC) / (E) → (E); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>IDIVB</td>
<td>233</td>
<td>(AC) / (E) → (AC)(E); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>DIV</td>
<td>234</td>
<td>(AC,AC+1) / (E) → (AC); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>DIVI</td>
<td>235</td>
<td>(AC,AC+1) / O,E → (AC); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>DIVM</td>
<td>236</td>
<td>(AC,AC+1) / (E) → (E); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>DIVB</td>
<td>237</td>
<td>(AC,AC+1) / (E) → (AC)(E); REMAINDER → (AC+1)</td>
</tr>
<tr>
<td>DADD</td>
<td>114</td>
<td>(AC,AC+1) + (E,E+1) → (AC,AC+1)</td>
</tr>
<tr>
<td>DSUB</td>
<td>115</td>
<td>(AC,AC+1) - (E,E+1) → (AC,AC+1)</td>
</tr>
<tr>
<td>DMUL</td>
<td>116</td>
<td>(AC,AC+1) * (E,E+1) → (AC,AC+1,AC+2,AC+3)</td>
</tr>
<tr>
<td>DDIV</td>
<td>117</td>
<td>(AC,AC+1,AC+2,AC+3) / (E,E+1) → (AC,AC+1)</td>
</tr>
</tbody>
</table>

### Floating-point Arithmetic Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAD</td>
<td>140</td>
<td>(AC) + (E) → (AC)</td>
</tr>
<tr>
<td>FADL</td>
<td>141</td>
<td>(AC) + (E) → (AC,AC+1)</td>
</tr>
<tr>
<td>FADM</td>
<td>142</td>
<td>(AC) + (E) → (E)</td>
</tr>
<tr>
<td>FADB</td>
<td>143</td>
<td>(AC) + (E) → (AC)(E)</td>
</tr>
<tr>
<td>FADR</td>
<td>144</td>
<td>(AC) + (E) → (AC)</td>
</tr>
<tr>
<td>FADRI</td>
<td>145</td>
<td>(AC) + E,O → (AC)</td>
</tr>
<tr>
<td>FADRM</td>
<td>146</td>
<td>(AC) + (E) → (E)</td>
</tr>
<tr>
<td>FADRB</td>
<td>147</td>
<td>(AC) + (E) → (AC)(E)</td>
</tr>
<tr>
<td>FSB</td>
<td>150</td>
<td>(AC) - (E) → (AC)</td>
</tr>
<tr>
<td>FSBL</td>
<td>151</td>
<td>(AC) - (E) → (AC,AC+1)</td>
</tr>
<tr>
<td>FSBM</td>
<td>152</td>
<td>(AC) - (E) → (E)</td>
</tr>
<tr>
<td>FSBB</td>
<td>153</td>
<td>(AC) - (E) → (AC)(E)</td>
</tr>
<tr>
<td>FSBR</td>
<td>154</td>
<td>(AC) - (E) → (AC)</td>
</tr>
<tr>
<td>FSBRI</td>
<td>155</td>
<td>(AC) - E,O → (AC)</td>
</tr>
<tr>
<td>FSBRM</td>
<td>156</td>
<td>(AC) - (E) → (E)</td>
</tr>
<tr>
<td>FSBRB</td>
<td>157</td>
<td>(AC) - (E) → (AC)(E)</td>
</tr>
<tr>
<td>FMP</td>
<td>160</td>
<td>(AC) * (E) → (AC)</td>
</tr>
<tr>
<td>FMPL</td>
<td>161</td>
<td>(AC) * (E) → (AC,AC+1)</td>
</tr>
<tr>
<td>FMPM</td>
<td>162</td>
<td>(AC) * (E) → (E)</td>
</tr>
<tr>
<td>FMPB</td>
<td>163</td>
<td>(AC) * (E) → (AC)(E)</td>
</tr>
<tr>
<td>FMPR</td>
<td>164</td>
<td>(AC) * (E) → (AC)</td>
</tr>
<tr>
<td>FMPRI</td>
<td>165</td>
<td>(AC) * E,O → (AC)</td>
</tr>
<tr>
<td>FMPRM</td>
<td>166</td>
<td>(AC) * (E) → (E)</td>
</tr>
<tr>
<td>FMPRB</td>
<td>167</td>
<td>(AC) * (E) → (AC)(E)</td>
</tr>
<tr>
<td>FDV</td>
<td>170</td>
<td>(AC) / (E) → (AC)</td>
</tr>
<tr>
<td>FDVL</td>
<td>171</td>
<td>(AC) / (E) → (AC) Remainder → (AC+1)</td>
</tr>
<tr>
<td>FDVM</td>
<td>172</td>
<td>(AC) / (E) → (E)</td>
</tr>
<tr>
<td>FDVB</td>
<td>173</td>
<td>(AC) / (E) → (AC)(E)</td>
</tr>
<tr>
<td>FDVR</td>
<td>174</td>
<td>(AC) / (E) → (AC)</td>
</tr>
<tr>
<td>FDVRI</td>
<td>175</td>
<td>(AC) / E,O → (AC)</td>
</tr>
<tr>
<td>FDVRM</td>
<td>176</td>
<td>(AC) / (E) → (E)</td>
</tr>
<tr>
