

Rank Xerox

Universal Time-Sharing System (UTS)

Sigma 6/7/9 Computers

A7

Command Processors
Technical Manual



Universal Time-Sharing System (UTS)

Sigma 6/7/9 Computers

Command Processors

Technical Manual

FIRST EDITION

90 19 93A

February 1973

Price: \$6.75

NOTICE

This publication documents command processors of the Universal Time-Sharing System (UTS) for Sigma 6/7/9 Computers. All material in this manual reflects the C01 version of UTS.

RELATED PUBLICATIONS

| <u>Title</u> | <u>Publication No.</u> |
|--|------------------------|
| UTS Overview and Index Technical Manual [†] | 90 19 84 |
| UTS Basic Control and Basic I/O Technical Manual | 90 19 85 |
| UTS System and Memory Management Technical Manual | 90 19 86 |
| UTS Symbiont and Job Management Technical Manual | 90 19 87 |
| UTS Operator Communication and Monitor Services Technical Manual | 90 19 88 |
| UTS File Management Technical Manual [†] | 90 19 89 |
| UTS Reliability and Maintainability Technical Manual | 90 19 90 |
| UTS Interrupt Driven Tasks Technical Manual [†] | 90 19 91 |
| UTS Initialization and Recovery Technical Manual | 90 19 92 |
| UTS System Processors Technical Manual | 90 19 94 |
| UTS Data Bases Technical Manual | 90 19 95 |

[†]Not published as of the publication date given on the title page of this manual. Refer to the PAL Manual for current availability.

The specifications of the software system described in this publication are subject to change without notice. The availability or performance of some features may depend on a specific configuration of equipment such as additional tape units or larger memory. Customers should consult their Xerox sales representative for details.

CONTENTS

| | |
|--|----|
| CCIR – CCI Executive Routine | 1 |
| Purpose | 1 |
| Overview | 1 |
| Usage | 4 |
| Input | 4 |
| Output | 4 |
| Interaction | 5 |
| I/O Subroutines | 6 |
| Errors | 8 |
| Description | 8 |
| Flowchart | 10 |
| CCITBLS – CCI Tables and Other Static Data | 17 |
| SUBR – General Subroutines | 17 |
| HEXCK – Convert EBCDIC to Hexadecimal | 18 |
| HEXBCD – Convert Hexadecimal to EBCDIC | 19 |
| GETDECVAL – Get Value Following Keyword | 20 |
| DECCNV – Convert Decimal to Binary | 21 |
| WDTBSRH – Search Word Table | 22 |
| GETDCBA – Get DCB Assignment | 23 |
| TOPPAGE – Output Top-of-Form | 24 |
| EOCCSCAN – Scan to End of Control Command | 25 |
| GETACCN – Get Account Number | 26 |
| GETSN – Get Tape Reel Serial Number | 27 |
| LISTDATE – List Date and Time | 28 |
| GETHEXVAL – Get Hexadecimal Field | 29 |
| BCDHEX – Convert EBCDIC to Hexadecimal | 30 |
| GETAVAL – Get Address Value | 31 |
| GETPASSW – Get Password | 32 |
| DCBCLS – Close Output DCBs | 33 |
| BINDECBCD – Convert Binary to Decimal EBCDIC | 34 |
| GETLOC – Get Relative or Absolute Hexadecimal Location | 35 |
| CHSTSHFT – Shift Character String | 37 |
| CALENDTE – Test for Legal Expiration Date | 38 |
| LIST – Subroutines for Listing CCI Control Commands and Error Messages | 40 |
| SERRLF – Set Error Listing Flags in the JIT | 41 |
| SCCLF – Set Control Command Listing Flags in the JIT | 42 |
| SCSELF – Set Command Error Pointer and Error Flag | 43 |
| ERRLFCK(CCLFCK) – Check for Output Device Duplication | 45 |
| CCLIST – Control Command List Routine | 47 |
| ERRLIST – List Error Message | 48 |
| LIST – Output a Message | 51 |
| CHARROUT – Subroutines for Performing Syntax Analysis on Character Strings | 52 |
| NXACTCHR – Get Next Active Character | 54 |
| NAMSCAN – Get Name from Command | 56 |
| CHARSCAN – Compare Current Character | 57 |
| HEXSCAN – Scan Hexadecimal Number | 58 |
| QUOTSCAN – Compare Two Character Strings | 59 |
| DECSCAN – Scan Decimal Number | 60 |
| CHTSCAN – Get Next Field from Input | 61 |
| GETCHST – Get Next Field if Buffer is Empty | 62 |
| OPNSTARF – Set Up Open FPT and Open File | 63 |
| OPNSTARF – Open Star File (*SYSIDX) | 63 |
| OPNF – Open File Routine | 65 |
| JOBR – Process JOB Control Command | 67 |
| LIMR – Process LIMIT, MESSAGE, TITLE, POOL, and SWITCH Control Commands | 70 |
| LIMR – Process LIMIT Control Command | 70 |
| MESSR – Process MESSAGE Control Command | 72 |
| TITLR – Process TITLE Control Command | 74 |
| POOLR – Process POOL Control Command | 75 |

| | |
|---|-----|
| SWITCHR – Process SWITCH Control Command | 77 |
| ASSGR – Process ASSIGN Control Command | 79 |
| LOADR – Process LOAD and OVERLAY Control Commands | 83 |
| LOCCT – Load Control Command Table | 87 |
| TREER – Process TREE and PTREE Control Commands | 90 |
| TREER – Process TREE Control Command | 90 |
| PTREER – Process PTREE Control Command | 93 |
| TELSCOPE – Pack the LOCCT, Tree Table, and ROMT into One Table | 95 |
| RUNR – Process RUN Control Command | 97 |
| RUN Table Format | 99 |
| DEBUGR – Process INCL Control Command and the Debug Control Commands | 101 |
| INCL – Process the INCL Control Command | 102 |
| MODFYR – Process the MODIFY Control Command | 104 |
| PMDR – Process the PMD, PMDI, and PMDC Control Commands | 106 |
| SNAPR – Process the SNAP and SNAPC Control Commands | 108 |
| IFR – Process the IF, AND, OR, and COUNT Control Commands | 110 |
| WRTDEBUG – Write Out Debug Table | 113 |
| READBI – Read BI Device and Write to System BI File | 114 |
| LOGRT – Perform Job Initialization Functions in Response to a JOB Control Command and Validate Resource Specifications | 116 |
| Flowchart of the Resource Checking Process | 119 |
| TEL – Terminal Executive Language | 120 |
| Overview | 120 |
| Purpose | 121 |
| Usage | 124 |
| Input | 124 |
| Output | 124 |
| Data Bases | 124 |
| Errors | 126 |
| Interactions | 128 |
| Descriptions | 128 |
| TEL Commands | 130 |
| BACKUP – Save a File on the System Backup Tape | 131 |
| BUILD – Create a Keyed File on Secondary Storage | 132 |
| BYE – Terminate a Terminal Session | 132 |
| CANCEL – Delete a Job Waiting in the Input Symbiont | 132 |
| COMMENT – Direct the Output of the M:DO DCB | 133 |
| CONTINUE – Continue an Interrupted Process | 133 |
| COPY – Copy a File or Device Input to the Specified File or Device | 133 |
| DELETE – Delete the Specified Files | 134 |
| DISPLAY – Display System Load Parameters | 134 |
| DONT COMMENT | 134 |
| DONT LIST | 134 |
| DONT OUTPUT | 134 |
| EDIT – Call the Edit Processor | 135 |
| END – Abort the Currently Active Process | 135 |
| FORT4 – Interface with FORTRAN IV Processor | 135 |
| GET – Restore User's Environment | 136 |
| GO – Continue an Interrupted Process | 137 |
| JOB – Check Status of Batch Jobs | 138 |
| L – List File Names and, Optionally, Attributes from the Account Directory, Tape, or Disk Pack | 138 |
| LIST – Direct the Output of the M:LO DCB | 138 |
| LMN – Initiate Execution of a Load Module | 140 |
| MESSAGE – Send a Message to the Operator Console | 140 |
| META – Interface with Meta-Symbol Processor | 140 |
| OFF – Terminate a Terminal Session | 141 |
| OUTPUT – Direct the Output of the M:GO DCB | 141 |
| PASSWORD – Create or Cancel a Password for a User's Account | 142 |
| PLATEN – Define the Terminal Output Page Size | 143 |
| PRINT – Send Output to be Placed in the Symbiont Queue | 143 |
| PROCEED – Continue an Interrupted Process | 143 |
| QUIT – Terminate the Current Job Stop | 144 |

| | |
|---|-----|
| RESET - Reset All DCB Assignments to their System Default Value | 144 |
| RESTORE - Restore the Previously Saved Core Image | 145 |
| RUN - Link One or More ROMs and Initiate Program Execution | 145 |
| SAVE - Check-point and Save a User's Environment for Later Use | 146 |
| SET - Create or Extend A/M Table Entries | 148 |
| START - Initiate the Execution of a Load Module | 149 |
| STATUS - List User Status | 150 |
| STOP - Terminate the Current Job Step | 151 |
| TABS - Enter Simulated Tab Stops | 152 |
| TERMINAL - Specify Terminal Type or Get Terminal Status | 152 |
| U - Imply 'UNDER DELTA' for the Next Command | 152 |
| TEL Subroutines | 153 |
| ABNRET - Abnormal Return Upon Reading the Terminal | 153 |
| ASSIGN - Assign-Merge Table Manipulator | 154 |
| BITS - Reset File Extension Bits | 155 |
| CHKDCBN - Validate DCB Name | 156 |
| DECBIN - Convert EBCDIC to Binary | 157 |
| EXPAND - Expand a Compacted Assign-Merge Table Entry | 158 |
| FID - Break a Complex FID | 159 |
| FILENT - Create Short Form PLIST | 160 |
| FLOP - Open Input File | 161 |
| NAME\$ - Create Unique Name for \$ Files | 162 |
| NFND - Convert to TEXTC Format | 162 |
| PENT - Insert a Parameter into Skeletal PLIST | 163 |
| RBLOG - Convert and Store in JIT a Remove Batch Work Station Name | 164 |
| READAM - Read Assign-Merge Table | 165 |
| REBIT - Reset Option Bit Upon DCB Assignment Release | 165 |
| SCAN - Parse Command Line | 166 |
| SYNTAX - Error Handler | 167 |
| SYSERR - Handle System Detected Error | 168 |
| TRUNDLE - Compact PLIST | 168 |
| WRITAM - Write Assign-Merge Table | 169 |
| LOGON, OFF - Initiate On-line Session; Terminate Session | 170 |
| Purpose | 170 |
| Usage | 170 |
| Input | 170 |
| Output | 171 |
| Interaction | 171 |
| Data Bases | 172 |
| Subroutines | 172 |
| Errors | 173 |
| Restrictions | 174 |
| Description | 174 |
| ACCNTSUM - Accounting Subroutine (Synonymous with ACCTSUM) | 176 |
| Installation Accounting Routines | 181 |

ID

CCIR - CCI Executive Routine

PURPOSE

CCI is the system processor that reads the batch control commands and either performs the specified function or calls the appropriate processor to perform it. CCIR is the main driving routine for CCI.

OVERVIEW

The user communicates with UTS in batch mode through control commands that enter the system via the control (C) device, normally a card reader. The control commands specify system functions such as compiling, loading, or executing a program. CCI reads the control commands and either performs the specified function or calls the appropriate processor to perform it. CCI also performs job initiation functions as well as outputs error messages when it detects an error in the processing of control commands.

CCI interfaces with the Monitor via CALs, tables, and files, and with other processors via tables and files using the assistance of the Monitor.

CCI uses the CALs for performing its I/O functions, for getting dynamic data pages for working storage, and for returning control to the Monitor. The information from the various control commands is passed to the other parts of the system either directly through a table, or indirectly through the table written to disk where it can be retrieved later by some other system processor or by the Monitor itself.

CCI also interfaces with the Monitor through tables that are embedded in the Monitor. Since CCI operates with a write key of 0, the Monitor is not write-protected from CCI. This allows CCI to change tables in the Monitor directly without going through a CAL. For example, CCI makes changes to the JIT (Job Information Table) and OPLBT2 (Operational Label 2) when it processes the LIMIT and STDLB control commands. Hardware locations X'4E' and X'4F' are used to locate the different tables in the Monitor. Location X'4E' always contains the address of a fixed format vector table (CCITAB), each word of which contains the address of a Monitor table. Location X'4F' always contains the address of the JIT.

Although CCI operates as a processor program, the Monitor treats CCI as a special processor by giving it control in the master mode with a write key of 0. Since the RUNFLAG in the JIT is zero when CCI is operating, the Monitor can test this flag to determine if CCI is in control or is the issuer of a CAL. The Monitor treats the M:EXIT call from CCI as an interpretive EXIT just as it does for TEL.

CCI is loaded and given control by the Monitor when the operator does the start key-in !Syyndd, I. When CCI gets control it reads the control command device until it reads either a JOB or FIN command, ignoring any other commands encountered.

CCI is also loaded and given control when any batch program (user or processor) gives up control or is terminated by the Monitor. If the previous program's termination was normal, CCI reads the control command device until a control command is found, ignoring any data records. If the previous program's termination was abnormal, CCI aborts the current job by reading the control device and ignoring all of the data records and control commands until it encounters a JOB or FIN command.

CCI relinquishes control to the Monitor when a processor or user program is to be loaded or when it encounters a FIN command and the system is to be put in the wait state.

Symbols used in the module descriptions are as follows:

| | | |
|-----|----------------|----|
| R0 | register | 0 |
| R1 | | 1 |
| R2 | | 2 |
| R3 | | 3 |
| R4 | | 4 |
| R5 | | 5 |
| R6 | | 6 |
| R7 | | 7 |
| SR1 | | 8 |
| SR2 | | 9 |
| SR3 | | 10 |
| SR4 | | 11 |
| D1 | | 12 |
| D2 | | 13 |
| D3 | | 14 |
| D4 | | 15 |
| CC1 | condition code | 1 |
| CC2 | | 2 |
| CC3 | | 3 |
| CC4 | | 4 |

CRPT - Character Routines Parameter Table (see below)

LOCCT - Load Control Command Table (see Section PA.09)

ROMT - Relocatable Object Module Table (see Section PA.09)

Format of Character Routines Parameter Table

This table is used by the character string manipulating routines (e.g., HEXSCAN, DECSCAN) for processing character strings (e.g., control commands). The table is set up in CCI's push down stack, TSTACK, when CCI is entered. Register 7 is set to contain the address of the table. The address must be in register 7 when the character routines are called.

| | | | |
|------|--|---------|-----------|
| Word | | | |
| 0 | ND | BADLM | |
| 1 | CONTCLMN | CONTR | |
| 2 | | | OUTR |
| 3 | | | CCP |
| 4 | F ₁ F ₂ F ₃ | 0 ——— 0 | CBUF |
| 5 | | | CSL |
| 6 | | | PCCP |
| 7 | | | RSZ |
| 8 | PLB | | |
| 15 | ~ ~ ~ ~ ~ | | ~ ~ ~ ~ ~ |

Buffer for
Character String

where

ND is the number of delimiters in the delimiter list.

BADLM is the byte address of delimiter list.

CONTCLMN is the character position to begin processing in a continuation record.

CONTR is the address of routine to read continuation records.

OUTR is zero or address of routine to output image in the input buffer.

CCP is the character position of next character in the input record.

F₁ 0 => blank is not active. 1 => blank is active.

F₂ 0 => do not blank out input record. 1 => set characters in input record to blanks as they are obtained.

F₃ 0 => buffer in parameter list is empty. 1 => buffer in parameter list is full.

CBUF is the address of buffer containing the input record.

CSL is the number of characters in character string in buffer in the parameter list.

PCCP is the character position of first character of last field obtained.

RSZ is the size of input record in bytes.

USAGE

CCI is entered at CCINIT in module CCIR from Monitor routine LDPRGM via an LPSD instruction.

Output registers:

- R6, R7, SR1 - contain the name of the processor, in TEXTC format, if a processor command was encountered. R6 is set to 0 if wait state is to be entered, and 2 if RUN command was processed.
- D2, D3 - contain the ':SYS' account.

INPUT

- ACCBUF - displacement in JIT for control command buffer.
- CCPL - location of control command parameter list.
- CJOB - contains address of JIT.
- CSL - displacement in Character Routines Parameter Table for character string length.
- J:BUP - location containing program lower limit.
- KEYHEAD - location of key for load module head record.
- LISTCNT - number of output list DCBs.
- LMNAME - displacement in the Load Control Command Table (LOCCT) for the load module name.
- M:C - address of M:C DCB.
- M:LO - address of M:LO DCB.
- M:X1 - address of M:X1 DCB.
- ONOFF - parameter list for CAL to type 'OFF' on OC.
- PLB - displacement in Character Routines Parameter Table for buffer containing first field of command.
- RLMNENTRY - displacement in LOCCT for load module name.
- SYSID - displacement in JIT for system ID of current job.
- TTE SIZE - Size of Tree Table.

OUTPUT

- ACCN - displacement in JIT for account number. Incremented by 1 to insure non-zero account.
- ALOCCT - displacement in JIT for address of LOCCT. Set to size of table packed by TELESCPE.
- CBUF - displacement in Character Routines Parameter Table for buffer address. Set to address of buffer in JIT.
- CCBEF - displacement in JIT for certain bit flags. Bit 29 is set if control command is assumed to be a processor command. Bit 28 is set or reset to indicate if control command buffer is full.

UTS TECHNICAL MANUAL

- CCLFLAGS - displacement in JIT for list flags. Bit 23 is set to cause skipping to end of command and is reset afterwards.
- CCLTFLGS - displacement in JIT for flags (bits 14-15) which are set to indicate if a LOAD or RUN command was processed.
- CCP - displacement in Character Routines Parameter Table for current character position. Initialized to 1.
- CCRSZ - displacement in character Routines Parameter Table for byte count.
- J:CCBUF - location of control command buffer.
- J:RNST - flags in first half of this word are reset on initial entry at CCINIT. Bit 1 is set before error exit is made via M:XXX.
- J:TELFLGS - set to 1 on initial entry at CCINIT.
- LDSYSID - displacement in LOCCT which is set to system ID of current job.
- MJCFLG - displacement in JIT for message flags. Bit 0 is set if 'Missing Job Command' message has been printed.
- MRT - displacement in JIT. Set to 0 when FIN command encountered.
- ROMTADR - displacement in LOCCT for ROM table address. Changed to a displacement after tables are packed by TELESCOPE.
- TREETADR - displacement in LOCCT for Tree Table address. Changed to displacement after tables are packed by TELESCOPE. Set to 0 or 1 if RUN or LOAD command precedes debugging command.
- TSTACK - Temp stack for saving registers, program lower limit and control command parameter list.

INTERACTION

Modules used:

- ASSGR - Process ASSIGN control command.
- CHARROUT - Get next character string (CHSTSCAN).
- LIMR - Process LIMIT control command (LIMR), MESSAGE control command (MESSR), TITLE control command (TITLR), POOL control command (POOLR), or SWITCH control command (SWITCHR).
- LIST - Set error listing flags in JIT (SERRLF), set control command listing flags in JIT (SCCLF), set control command error listing flag in JIT (SCCELF), test if C device is equal to OC device (CCLFCK), list control command (CCLIST), list error message (ERRLIST), or list specified output on specified device (LIST).
- OPNSTARF - Open star file, *SYSIDD or *SYSIDT (OPNSTARF), or open file for load module specified on RUN command (OPNF).
- READBI - Move BI information to disc file *SYSIDB.
- RUNR - Process RUN control command.
- SUBR - Search specified word table (WDTBLSRH), position device to top of page (TOPPAGE), scan to end of control command (EOCCSCAN),

- TELESCOPE - close CCI's output DCB's (DCBCLS), or shift character string (CHSTSHFT). Pack the LOCCT, Tree Table and ROM Table into one table (TELESCPE).
- TREER - Process TREE control command (TREER) or PTREE control command (PTREER).
- DEBUGR - Process INCLUDE control command (INCLR), process MODIFY control command (MODYFR), process PMD, PMDC, or PMDI control command (PMDR), process SNAP or SNAPC control command (SNAPR), process IF, AND, OR or COUNT control command (IFR), or search Tree Table for segment name (SEGSRCH).
- JOBR - Process JOB control command.

Monitor services used (handwritten CALs and FPTs):

- M:XXX - error exit to Monitor.
- M:CLOSE - close and save debug file, load module file, or SYSIDT file (M:X1), or close and release old debug file.
- M:WRITE - write load module name as key in SYSIDT file.
- M:GCP - get pages required for debug control command processing.
- M:FCP - free common pages used for debug control command processing.
- M:READ - read load module head and tree table records, or read C device.
- M:GP - get pages needed for reading in tree table.
- M:OPEN - open M:C DCB.
- M:EXIT - normal exit to Monitor (enter wait state).
- M:TYPE - type 'OFF' on .OC at end of job.

SUBROUTINES

CCREAD

Purpose: To read the C device until a control command is read and then get the first field of the command.

Usage:

Calling sequence

BAL, SR4 CCREAD

Input registers

R5 - contains address of JIT

R7 - contains address of CRPT (Character Routines Parameter Table)

Output registers

R1 - contains first four characters of command field from the control command read.

SR1 - contains current character.

Description: The next record is read from the C device by calling the CREAD routine. If the record read is not a control command but the previous record was a control command, the message DATA INFORMATION - IGNORED is output and another read is done. If the record read is a control command, the record size and buffer address are stored in the CRPT, the first field of the command is obtained, and R1 is loaded with the first four characters of the command.

CREAD

Purpose: To read one record from the C device.

Usage:

Calling sequence

BAL, SR4 CREAD

Input register

R5 - contains address of JIT.

Output register

SR3- contains 0 if data record was read. Contains 6 in byte 0 if control command was read.

Description: If the flag CCBEF is set, which indicates that the control command buffer is full, the flag is reset and the C device is not read. In this case, the character count is stored in the Character Routines Parameter Table and an exit is made.

If the flag CCBEF is not set on entry, the C device is read via the M:READ procedure.

When CCI reads the C device, the Monitor reads the record into the control command buffer in the JIT. However, a buffer address in CCI is used in the FPT for reading the C device. This is to prevent the Monitor from giving an error return for an illegal buffer address since the Monitor checks the buffer address for all reads to be sure it is not in a protected memory area.

Any tab characters found in the buffer are replaced with blanks.

MOVEIT

Purpose: To move the load table or run table to common dynamic storage.

Usage:

Calling sequence

BAL, SR4 MOVEIT

Input register

R2 - contains location of table to be moved.

R3 - contains size of table in words.

Output register

R2 - contains number of pages freed plus one.

Description: An M:GCP procedure is used for the number of pages required for the word size in register 3. The size of the table (contents of register 3 plus one) is stored in the first word of the move area. The table is then moved to dynamic storage. The pages acquired are released via an M:FCP procedure.