<td>Instruction</td>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>FDVRB</td>
<td>177</td>
<td>(AC) / (E) -&gt; (AC)(E)</td>
</tr>
<tr>
<td>UFA</td>
<td>130</td>
<td>(AC) + (E) -&gt; (AC+1) without normalization</td>
</tr>
<tr>
<td>DFN</td>
<td>131</td>
<td>-(AC,E) -&gt; (AC,E)</td>
</tr>
<tr>
<td>FSC</td>
<td>132</td>
<td>(AC) * 2**E -&gt; (AC)</td>
</tr>
<tr>
<td>GFSC</td>
<td>031</td>
<td>(AC,AC+1) * 2**E -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>FLTR</td>
<td>127</td>
<td>(E) floated, rounded -&gt; (AC)</td>
</tr>
<tr>
<td>GFLTR</td>
<td>030</td>
<td>(E) floated, rounded -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DGFLTR</td>
<td>027</td>
<td>(E,E+1) floated, rounded -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>FIX</td>
<td>122</td>
<td>(E) fixed -&gt; (AC)</td>
</tr>
<tr>
<td>FIXR</td>
<td>126</td>
<td>(E) fixed, rounded -&gt; (AC)</td>
</tr>
<tr>
<td>GIFX</td>
<td>024</td>
<td>(E,E+1) fixed -&gt; (AC)</td>
</tr>
<tr>
<td>GIFXR</td>
<td>026</td>
<td>(E,E+1) fixed, rounded -&gt; (AC)</td>
</tr>
<tr>
<td>GDFIX</td>
<td>023</td>
<td>(E,E+1) fixed -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>GDFIXR</td>
<td>025</td>
<td>(E,E+1) fixed, rounded -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>GSNGL</td>
<td>021</td>
<td>(E,E+1) converted -&gt; (AC)</td>
</tr>
<tr>
<td>GDBLE</td>
<td>022</td>
<td>(E) converted -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DFAD</td>
<td>110</td>
<td>(AC,AC+1) + (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DFSB</td>
<td>111</td>
<td>(AC,AC+1) - (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DFMP</td>
<td>112</td>
<td>(AC,AC+1) * (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>DFDV</td>
<td>113</td>
<td>(AC,AC+1) / (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>GFAD</td>
<td>102</td>
<td>(AC,AC+1) + (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>GFSB</td>
<td>103</td>
<td>(AC,AC+1) - (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>GFMP</td>
<td>106</td>
<td>(AC,AC+1) * (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
<tr>
<td>GFDV</td>
<td>107</td>
<td>(AC,AC+1) / (E,E+1) -&gt; (AC,AC+1)</td>
</tr>
</tbody>
</table>

### Fullword Instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCH</td>
<td>250</td>
<td>(AC) &lt;--&gt; (E)</td>
</tr>
<tr>
<td>MOVE</td>
<td>200</td>
<td>(E) -&gt; (AC)</td>
</tr>
<tr>
<td>MOVEI</td>
<td>201</td>
<td>O,E -&gt; (AC)</td>
</tr>
<tr>
<td>MOVEM</td>
<td>202</td>
<td>(AC) -&gt; (E)</td>
</tr>
<tr>
<td>MOVES</td>
<td>203</td>
<td>If AC ≠ O: (E) -&gt; (AC)</td>
</tr>
<tr>
<td>MOVs</td>
<td>204</td>
<td>(E)S--&gt; (AC)</td>
</tr>
<tr>
<td>MOVSI</td>
<td>205</td>
<td>E,O -&gt; (AC)</td>
</tr>
<tr>
<td>MOVSM</td>
<td>206</td>
<td>(AC)S --&gt; (E)</td>
</tr>
<tr>
<td>MOVSS</td>
<td>207</td>
<td>(E)S --&gt; (E) If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>MOVN</td>
<td>210</td>
<td>-(E) -&gt; (AC)</td>
</tr>
<tr>
<td>MOVNI</td>
<td>211</td>
<td>-[O,E] --&gt; (AC)</td>
</tr>
<tr>
<td>MOVNM</td>
<td>212</td>
<td>-(AC) --&gt; (E)</td>
</tr>
<tr>
<td>MOVNS</td>
<td>213</td>