ERRORS

ILLEGAL CONTROL COMMAND OR CONTROL COMMAND SEQ.

Self-explanatory

MISSING JOB COMMAND, JOB IGNORED.

Self-explanatory

Other errors which are encountered in command processing are reported by routine ERRLIST in module LIST.

Return to the Monitor is made via the M:XXX procedure after any error conditions.

DESCRIPTION

Module CCIR is entered from the Monitor at location CCINIT. CCINIT is the control routine that calls all other major routines in CCI, reads the control commands, and interprets the command field. If the command is a processor command, it calls the Monitor via an M:EXIT call with the processor name in TEXTC format in registers 6, 7, and 8. If it is not a processor command, it calls the appropriate CCI routine(s) to process the command. If it reads a JOB or FIN command, it calls the job termination routines.

When CCI reads a control command, it gets the command field and checks the first four characters of the field to determine if it is a legal CCI control command. If it is not, CCI assumes it to be a processor control command, blanks out the ! character in the control command buffer, and marks the buffer as full and containing a processor command by setting the PCCF flag in the JIT. It then gives control to the Monitor to load the processor.

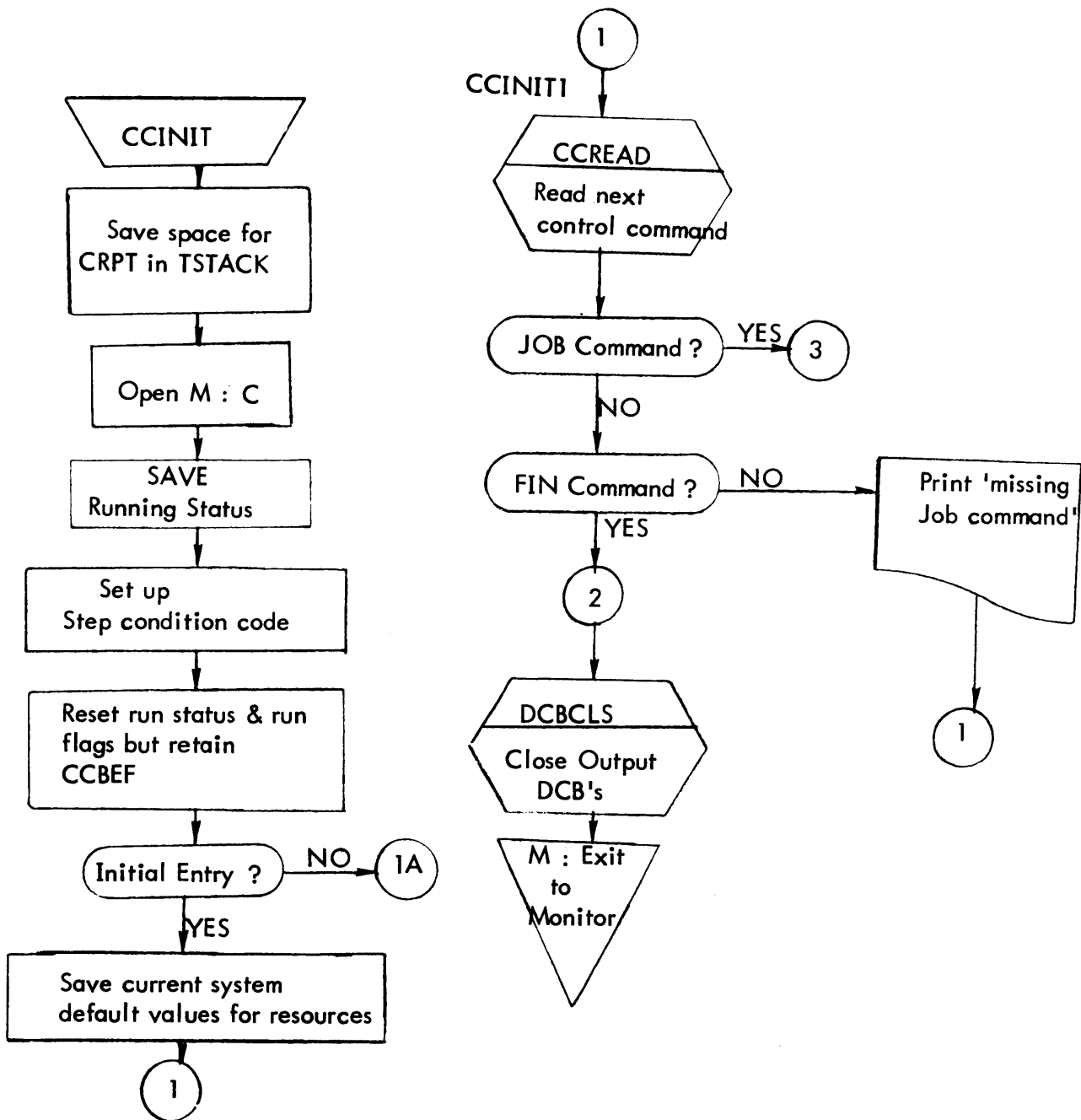
If the command is a legal CCI control command, CCI calls the appropriate subroutine

in CCI to process the command. The routine that processes a particular control command knows the correct format and syntax for the command and which keywords and values are legal for it. Processing of most control commands results in a table being built with the information from the command in the table. The table is then written to disk for later use by some other part of the system. For the LIMIT command, the values are stored directly into the JIT.

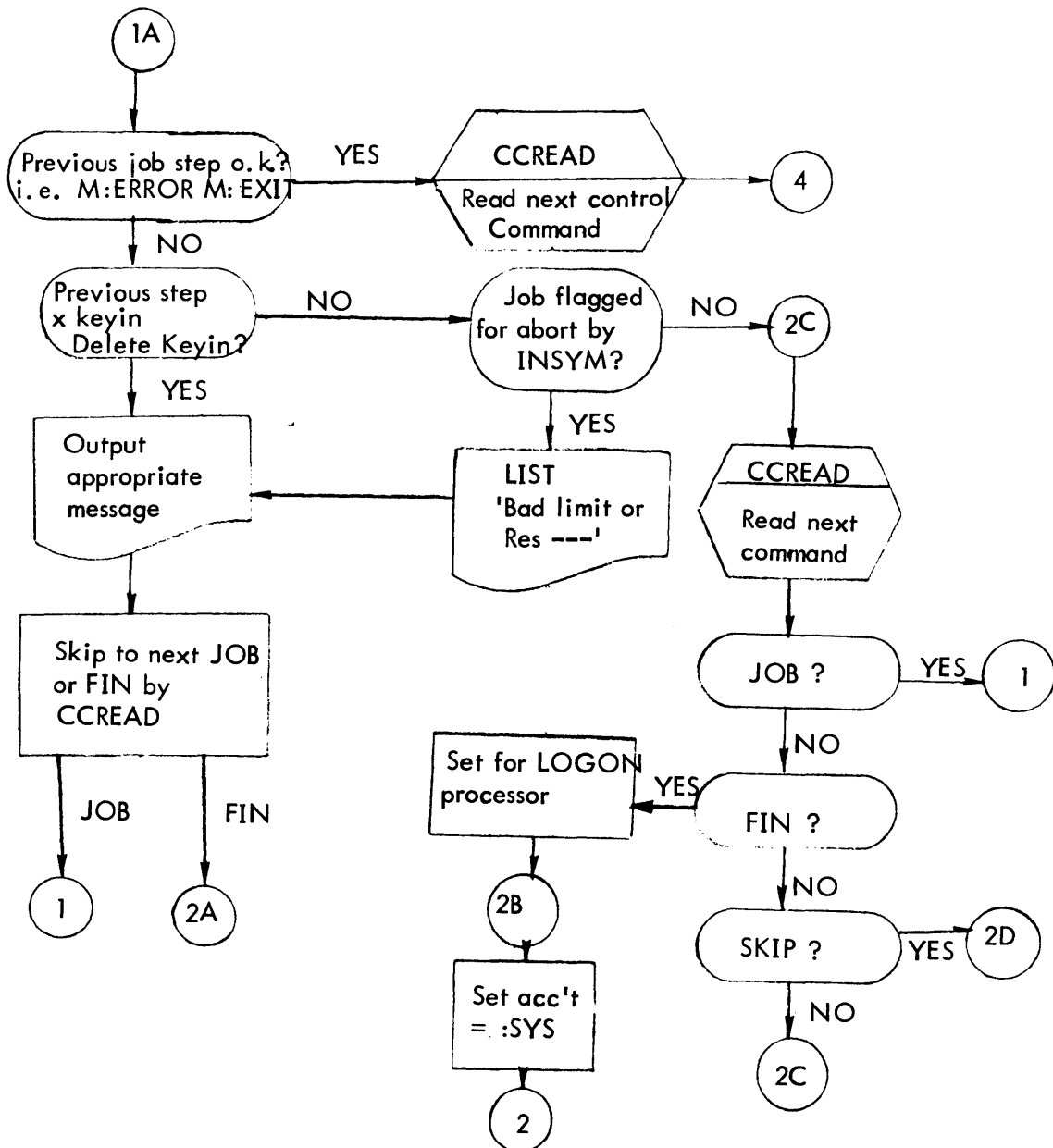
When a LOAD or OVERLAY command is processed, a TREE or PTREE command and any MODIFY commands following the LOAD or OVERLAY command are processed before the Overlay Loader is loaded. If a TREE or PTREE command does not follow a LOAD or OVERLAY command, a default Tree Table with one entry is built, with the segment name equal to the load module name.

When a RUN command is processed, all MODIFY, PMD, PMDI, PMDC, IF, AND, OR, COUNT, SNAP and SNAPC commands following the RUN command are processed until a DATA or non-debug command is read. Then an M:EXIT is made with the address of the Run Table in register 6.