<td>-(E) --&gt; (E) If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>MOVm</td>
<td>214</td>
<td></td>
</tr>
<tr>
<td>MOVmi</td>
<td>215</td>
<td>O,E --&gt; (AC)</td>
</tr>
<tr>
<td>MOVMM</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>MOVMS</td>
<td>217</td>
<td></td>
</tr>
<tr>
<td>XMoven</td>
<td>415</td>
<td>E --&gt; (AC) Non-local AC reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,E --&gt; (AC) Local AC reference</td>
</tr>
</tbody>
</table>
### TOPS-20 Monitor Calls Quick Reference Guide
#### PDP-10 Instruction Set

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Number</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) --> ((AC)R)
```

#### XBLT 020
Move |(AC)| words

- If (AC) > 0:
  - start with ((AC+1) --> ((AC+2)) and go up
- If (AC) < 0:
  - start with ((AC-1)-1) --> ((AC+2)-1)
  - and go down

---

### Halfword Instructions (Source Left)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLL</td>
<td>500</td>
<td>(E)L --&gt; (AC)L</td>
</tr>
<tr>
<td>HLLI</td>
<td>501</td>
<td>O --&gt; (AC)L</td>
</tr>
<tr>
<td>HLLM</td>
<td>502</td>
<td>(AC)L --&gt; (E)L</td>
</tr>
<tr>
<td>HLLS</td>
<td>503</td>
<td>If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HLLZ</td>
<td>510</td>
<td>(E)L,O --&gt; (AC)</td>
</tr>
<tr>
<td>HLLZI</td>
<td>511</td>
<td>O --&gt; (AC)</td>
</tr>
<tr>
<td>HLLZM</td>
<td>512</td>
<td>(AC)L,O --&gt; (E)</td>
</tr>
<tr>
<td>HLLZS</td>
<td>513</td>
<td>O --&gt; (E)R</td>
</tr>
<tr>
<td>HLO</td>
<td>520</td>
<td>(E)L,777777 --&gt; (AC)</td>
</tr>
<tr>
<td>HLOI</td>
<td>521</td>
<td>O,777777 --&gt; (AC)</td>
</tr>
<tr>
<td>HLOM</td>
<td>522</td>
<td>(AC)L,777777 --&gt; (E)</td>
</tr>
<tr>
<td>HLOS</td>
<td>523</td>
<td>777777 --&gt; (E)R; If AC ≠ O: (E) --&gt; (AC)</td>
</tr>
<tr>
<td>HLE</td>
<td>530</td>
<td>(E)L,[(E)O * 777777] --&gt; (AC)</td>
</tr>
<tr>
<td>HLEI</td>
<td>531</td>
<td>O --&gt; (AC)</td>
</tr>
<tr>
<td>HLEM</td>
<td>532</td>
<td>(AC)L,[(AC)O * 777777] --&gt; (E)</td>
</tr>
<tr>
<td>HLES</td>
<td>533</td>
<td>(E)O * 777777 --&gt; (E)R;</td>
</tr>
</tbody>
</table>
  - If AC ≠ O: (E) --> (AC) |
| HLR | 544 | (E)L --> (AC)R |
| HLRI | 545 | O --> (AC)R |
| HLRM | 546 | (AC)L --> (E)R |
| HLRS | 547 | (E)L --> (E)R; If AC ≠ O: (E) --> (AC) |
| HLRO | 554 | O,(E)L --> (AC) |
| HLROI | 555 | O --> (AC) |
| HLRZ | 556 | O,(AC)L --> (E) |
| HLRZI | 557 | O,(E)L --> (E); If AC ≠ O: (E) --> (AC) |
| HLRZM | 558 | 777777, (E)L --> (AC) |
| HLRZS | 559 | 777777, (AC)L --> (E) |
| HLROS | 560 | 777777, (E)L --> (E); If AC ≠ O: (E) --> (AC) |
| HLRE | 574 | [(E)O * 777777],(E)L --> (AC) |
| HLREI | 575 | O --> (AC) |
| HLRM | 576 | [(AC)O * 777777],(AC)L --> (E) |
| HLRSE | 577 | [(E)O * 777777],(E)L --> (E); |
  - If AC ≠ O: (E) --> (AC) |
| XHLLI | 501 | If zero section: E(L) --> (AC)L |

If non-zero section:

211
TOPS-20 Monitor Calls Quick Reference Guide
PDP-10 Instruction Set

1. 0 --> (AC) bits 0-5
2. section # --> (AC) bits 6-17; if E is a local AC address, section # = 1

Halfword Instructions (Source Right)

HRR  540  (E)R -->(AC)R
HRRI 541   E -->(AC)R
HRRM 542  (AC)R -->(E)R
HRRS  543  If AC ≠ O: (E) --> (AC)
HRRZ  550  O,(E)R -->(AC)
HRRZI 551   O,E -->(AC)
HRRZM 552  O,(AC)R -->(E)
HRRZS  553  O -->(E)L
HRRO  560  777777,(E)R -->(AC)
HRROI 561  777777,E -->(AC)
HRROM 562  777777,(AC)R -->(E)
HRROS  563  777777 -->(E)L
HRRE  570  [(E)18 * 777777],(E)R -->(AC)
HRREI 571  [E18 * 777777],E -->(AC)
HRREM 572  [(AC)18 * 777777],(AC)R -->(E)
HRRES  573  (E)18 * 777777 -->(E)L;
If AC ≠ O: (E) -->(AC)
HRL  504  (E)R -->(AC)L
HRLI  505  E -->(AC)L
HRLM  506  (AC)R -->(E)L
HRLS  507  (E)R -->(E)L; If AC ≠ O: (E) -->(AC)
HRLZ  514  (E)R,O -->(AC)
HRLM  515  E,O -->(AC)
HRLZM 516  (AC)R,O -->(E)
HRLZS  517  (E)R,O -->(E); If AC ≠ O: (E) -->(AC)
HRLD  524  (E)R,777777 -->(AC)
HRLMO 525  E,777777 -->(AC)
HRLM  526  (AC)R,777777 -->(E)
HRLDS 527  (E)R,777777 -->(E); If AC ≠ O: (E) -->(AC)
HRLE  534  (E)R,[((E)18 *777777] -->(AC)
HRLEI  535  E,[E18 * 777777] -->(AC)
HRLEM  536  (AC)R,[((AC)18 * 777777] -->(E)
HRLES  537  (E)R,[((E)18 * 777777] -->(E);
If AC ≠ O: (E) -->(AC)

I/O Instructions

DATAO  70014  (E) --> DATA
DATAI  70004  DATA --> (E)
CONO  70020  E --> COMMAND
CONI  70024  STATUS --> (E)
CONSZ  70030  If STATUS(R) & E = O: skip
CONSO  70034  If STATUS(R) & E ≠ O: skip
BLKI  70000  (E) + 1,1 --> (E); DATA --> ((E)R);
If (E)L ≠ O: skip

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
TLZCE 623  If (AC)L & E = 0: skip; (AC)L & ¬E --> (AC)L
TLZD  625  (AC)L & ¬E --> (AC)L; skip
TLZN  627  If (AC)L & E ≠ 0: skip; (AC)L & ¬E --> (AC)L
TLC  641  (AC)L ! E --> (AC)L
TLC  643  If (AC)L & E = 0: skip; (AC)L ! E --> (AC)L
TLCB  645  (AC)L ! E --> (AC)L; skip
TLCN  647  If (AC)L & E ≠ 0: skip; (AC)L ! E --> (AC)L
TLDO 661  (AC)L ! E --> (AC)L
TLDOE 663  If (AC)L & E = 0: skip; (AC)L ! E --> (AC)L
TLDOA  665  (AC)L ! E --> (AC)L; skip
TLDON 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
TRZCE 622  If (AC)R & E = 0: skip; (AC)R & ¬E --> (AC)R
TRZD  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
TRCB  644  (AC)R ! E --> (AC)R; skip
TRCN  646  If (AC)R & E ≠ 0: skip; (AC)R ! E --> (AC)R
TRO  660  (AC)R ! E --> (AC)R
TROE 662  If (AC)R & E = 0: skip; (AC)R ! E --> (AC)R
TROA  664  (AC)R ! E --> (AC)R; skip
TRON 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)
TDC  632  If (AC) & (E) = 0: skip;
(TC) & ¬(E) --> (AC)
TDCZ  634  (AC) & ¬(E) --> (AC); skip
TDCZ  636  If (AC) & (E) ≠ 0: skip; (AC) & ¬(E) --> (AC)
TDC  650  (AC) ! (E) --> (AC)
TDC  652  If (AC) & (E) = 0: skip; (AC) ! (E) --> (AC)
TDC  654  (AC) ! (E) --> (AC); skip
TDCN 656  If (AC) & (E) ≠ 0: skip; (AC) ! (E) --> (AC)
TDO  670  (AC) ! (E) --> (AC)
TDOE 672  If (AC) & (E) = 0: skip; (AC) ! (E) --> (AC)
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Hex</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDDA</td>
<td>674</td>
<td>(AC) ! (E) --&gt; (AC); skip</td>
</tr>
<tr>
<td>TDDN</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>TSZN</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>TSCE</td>
<td>653</td>
<td>If (AC) &amp; (E)S = 0: skip; (AC) ! (E)S --&gt; (AC)</td>
</tr>
<tr>
<td>TSCA</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**

- **XCT** 256 Execute (E)