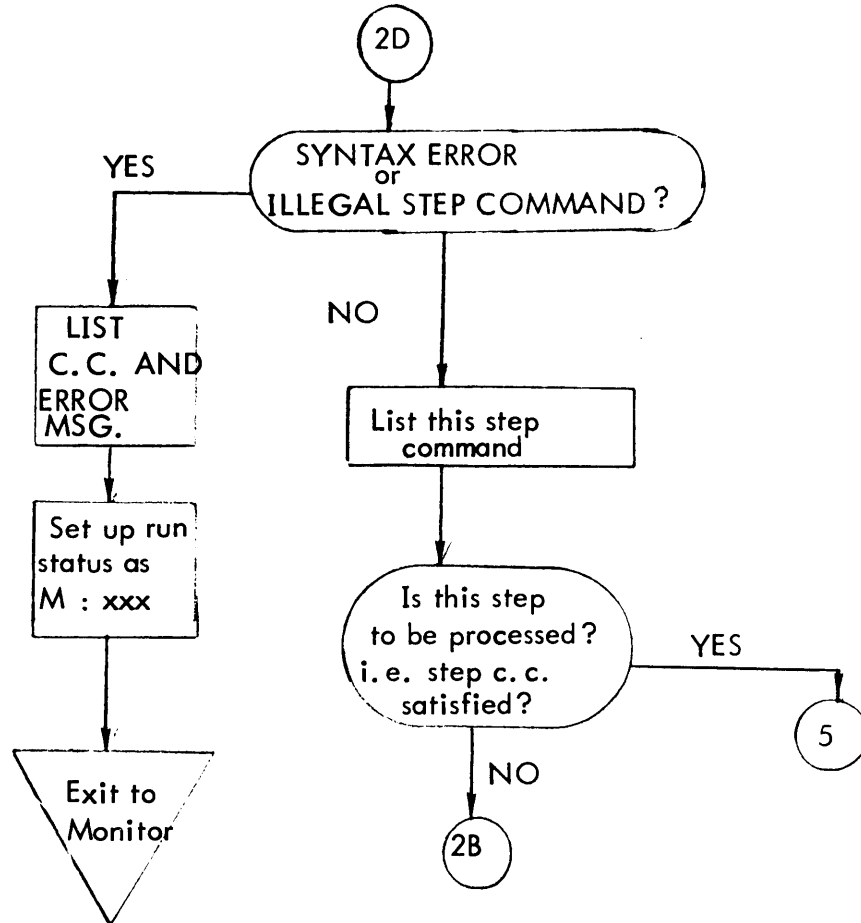
UTS TECHNICAL MANUAL

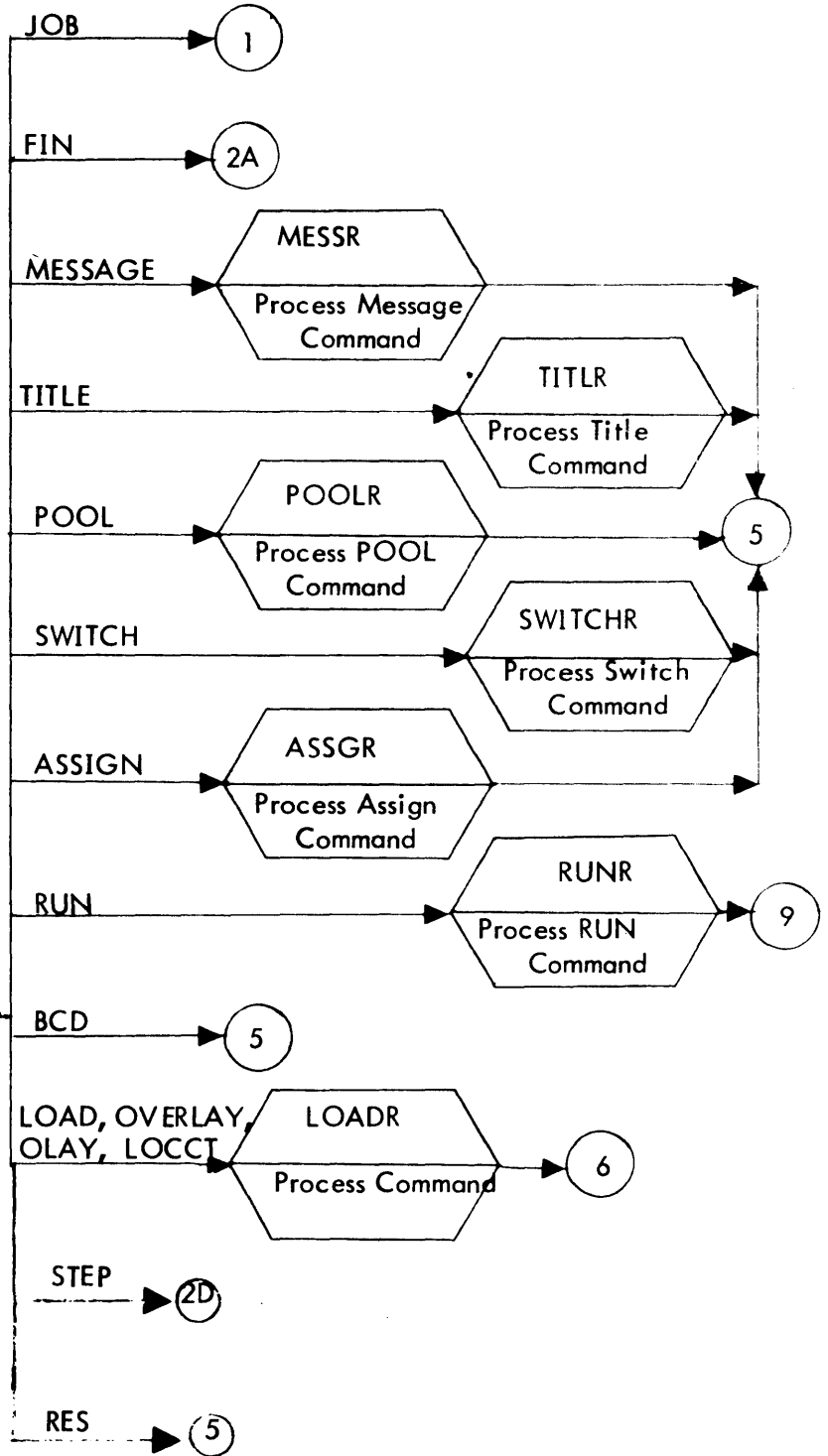
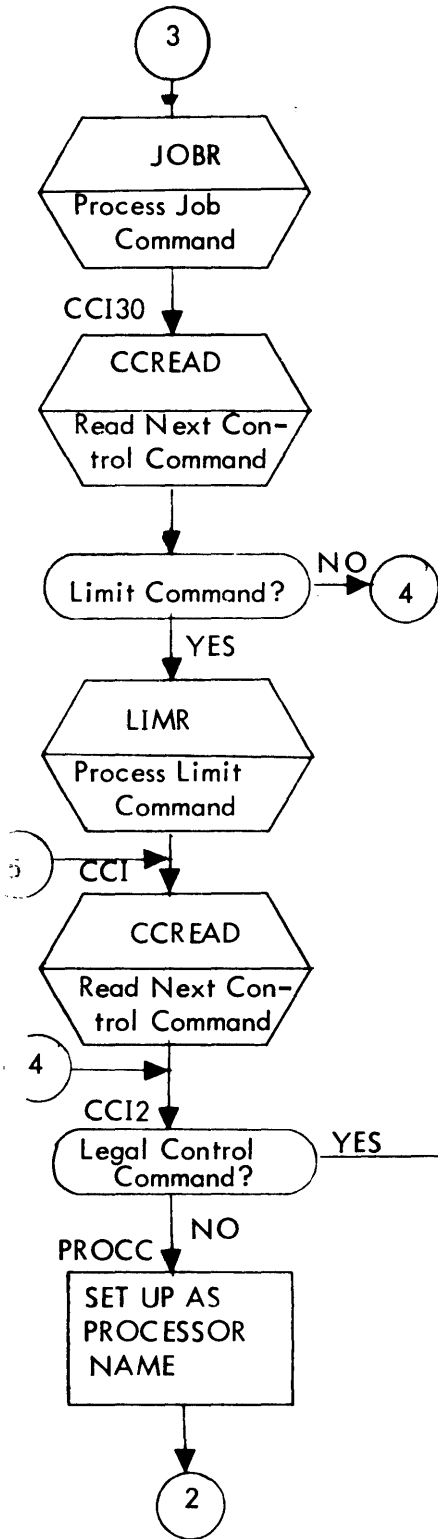


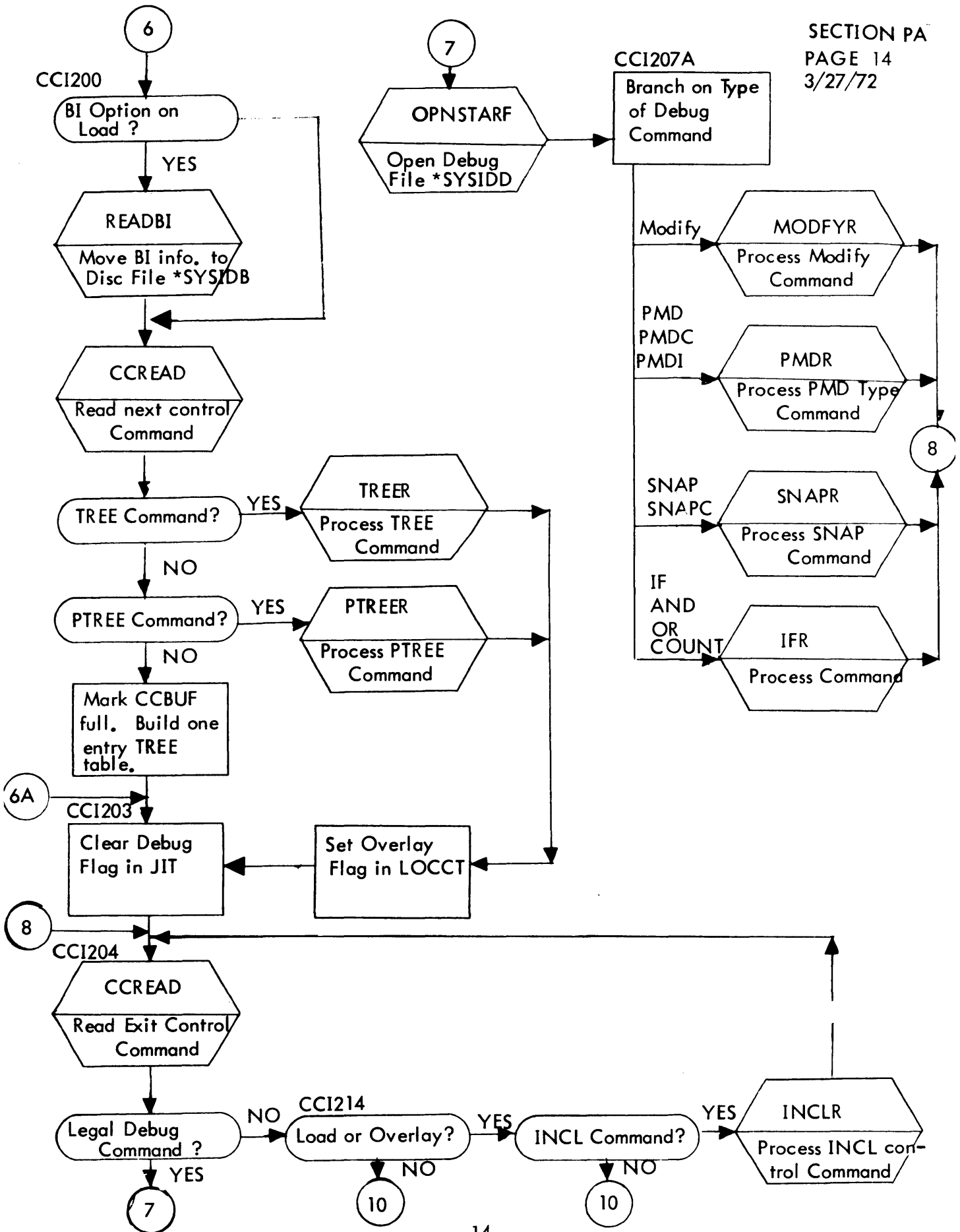
UTS TECHNICAL MANUAL



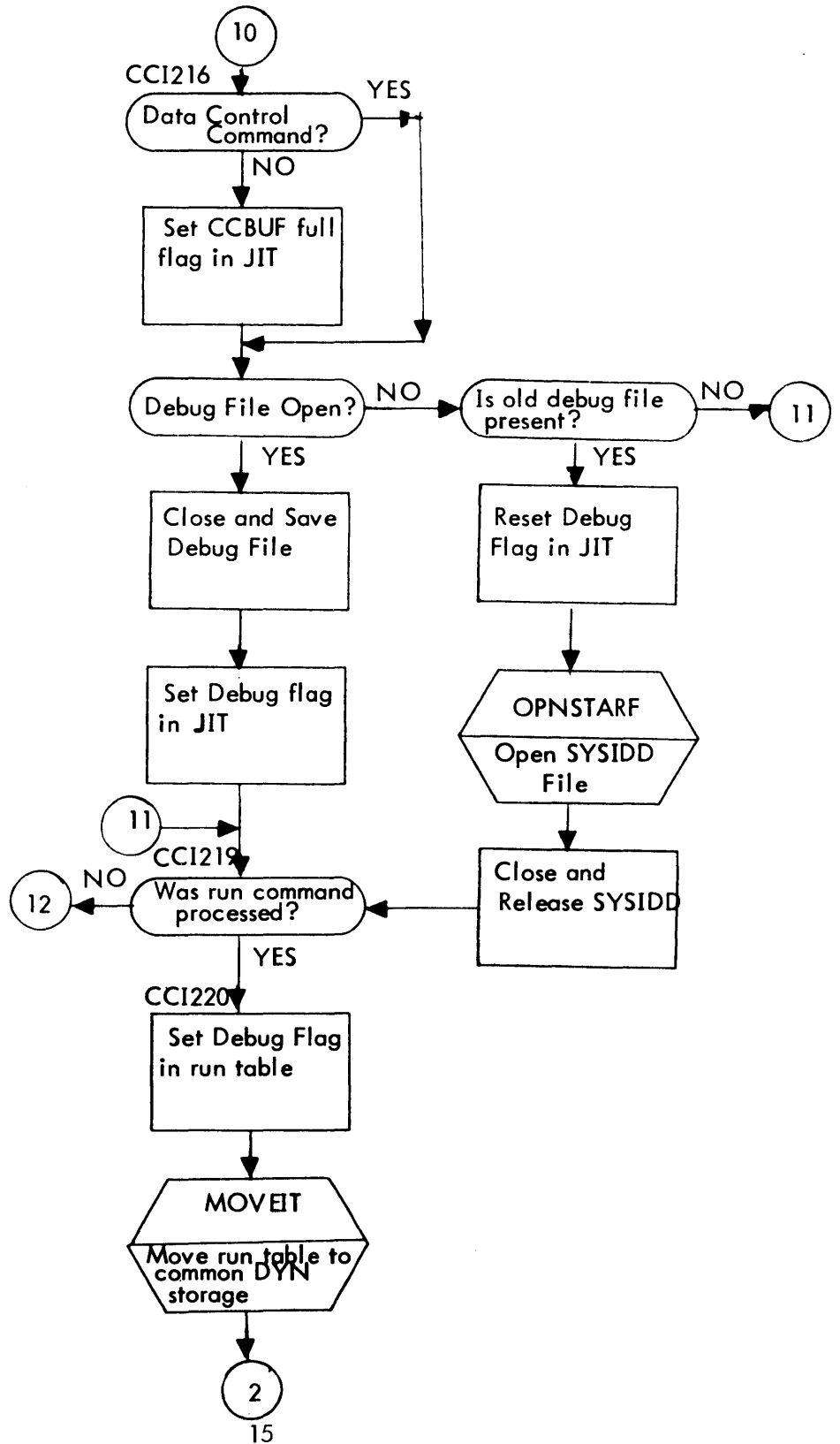
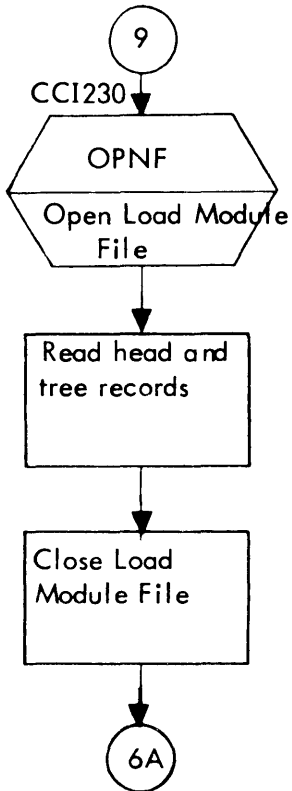
UTS TECHNICAL MANUAL



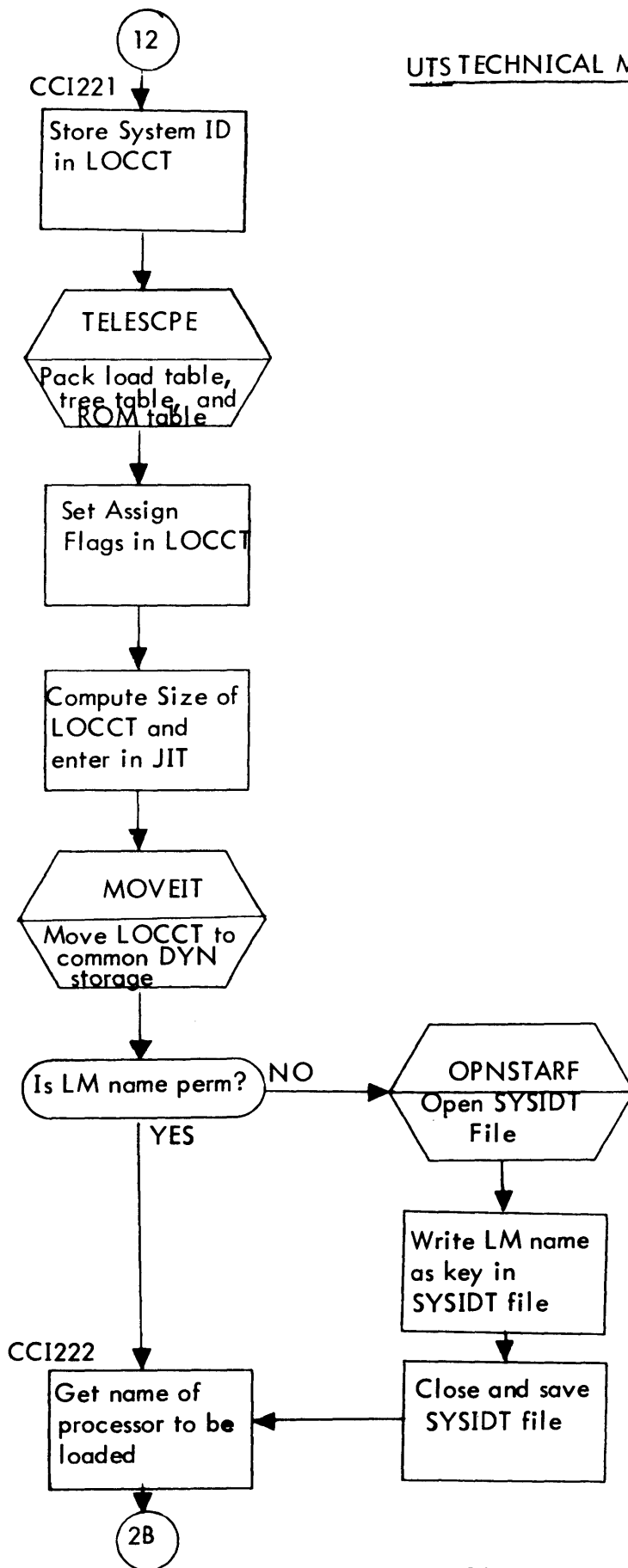




UTS TECHNICAL MANUAL



UTS TECHNICAL MANUAL



ID

CCITBLS - CCI Tables and other Static Data

PURPOSE

To provide tables, parameter lists, and definitions for reference by other CCI modules.

DESCRIPTION

This module contains the following tables and data:

1. Displacements of parameters in a DCB.
2. The push-down stack (TSTACK).
3. Various constants (K, X, or Y symbols).
4. FPT codes for various procedures.
5. Address table of output DCB's (LISTDCBT).
6. Keys for reading desired records.
7. Displacements of parameters in LOCCT.
8. Default copy of the control command parameter list.
9. Error codes for various errors detected in processing the control commands.

ID

HEXCK - Convert EBCDIC to hexadecimal

PURPOSE

To convert a valid EBCDIC character to hexadecimal.

USAGE

Calling sequence

BAL, D4 HEXCK

Input register

R2 - EBCDIC character in low-order byte.

Output register

R2 - bits 28-31 contain hexadecimal value.

DESCRIPTION

The EBCDIC character is checked for being a legal hexadecimal character (X'F0' through X'F9' or X'C1' through X'C6') and if so, converted from X'F0' through X'F9' to X'0' through X'9', and X'C1' through X'C6' to X'A' through X'F'. CC1 is set to 0 for a normal return and is set to 1 for an error return (invalid EBCDIC character).

ID

HEXBCD - Convert hexadecimal to EBCDIC

PURPOSE

To convert an 8-digit hexadecimal value to an 8-character EBCDIC value.

USAGE

Calling sequence

BAL, D4 HEXBCD

Input register

D1 - 8-digit hexadecimal number

Output registers

D1, D2 - 8-character EBCDIC converted value

DESCRIPTION

The input 8-digit hexadecimal number is converted to EBCDIC and returned in registers D1 and D2.

UTS TECHNICAL MANUAL

ID

GETDECVAL - Get value following keyword

PURPOSE

To get the value following a keyword and convert it to binary.

USAGE

Calling sequence

BAL, SR4 GETDECVAL

Input registers

- R1 - maximum allowed value
- R7 - address of CRPT
- SR1 - current character or zero

Output registers

- R2 - binary value if normal return (CC1 = 0)
- SR3 - error code if error return (CC1 = 1)

INPUT

- CSL - displacement in CRPT for character string length.
- PLB - displacement in CRPT for character string to be converted.

INTERACTION

Modules used:

- CHARROUT - check for comma following keyword (CHARSCAN) and scan decimal number (DECSCAN).
- SUBR - convert from EBCDIC to binary (DECCNV).

ERRORS

The appropriate error code is returned in SR3 and CC1 is set to 1.

DESCRIPTION

If the next field is a legal decimal value and less than the specified value, it is returned in R2. If the comma is not found or the next field is not a legal decimal value less than the specified value, an error return is given with the appropriate error code in SR3.

ID

DECCNV - Convert decimal to binary

PURPOSE

To convert an EBCDIC decimal number to binary.

USAGE

Calling sequence

BAL, SR4 DECCNV

Input registers

R0 - number of characters to convert

R1 - word address of the first character

Output registers

R2 - binary value if normal return (CC1 = 0)

ERRORS

CC1 is set to 1 if conversion error.

DESCRIPTION

The EBCDIC decimal number is converted and returned in R2. If one of the EBCDIC characters is not a legal decimal digit, or if the converted number is greater than $2^{31} - 1$, an error return is given.

UTS TECHNICAL MANUAL

ID

WDTBLSRH - Search word table

PURPOSE

To search a word table for the specified input word.

USAGE

Calling sequence

BAL, SR4 WDTBLSRH

Input registers

R1 - word to be used in search

R2 - address of table to be searched

R3 - size of table to be searched, including dummy 0th entry.

R4 - return address if search fails.

Output register

R3 - index into table where item was found.

DESCRIPTION

Using the contents of R1, the specified table is searched for a match. If a match is found, the relative position of the match in the table searched is returned in R3, and exit is made to the address in SR4. Otherwise, the return is to the address specified in R4.

ID

GETDCBA - Get DCB assignment

PURPOSE

To get the device or CFU assignment for the DCB.

USAGE

Calling sequence

BAL, D4 GETDCBA

Input register

R6 - DCB address

Output register

R1 - DEV value or CFU address

INPUT

DSI - displacement of DEV value in DCB.

INTERACTION

Monitor service used (handwritten CAL and FPT):

M:OPEN - open DCB if not already open

DESCRIPTION

If the DCB is closed, a CAL is issued to open it. If the ASN field of the DCB is 1, indicating file assignment, the CFU address is returned in register 1. Otherwise the DEV/OPLB field of the DCB is returned in register 1.