- **JFFO** 243 If (AC) = O: O --> (AC+1) If (AC) ≠ O: E --> (PC)
- **JFCL** 255 If AC & FLAGS ≠ 0: E --> (PC); ¬AC & FLAGS --> FLAGS
- **JRST** 25400 E --> (PC)
- **PORTAL** 25404 O --> PUBLIC; E --> (PC)
- **JRSTF** 25410 (X)L or (Y)L --> FLAGS; E --> (PC)
- **HALT** 25420 E --> (PC); stop
- **XJRSTF** 25424 (E)L --> FLAGS; (E+1) --> (PC)
- **XJEN** 25430 Dismiss PI; (E)L --> FLAGS; (E+1) --> (PC)
- **XPCW** 25434 FLAGS,O --> (E); PC+1 --> (E+1); (E+2)L --> FLAGS; (E+3) --> (PC)
- **JEN** 25450 Dismiss PI; (X)L or (Y)L --> FLAGS; E --> (PC)
- **SFM** 25460 FLAGS,O --> (E)
- **JSR** 264 If PC(L) = O: FLAGS,PC(R)+1 --> (E); E+1 --> (PC) If PC(L) ≠ O: PC+1 --> (E); E+1 --> (PC)
- **JSP** 265 If PC(L) = O: FLAGS,PC(R)+1 --> (AC); E --> (PC)
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)

ROT 241
Rotate (AC) E places

LSH 242
Shift (AC) E places

ASHC 244
(AC,AC+1) * 2**E --> (AC,AC+1)

ROTC 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)

ADJSB 105
If PC(L) = 0 or (AC)0,6-17 <= 0:
(AC) + [+-]E(R),E(R) --> (AC)
If PC(L) ≠ 0 and (AC)0,6-17 > 0:
(AC) + [+-]E(R) --> (AC)
MACRO-20 PSEUDO-OPS

Pseudo-op/Arguments

ARRAY addr[expr]

ASCII dtextd

ASIZ dtextd

.ASSIGN sym1,sym2,increment

ASUPPRESS

BLOCK expr

BYTE (n)expr

COMMENT dtextd

.COMMON symbol[expr]

.CREF

DEC expr,...,expr

DEFINE macro(dummyarg)<macrotex>

DEPHASE

.DIRECTIVE directive,...,directive

Function

Reserves a block of storage with length expr at address addr

Enters ASCII text; d is any delimiter not in text

Enters ASCII text with guaranteed trailing null; d is any delimiter not in text

Assigns value of sym2 to sym1 and adds increment to sym2

Causes all local or INTERNAL symbols not referenced after ASUPPRESS to be deleted from symbol table

Reserves a block of length expression

Stores value of expression in n-bit bytes

Treats text as comment; d is any delimiter not in text

Defines FORTRAN or FORTRAN-compatible COMMON block

Resumes output of suspended cross-referencing

Defines local radix as decimal

Defines macro macro

Suspends effect of PHASE pseudo-op

Sets switches to enable/disable MACRO features

.ITABM Include spaces and tabs in passed arguments

.XTABM Strip leading/trailing spaces and tabs from passed arguments

MACMPD Match paired delimiters in MACRO call

LITLST List binary code for in-line literals
END expr
.ENDPS

ENTRY symbol,...,symbol

EXP expr,...,expr

EXTERN symbol,...,symbol

.HWFRMT

.IF expr,qualifier,<code>

.IFN expr,qualifier,<code>

IFx expr,<code>

E
N
G
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 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 only 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's complement (expr1)<,<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 (})
MLON
.NODDT symbol,...,symbol

NOSYM

OCT expr,...,expr

OPDEF symbol[expr]

.ORG addr

PAGE

PASS2

PHASE addr

POINT bytesize,addr,.bitplace

PRGEND

PRINTX text

.PSECT name/attribute,origin

PURGE symbol,...,symbol

RADIX n
RADIX50 code,symbol

RELOC expr

REMARK text
REPEAT n,<code>
.REQUEST filespec

.REQUIRE filespec

SALL