UTS TECHNICAL MANUAL

ID

TOPPAGE - Output top-of-form

PURPOSE

To output a top of form on the specified devices.

USAGE

Calling sequence

BAL, SR4 TOPPAGE

Input registers

R4 - logical device indicators in bits 25-31

bit 25 = AL
26 = PO
27 = DO
28 = LO
29 = SL
30 = LL
31 = OC

INPUT

LISTDCBT - table of output DCB addresses

INTERACTION

Module used:

LIST - check for device duplication (ERRLFCK)

Monitor service used (handwritten CAL and FPT):

M:DEVICE(PAGE) - output top of form

DESCRIPTION

A top of form request is made for the logical devices whose corresponding bits are set in R4. If any of the logical devices are assigned to the same physical device, only one top of form is output for that physical device.

ID

EOCCSCAN - Scan to the end of a control command

PURPOSE

To scan to the end of a control command, skipping over all continuation records.

USAGE

Calling sequence

BAL, SR4 EOCCSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

INTERACTION

Module used:

CHARROUT - get next active character (NXACTCHR)

DESCRIPTION

The current control command is scanned until a X'26' (EOB), X'15' (NL) or X'FF' (next continuation record not obtainable) character is found. All continuation records are read and output on the specified devices.

UTS TECHNICAL MANUALID

GETACCN - Get account number

PURPOSE

To get the next field from the current control command and check if it is a legal account number.

USAGE

Calling sequence

BAL, SR4 GETACCN

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - error code if error return (CC1 = 1).

INPUT

CSL - displacement in CRPT for character string length

OUTPUT

FLAGS - displacement in CRPT for flag word. Bit 2 is set to indicate buffer is full.

INTERACTION

Module used:

CHARROUT - scan for decimal number (DECSCAN), and scan for alphanumeric name (NAMSCAN)

ERRORS

The appropriate error code is returned in SR3 if the account number is not legal.

DESCRIPTION

The next field is moved to the buffer in the CRPT and checked for being a legal account number. If it is not a legal account number, an error return is given with the error code in SR3. A legal account number is a string of one to eight decimal or alphanumeric characters.

ID

GETSN - Get tape reel serial number

PURPOSE

To get the next field from the current control command and check if it is a legal tape serial number.

USAGE

Calling sequence

BAL, SR4 GETSN

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - error code if error return (CC1 = 1)

INPUT

CSL - displacement in CRPT for character string length

OUTPUT

FLAGS - displacement in CRPT for flag word. Bit 2 is set to indicate buffer is full.

INTERACTION

Module used:

CHARROUT - scan for alphanumeric name (NAMSCAN) and scan for decimal number (DECSCAN).

ERRORS

The appropriate error code is returned in SR3 if the serial number is not legal.

DESCRIPTION

The next field is moved to the buffer in the CRPT and is checked for being a legal serial number. If it is not legal, an error return is given with the error code in SR3. A legal serial number is a string of one to four decimal or alphanumeric characters.

UTS TECHNICAL MANUAL

ID

LISTDATE - list date and time on specified devices

PURPOSE

To obtain and list the date, time, and job id on the listing log.

USAGE

Calling sequence

BAL, SR4 LISTDATE

Input register

R5 - address of JIT

INPUT

SYSID - displacement in JIT for system ID of current job

OUTPUT

TSTACK - temp stack which is used for buffer space for line to be output

INTERACTION

Modules used:

LIST - check for device duplication (ERRLFCK), and output line to LL and OC (LIST).

SUBR - convert hexadecimal ID to BCD (HEXBCD).

DESCRIPTION

The current time and date are obtained via the M:TIME procedure and the job id is obtained from the JIT (Job Information Table). The items are put together as one image and output on the LL device. The word 'ON' is typed on the OC device.

UTS TECHNICAL MANUAL

ID

GETHEXVAL - Get hexadecimal field and convert to hexadecimal

PURPOSE

To get a hexadecimal field from a control command and convert it from EBCDIC to hexadecimal.

USAGE

Calling sequence

BAL, SR4 GETHEXVAL

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - error code if error return (CC1 = 1)

D3 - hexadecimal value if normal return (CC1 = 0)

INPUT

CSL - displacement in CRPT for character string length

PLB - displacement in CRPT for character string buffer

INTERACTION

Modules used:

CHARROUT - scan hexadecimal field (HEXSCAN)

SUBR - convert to hexadecimal (BCDHEX)

ERRORS

The appropriate error code is returned in SR3 if the hexadecimal field is not valid.

DESCRIPTION

The next field is moved to the buffer in the CRPT. Each character is checked for being a legal hexadecimal character and, if legal, is converted to its corresponding hexadecimal digit. The converted value is returned in D3. If the value is greater than eight digits, or if one of the characters is not a hexadecimal character, an error return is given with the error code in SR3.

UTS TECHNICAL MANUALID

BCDHEX - convert a hexadecimal EBCDIC string to hexadecimal

PURPOSE

To convert a string of one to eight hexadecimal EBCDIC characters to hexadecimal digits.

USAGE

Calling sequence

BAL, SR4 BCDHEX

Input registers

D1, D2 - one to eight hexadecimal EBCDIC characters, left-justified and blank filled

Output register

D3 - converted hexadecimal value

INTERACTION

Module used:

SUBR - convert EBCDIC character to hexadecimal (HEXCK)

ERRORS

Error exit is made if character is not hexadecimal. CCI is set to 1.

DESCRIPTION

If all of the EBCDIC characters are hexadecimal, the converted result is returned in D3 right-justified. If not, an error return is given with the error code in SR3.

UTS TECHNICAL MANUAL

ID

GETAVAL - get address value from command

PURPOSE

To get the next field from the current control command, check if it is a legal hexadecimal address, and convert it.

USAGE

Calling sequence

BAL, SR4 GETAVAL

or

BAL, SR4 GETAVAL10

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output registers

R2 - converted hexadecimal address value (CC1 = 0 for normal return)

SR3 - error code if error return (CC1 = 1)

INTERACTION

Modules used:

CHARROUT - test if next character is a comma (CHARSCAN).

SUBR - convert hexadecimal EBCDIC field to hexadecimal (GETHEXVAL)

ERRORS

The appropriate error code is returned in SR3 and CC1 set to 1 if expected comma is missing or if hexadecimal field is invalid.

DESCRIPTION

If the routine is entered at GETAVAL, the next active character in the control command is checked for being a comma. If the routine is entered at GETAVAL10, the check for the comma is by-passed. The next field is obtained from the control command, checked for being a legal hexadecimal value, and then converted to hexadecimal. The result must be less than 2¹⁷.

If the check for the comma fails, or the field is not a legal hexadecimal address, the error exit is taken.

ID

GETPASSW - get password from command

PURPOSE

To get the next field from the control command and check if it is a legal file password.

USAGE

Calling sequence

BAL, SR4 GETPASSW

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - error code if error return (CC1=1)

OUTPUT

FLAGS - displacement in CRPT for flag word. Bit 1 is set and reset for blank-out mode

INTERACTION

Module used:

CHARROUT - scan character string (CHSTSCAN)

ERRORS

The appropriate error code is returned in SR3 and CC1 set to 1 if subroutine CHSTSCAN makes an error return.

DESCRIPTION

The next field is moved into the buffer in the CRPT and checked if it is a legal character string. The password is left-justified and blank filled in the buffer. This routine does not check for valid length of the password.

UTS TECHNICAL MANUAL

ID

DCBCLS - Close output DCBs

PURPOSE

To close output DCBs M:LL, M:SL, M:LO, M:DO, M:PO, and M:AL

USAGE

Calling sequence

BAL, SR4 DCBCLS

INPUT

LISTDCBT - table of output DCB addresses

INTERACTION

Monitor service used (handwritten CAL and FPT):

M:CLOSE - close DCB

DESCRIPTION

Any of the DCBs named above that are open are closed via the M:CLOSE call.

UTS TECHNICAL MANUAL

ID

BINDECBCD - Convert binary to decimal EBCDIC

PURPOSE

To convert a binary value to decimal EBCDIC characters.

USAGE

Calling sequence

BAL, SR4 BINDECBCD

Input register

D1 - binary number

Output registers

D1, D2 - 8 character decimal conversion

DESCRIPTION

The binary number is converted to its EBCDIC decimal equivalent. The input value must be less than 10^8 or the result will be incorrect.

UTS TECHNICAL MANUAL

ID

GETLOC - Get relative or absolute hexadecimal location

PURPOSE

To get the next field on the current control command and check if it represents a legal relative or absolute hexadecimal location of the form:

1. name
2. res (name)
3. name \pm hex value
4. res (name) \pm hex value
5. + hex value

USAGE

Calling sequence

BAL, SR4 entry

Error return

Normal return

where entry = GETLOC, GETLOC1, GETLOC2 or GETLOC3

Input registers

R3 - address of buffer to contain the relative or absolute hexadecimal location

R7 - address of CRPT

SR1 - current character or zero

Output registers

R4 - flags indicating type of resolution

SR3 - contains error code if error return is made

INPUT

PLB - displacement in CRPT for character string buffer.

CSL - displacement in CRPT for character string length.

OUTPUT

FLAGS - displacement in CRPT for flag word. Bit 2 is set to indicate buffer is full.

TSTACK - address of temp stack. SR4 in stack is incremented by one for normal return.

UTS TECHNICAL MANUAL

INTERACTION

Modules used:

- CHARROUT - test for presence of + or - sign or right parenthesis (CHARSCAN) or scan for legal alphanumeric name (NAMSCAN).
- SUBR - change name to a TEXTC format (CHSTSHFT) or get address value from command (GETAVAL10).

ERRORS

Various error conditions cause error return to (SR4) with error code in SR3. Normal return is to (SR4)+ 1.

DESCRIPTION

The different entry points are for obtaining the various forms:

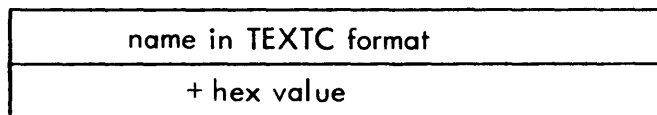
- GETLOC - name
- GETLOC1 - res (name)
- GETLOC2 - name ± value or + value
- GETLOC3 - res (name) or + value

where

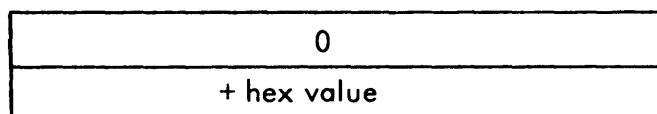
res is BA for byte address resolution - (R4) = 0; HA for halfword address resolution - (R4) = 1; WA for word address resolution - (R4) = 2; DA for doubleword address resolution - (R4) = 4

name is the external definition in a program

Fields of the above type are found on the various control commands that have fields for specifying a location in a program such as on the MODIFY control command. The information is stored in the caller's buffer in the following format:



Or if no name is present:



UTS TECHNICAL MANUAL

ID

CHSTSHFT - shift character string

PURPOSE

To shift a character string a specified number of bytes to the right and store the character length in byte zero.

USAGE

Calling sequence

BAL, SR4 CHSTSHFT

Input registers

R0 - word address of buffer containing the character string

R1 - number of bytes in character string

R2 - number of bytes to shift the character string

DESCRIPTION

Operation - The character string is shifted the specified number of characters to the right and the byte count is stored in byte zero. No check is made to determine if the buffer is large enough to shift the characters the specified number of bytes.

UTS TECHNICAL MANUAL

ID

CALENDTE - Test for legal expiration date

PURPOSE

To check if the next active character for the current control command is a comma, and check if the next field is a legal expiration date.

USAGE

Calling sequence

BAL, SR4 CALENDTE

Error return

Normal return

Input registers

R0 - address of location to store the expiration date

R7 - address of CRPT.

SR1 - current character or zero

Output register

SR3 - contains error code if error return

INPUT

PLB - displacement in CRPT for buffer containing current field

CSL - displacement in CRPT for character string length

INTERACTION

Module used:

CHARROUT - test for presence of a comma (CHARSCAN), compare quote constant with character string (QUOTSCAN), or scan for decimal number (DECSCAN).

ERRORS

Various error conditions cause error return to (SR4) with error code in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

The next active character is checked for a comma. Then the next field is obtained and checked if it is a legal date. If it is legal, it is moved to the buffer specified

UTS TECHNICAL MANUAL

in R0 with the following format:

if mm, dd, yy specified

| | | | |
|---|---|---|---|
| m | m | d | d |
| 0 | 0 | y | y |

or

if ddd specified

| | | | |
|---|---|---|---|
| ♣ | d | d | d |
| 0 | 0 | ♣ | ♣ |

or if date is 'NEVER'

| | | | |
|---|---|---|---|
| N | E | V | E |
| R | ♣ | ♣ | ♣ |

where

- mm is month, 01 through 12
 - dd is day, 01 through 31
 - yy is year, 00 through 99
 - ♣ is x '40'
 - 0 is x 'FO'
- all in Hexadecimal

If the next active character is not a comma, or a legal expiration date is not found, an error return is made with an error code in SR3.

ID

LIST - Routines for Listing CCI Control Commands and Error Messages

DESCRIPTION

This module contains the routines for listing the CCI control commands and error messages, and for checking if two logical devices are equivalent to the same physical device. It also contains the control command error messages.

CCI outputs control commands and error messages on seven logical devices (AL = Accounting Log, PO = Punched Output, DO = Diagnostic Output, LO = Listing Output, SL = System Log, LL = Listing Log, OC = Operator's Console). All control commands and error messages are output on the LL device. When error messages or control commands are output, a check must be made to determine if two or more logical devices are the same physical device so that the control command or error message is not output more than once on a particular physical device. This module contains the routines which perform this function.

UTS TECHNICAL MANUAL

ID

SERRLF - Set error listing flags in the JIT.

PURPOSE

To set the error listing flags in the JIT. These flags indicate the devices CCI shall use for outputting error messages.

USAGE

Calling sequence

BAL, SR4 SERRLF

Input registers

R4 - error flags for logical devices where

- Bit 25 = AL device
- 26 = PO device
- 27 = DO device
- 28 = LO device
- 29 = SL device
- 30 = LL device
- 31 = OC device

R5 - address of JIT

INPUT

ERLFMASK - mask for store select to enter flags in the JIT

OUTPUT

ERLFLAGS - displacement in the JIT of error flag word. Set to error listing flags.

INTERACTION

Module used:

LIST - check for output device duplication (ERRLFCK).

DESCRIPTION

The physical device assignments associated with the logical devices specified in R4 are checked for duplication. If any duplication exists, the associated bits are reset in R4 and bits 25-31 are stored into bits 25-31 of word 35 in the JIT.

UTS - TECHNICAL MANUALID

SCCLF - Set control command listing flags in the JIT

PURPOSE

To set the control command listing flags in the JIT. These flags indicate the devices on which CCI shall output the control commands.

USAGE

Calling sequence

BAL, SR4 SCCLF

Input registers

R4 - control command listing flags for logical devices where

25 = AL device

26 = PO device

27 = DO device

28 = LO device

29 = SL device

30 = LL device

31 = OC device

R5 - address of JIT

INPUT

CCLFMASK - mask for store select to enter flags in the JIT.

OUTPUT

CCLTFLGS - displacement in the JIT for list flag word. Set to command listing flags.

INTERACTION

Module used:

LIST - check for output device duplication (CCLFCK).

DESCRIPTION

The physical device assignments associated with the logical devices specified in R4 are checked for duplication. If any duplication exists, the associated bits are reset in R4 and bits 25-31 are stored in bits 25-31 of word 34 in the JIT.

ID

SCCELF - Set command error pointer and error flag

PURPOSE

To set the \$ character in the appropriate position in the control command error buffer, and to set the output control flag in the JIT for the control command error buffer when a control command has a syntax error.

USAGE

Calling sequence

BAL, SR4 SCCELF

Input registers

R4 - 0 if current character is in error; 1 if current field is in error

R5 - address of JIT

R7 - address of CRPT

INPUT

CCP - displacement in the CRPT for the current control command pointer.

PCCP - displacement in the CRPT for the pointer to the beginning of the current field.

OUTPUT

CCEBFM1 - location of first word of error buffer which includes byte count in first byte.

CCEBF - location of second word of error buffer. '\$' is entered as error pointer.

CCLTFLGS - displacement in the JIT for list flag word. Flags are set to indicate if error buffer precedes or follows next record.

DESCRIPTION

The \$ character is stored in the control command error buffer so that it will be output below the current character in error, or below the first character of the field in error. Then the rest of the buffer is filled with blanks and the flag is set (bits 22, 23 of word 34 of the JIT) for outputting the control command buffer. If the current field is in error and was continued from one record to the next, the first part of the field in error will have already been output. Therefore, to put the \$ character under the first character of the field in error, the control command error buffer must be output before the next record of the control command. Hence, the flag setting has three

possible states:

- 0 = do not output buffer.
- 1 = output buffer before next record of the control command
- 2 = output buffer after next record of the control command

The buffer is output by the CCLIST routine.

UTS TECHNICAL MANUALID

ERRLFCK (CCLFCK) - Check for output device duplication.

PURPOSE

To check if specified logical devices and the physical device are the same.

USAGE

Calling sequence

BAL, SR4 ERRLFCK

or

BAL, SR4 CCLFCK

Input register

R4 - output flags for logical devices where

- Bit 25 = AL device
- 26 = PO device
- 27 = DO device
- 28 = LO device
- 29 = SL device
- 30 = LL device
- 31 = OC device

Output register

R4 - output flags for logical devices, with duplicate flags reset

INPUT

LISTCNT - number of entries in table LISTDCBT

LISTDCBT - table of output DCB addresses

OUTPUT

TSTACK - address of temp stack. Used as buffer for storing DCB assignments

INTERACTION

Module used:

SUBR - get DCB assignment (GETDCBA)

DESCRIPTION

For each bit set in R4 (25-31), the associated DCB's assignment is obtained. Then all of the assignments obtained are compared and for each duplication the associated bit in R4 is reset. The results are returned in R4. If the

UTS TECHNICAL MANUAL

routine is entered at CCLFCK, the assignment for the M:C DCB is obtained and checked if it is to a typewriter. If it is, then any output flag (or flags) associated with that physical device is reset so that the control command will not be output on the device through which it is being input.

ID

CCLIST - Control command list routine

PURPOSE

To output the control command and error buffer on the specified devices.

USAGE

Calling sequence

BAL, SR4 CCLIST

Input register

R5 - address of JIT

R7 - address of CRPT

INPUT

CBUF - displacement in the CRPT for the address of the command buffer.

OUTPUT

CCLTFLGS - displacement in the JIT for list flags. Error buffer list flag is reset.

INTERACTION

Module used:

LIST - output control command and error buffer (LIST).

DESCRIPTION

If the 'No CC list' flag is set in the JIT, an immediate exit is made. Otherwise, the error buffer flag is checked. If set, a further test is made to determine if the error buffer is to be output before or after the command. The control command and, if applicable, the error buffer are then output in the desired order.

UTS TECHNICAL MANUAL

ID

ERRLIST - List error message

PURPOSE

To output an error message on the specified devices.

USAGE

Calling sequence

BAL, SR4 ERRLIST

Input register

R5 - address of JIT

SR3- error code or address of message to be output

INPUT

ERRLFLGS - displacement in the JIT for error listing flags

INTERACTION

Module used:

LIST - list specified message (LIST)

ERRORS

One of the following messages is listed by this routine:

ABOVE CONTROL COMMAND IN ERROR
EXPECTED BLANK MISSING
EXPECTED COMMA MISSING
EXPECTED LEFT PARENTHESIS MISSING
EXPECTED RIGHT PARENTHESIS MISSING
EXPECTED TERMINATOR MISSING
EXPECTED HYPHEN MISSING
ILLEGAL CHARACTER
SYNTAX ERROR
ILLEGAL KEYWORD
ILLEGAL CHARACTER STRING
ILLEGAL ALPHANUMERIC NAME
ILLEGAL DECIMAL NUMBER
ILLEGAL HEXADECIMAL NUMBER
DUPLICATION OF FIELDS
ILLEGAL VALUE
ILLEGAL RELOCATABLE OBJECT MODULE NAME

UTS TECHNICAL MANUAL

DUPLICATE OR CONFLICTING OPTION
TOO MANY ASSIGNMENTS
TOO MANY INSNS OR OUTSNS
ERROR READING A/M
ERROR WRITING A/M
TOO MANY TABS
OP LABEL ERROR
INVALID NAME OR ACCOUNT
ILLEGAL ASSIGNMENT
MAXIMUM NUMBER OF TEMPORARY FILES EXCEEDED
INSUFFICIENT PARAMETERS
MAXIMUM NUMBER OF READ AND/WRITE ACCOUNTS EXCEEDED
ILLEGAL DCB NAME
SYSTEM DCB NOT-ASSIGNABLE
ILLEGAL BIAS VALUE, OUT OF RANGE
ILLEGAL SEGMENT NAME
TOO MANY EFS
ABN. COND. IN OPENING M:BI DCB
ABN. COND. IN OPENING DCB
ABN. COND. ON READ
ABN. COND. ON WRITE
ILLEGAL OR INSUFFICIENT INFORMATION ON RUN CC
I/O ERROR/ABNORMAL IN READBI SUBROUTINE
LIMIT GREATER THAN DEFAULT
ABORTED BY INSTALLATION, SEE SOMEBODY
MAXIMUM AUTHORIZED PRIORITY EXCEEDED
CAN'T GET DYNAMIC PAGES
SPEC. CORE EXCEEDS AUTHORIZED VALUE
SPEC. 9 TRACK TAPE EXCEEDS AUTHORIZED VALUE
SPEC. 7 TRACK TAPE EXCEEDS AUTHORIZED VALUE
SPEC. # OF TAPES EXCEEDS AUTHORIZED VALUE
SPEC. # OF SPINDLES EXCEEDS AUTHORIZED VALUE
USER NOT AUTHORIZED FOR REMOTE BATCH
TSTORE EXCEEDS REMAINING RAD SPACE ALLOCATED
TSTORE EXCEEDS REMAINING PACK SPACE ALLOCATED
PSTORE EXCEEDS REMAINING RAD SPACE ALLOCATED
PSTORE EXCEEDS REMAINING PACK SPACE ALLOCATED
FPOOL VALUE NOT WITHIN BOUNDS
IPOOL VALUE NOT WITHIN BOUNDS
JOB DELETED BY OPERATOR

UTS TECHNICAL MANUAL

DESCRIPTION

If (SR3) is less than or equal to X'300' it is interpreted as an error code and the address of the message is obtained from a table using (SR3) as an index after subtracting a base code:

$$\begin{aligned} \text{index} &= (\text{SR3}) \text{ if } (\text{SR3}) < \text{X}'100' \\ &= (\text{SR3}) - \text{X}'100' \text{ if } \text{X}'100' \leq (\text{SR3}) < \text{X}'1FF' \\ &= (\text{SR3}) - \text{X}'200' \text{ if } \text{X}'200' \leq (\text{SR3}) < \text{X}'300' \end{aligned}$$

If (SR3) is greater than X'300', it is interpreted as the address of the message to be output. The error message is output on the devices as specified in bits 25-31 of word 35 in the JIT.