SEARCH tabnam(filespec)

Suspended MLOGF
Suppresses DDT
recognition of symbol
Suppress listing of
symbol table in listing
file
Defines radix of expr as
octal
Defines symbol as
equivalent to expr
Sets location counter to
addr
Causes assembler to skip
to top of next listing
page
Switches assembler to
Pass 2 processing of
remaining code
Assembles part of program
so that it can be moved
to other location for
execution
Generates a byte pointer
for machine byte
instructions
Replaces END for all but
last program in
multi-program assembly
Causes text to be output
during assembly to TTY
and/or listing device
Specifies relocation
counter for code
following
Deletes symbol from
symbol table
Sets radix to value of n
Packs symbol into B4-35
of storage word, with
code in BO-3
Sets location counter to
value of expr and assigns
relocatable addresses to
code following
Marks text as comment
Generates code n times
Causes file filespec to
be loaded to satisfy a
global request
Causes file filespec to
be loaded automatically
Causes suppression of all
macro and REPEAT
expansions
Defines list of symbol
tables to be searched
SIXBIT dtextd

SQUEZE code,symbol

STOP

SUBTTL subtitle

SUPPRESS symbol,...,symbol

SYN sym1,sym2

TAPE

.TEXT dtextd

TITLE title

TWOSEG addr

UNIVERSAL tabnam

VAR

XALL

.XCREF symbol,...,symbol

XLIST

XPUNGE

XWD word1,word2

Enters string text in 6-bit format; d is any delimiter not in text
Same as RADIUS
Ends IRP or IRPC before all subarguments or characters are used
Defines subtitle (80 chars max) to be printed at top of each listing page
Turns on suppress bit for symbol in symbol table; turned off when symbol referenced
Defines sym2 as synonymous with sym1
Causes assembler to begin assembling next source file in MACRO command string
Generates ASCIZ REL block for LINK and inserts text directly into .REL file; d is any delimiter not in text
Names program title and causes title to be printed on each page of listing
Directs MACRO to assemble two-segment program with HISEG beginning at addr
Declares symbol table of current program as available to other programs and stores tabnam in MACRO's internal UNIVERSAL table
Causes variable symbols defined with symbol#, ARRAY, or INTEGER to be assembled as BLOCK statements
Resumes standard listing after LALL or SALL
Suspends output of cross-referencing for symbol
Suspends output to program listing file for Pass 2 until next LIST
Deletes all local symbols during Pass 2
Enters low-order 18 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 (B18-35)
READER'S COMMENTS

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

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

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

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

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

________________________________________________________________________________________

Please indicate the type of 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 ________________________
SOFTWARE PUBLICATIONS
200 FOREST STREET   MRO1-2/L12
MARLBOROUGH, MA    01752