UTS TECHNICAL MANUAL

ID

LIST - Output a message

PURPOSE

To output a specified message on specified devices.

USAGE

Calling sequence

BAL, SR4 LIST

Input registers

R2 - word address of message in TEXTC format

R4 - output flags for logical devices where

Bit 25 = AL device

26 = PO device

27 = DO device

28 = LO device

29 = SL device

30 = LL device

31 = OC device

D4 - byte displacement

INPUT

LISTDCBT - list of DCB addresses which has the addresses set for the device DCBs to be used.

INTERACTION

Monitor services used (handwritten CALs and FPTs):

M:WRITE - write message

M:DEVICE(DRC) - set direct bit if output is through M:PO.

DESCRIPTION

Operation - First, the output message is scanned for blanks and the byte count reduced by the number of trailing blanks. Then the message is output on the devices specified by the set bits 25-31 in R4. If the message is output through M:PO, the DRC bit is set in the DCB.

ID

CHARROUT - Perform Syntax Analysis on Character Strings

DESCRIPTION

This module contains a set of subroutines for doing syntax analysis on character strings. CCI uses these routines for doing the syntax analysis on the control commands.

The basic functions are to get the next active character and the next field from the control command. Getting the next active character consists of getting the next character if the blank character is active, or getting the next non-blank character if the blank character is not active. Getting the next field consists of getting the next group of active characters until a delimiter is encountered.

A routine is provided for testing the next active character to determine if it is a particular character such as a comma. Also routines are provided to determine the type of field, whether it consists of only decimal characters, or alphabetic characters, etc. The routines in this module are supported by the Character Routines Parameter Table (CRPT) and the Character Table (CHTBL). The CRPT specifies delimiter characters, the address of the buffer containing the character string to be analyzed, the length of the character string in the buffer, an eight word buffer to contain the next field, etc. The CHTBL, which is embedded in this module, classifies each of the 256 EBCDIC characters into one of four classes:

- 0 - special character (all non-printable characters, i. e. those not included in classes 1, 2, and 3, such as /, !, *, etc).
- 1 - numeric (0-9).
- 2 - letters A through F or digits X'A' through X'F'
- 3 - alphabetic only and special characters (G through Z, \$, @, #, :, -, |).

Any character may be a delimiter regardless of its classification. The semicolon is the continuation character. Whenever the next active character is a semicolon, the read routine specified in the CRPT is called.

For using the routines in this module, register seven (R7) must contain the address of the CRPT and register eight (SR1), the current character or zero. The current character is the last character obtained in processing the current string. When the next active character is obtained, it is loaded into SR1 and becomes the current character. Setting SR1 to zero marks the current character as used. If the routine that gets the next active character is called with SR1 containing a non-zero value, it will return the character in SR1 as the next active character. If it is called with SR1 = 0, it will get the next active character string in the buffer.

The eight-word buffer in the CRPT has a flag associated with it which, when set, means that the buffer in the CRPT contains the next field; that is, the different routines that operate on a field will use the one in the buffer in the CRPT. If the flag is reset, they will get the next field from the character string and operate on

UTS TECHNICAL MANUAL

that field. When a field is moved to the buffer in the CRPT, it is stored left-justified and blank filled. A field between two consecutive delimiters is a null field, has zero length, and is not valid. Also, a field greater than 31 characters is not a legal field.

ID

NXACTCHR - Get next active character

PURPOSE

To get the next active character from the character string being processed.

USAGE

Calling sequence

BAL, SR4 NXACTCHR

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output register

SR1 - next active character

INPUT

CCRSZ - displacement in CRPT for the command record size

CBUF - displacement in CRPT for the command buffer address

FLAGS - displacement in CRPT for the flag word

CLD - displacement in CRPT for a word containing the number of delimiters recognized and the byte address of the delimiter table

OUTR - displacement in CRPT for address of output list routine (CCLIST)

CONTR - displacement in CRPT for address of control command read routine (CREAD)

OUTPUT

CCP - byte displacement of current character in command buffer. Normally incremented by one. If semi-colon is encountered, bit 4 of this word is set to indicate continuation position.

INTERACTION

Modules used:

LIST - list control command (CCLIST)

CCIR - read C device (CREAD)

DESCRIPTION

If (SR1) = 0, the next character is obtained from the character string. If (SR1) is

not zero, the character in SR1 is used as the next character. Then the character in SR1 is checked whether it is a delimiter, or a continuation character (;), or a blank. If the character is a delimiter, it is returned with CC1 = 1. If the character is a continuation character, the output routine is called, if specified in the CRPT, and the read routine is then called to get the next record and then the next character. If the character is a blank and the blank active flag is not set, the blank is ignored and the next character is obtained and checked. Otherwise, the blank is returned as a delimiter in SR1 with CC1 = 1.

If the end of the character string is encountered in trying to get the next character, an EOB (X'26') is returned in SR1 with CC1 = 1. If the read routine gives an error return when a continuation record is to be read on X'FF' is returned in SR1 with CC1 = 1.

UTS TECHNICAL MANUAL

ID

NAMSCAN - get name from control command

PURPOSE

To get the next field from the control command and check if it is a legal alphanumeric name.

USAGE

Calling sequence

BAL, SR4 NAMSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

OUTPUT

FLAGS - displacement in CRPT of flag word. Buffer full flag is reset if legal name.

INTERACTION

Module used:

CHARROUT - get character string (GETCHST)

DESCRIPTION

If the buffer in the CRPT is empty, the next field is obtained from the specified character string and moved to the buffer where the field is checked for being a legal alphanumeric field. If it is, control is returned with CCI = 0; otherwise, with CCI = 1.

A legal alphanumeric field is any string of 1 to 31 alphabetic or numeric characters. At least one of the characters must be alphabetic. If the field is a legal field (1 to 31 characters, excluding delimiters), but not a legal alphanumeric name, the buffer in the CRPT is marked full. Otherwise, it is marked empty.

ID

CHARSCAN - Compare current character with character in SR2.

PURPOSE

To compare the next active character from the input character string with the specified scan character.

USAGE

Calling sequence

BAL, SR4 CHARSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

SR2 - comparison character

Output register

SR1 - set to current character if characters do not compare, otherwise set to 0.

INTERACTION

Module used:

CHARROUT - get next active character (NXACTCHR).

DESCRIPTION

The next active character is obtained from the input character string and compared with the scan character in SR2. If the two characters are equal, SR1 is set to 0 and CC1 is set to 0. If they do not compare, SR1 contains the next active character and CC1 is set to 1.

UTS TECHNICAL MANUAL

ID

HEXSCAN - scan hexadecimal number

PURPOSE

To get the next field from the specified input character string and check if it is a legal hexadecimal character field.

USAGE

Calling sequence

BAL, SR4 HEXSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

OUTPUT

FLAGS - displacement in CRPT for the flag word. The buffer full flag (bit 2) is set for empty if field is legal.

INTERACTION

Module used:

CHARROUT - get character string (GETCHST)

DESCRIPTION

If the buffer in the CRPT is empty, the next field is obtained from the specified character string and moved to the buffer. Then the field in the buffer is checked for being a legal hexadecimal field. If the field is a legal hexadecimal field, control is returned with the field in the CRPT buffer and CC1 = 0. Otherwise, CC1 = 1.

A legal hexadecimal field is any field of 1 to 31 hexadecimal characters 0 through F.

If the field obtained is a legal character field (1 to 31 characters, excluding delimiters), but not a legal hexadecimal field, the buffer in the CRPT is marked full. Otherwise, it is marked empty.

UTS TECHNICAL MANUALID

QUOTSCAN - compare two character strings

PURPOSE

To get the next field from the specified input character string and compare it with another string of characters.

USAGE

Calling sequence

BAL, SR4 QUOTSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

SR2 - address of the character string (in TEXTC format) which is to be compared.

OUTPUT

FLAGS - displacement in CRPT for the flag word. The buffer full flag (bit 2) is set for empty if fields are alike.

INTERACTION

Module used:

CHARROUT - get character string (GETCHST).

DESCRIPTION

If the buffer in the CRPT is empty, the next field is obtained from the specified character string and moved to the buffer. If the field obtained is legal (any string of 1 to 31 characters, excluding delimiters), it is compared to the character string specified by the address in SR2. First the lengths are compared, and if equal, the characters are compared one at a time until either they do not compare or until all of them do. If the strings are alike the buffer in the CRPT is marked empty. Otherwise, it is marked full.

ID

DECSCAN - scan decimal number

PURPOSE

To get the next field from the specified input character string and check if it is a legal decimal character field.

USAGE

Calling sequence

BAL, SR4 DECSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

OUTPUT

FLAGS - displacement in CRPT for the flag word. The buffer full flag (bit 2) is set for empty if field is legal.

INTERACTION

Module used:

CHARROUT - get character string (GETCHST).

DESCRIPTION

If the buffer in the CRPT is empty, the next field is obtained from the specified character string and moved to the buffer. Then the field in the buffer is checked for being a legal decimal field. If the field is a legal decimal field, control is returned with the field in the CRPT buffer and CC1 = 0. Otherwise, CC1 = 1.

A legal decimal field is any field of 1 to 31 decimal characters, 0 through 9.

If the field obtained is a legal character field (1 to 31 characters, excluding delimiters), but not a legal decimal field, the buffer in the CRPT is marked full. Otherwise, it is marked empty.

ID

CHSTSCAN - get next field from input

PURPOSE

To get the next field from the specified character string and check if it is a legal field.

USAGE

Calling sequence

BAL, SR4 CHSTSCAN

Input registers

R7 - address of CRPT

SR1 - current character or zero

INPUT

CCP - displacement in CRPT for current character string pointer.

OUTPUT

CSL - displacement in CRPT for character string length. Set to length of field.

FLAGS - displacement in CRPT for flag word. Blank active flag is set or reset.

PCCP - displacement in CRPT for previous character string pointer. Set to position of first non-blank character scanned.

INTERACTION

Module used:

CHARROUT - get next active character (NXACTCHR).

DESCRIPTION

First the buffer in the CRPT is filled with blanks; then characters are obtained one at a time from the input character string. If the character obtained is not a delimiter, it is stored in the buffer in the CRPT. When a delimiter is found or when 32 characters are collected without encountering a delimiter, no more characters are obtained. A legal field is any string of N consecutive characters, $0 < N \leq 31$, excluding delimiters.

CC1 is set to 1 if the field is not a legal field; otherwise, it is set to 0.

ID

GETCHST - Get next field if buffer is empty

PURPOSE

To get the next field if the buffer in the CRPT is empty, and check if it is a legal field.

USAGE

Calling sequence

BAL, SR4 GETCHST

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output registers

R1 - number of characters in field

R2 - byte index to buffer in the CRPT.

OUTPUT

FLAGS - displacement in CRPT for flag word. Buffer full flag (bit 2) is set.

INTERACTION

Module used:

CHARROUT - get next field (CHSTSCAN)

DESCRIPTION

If the buffer in the CRPT is full, control is returned with CC1 = 0 and R1 and R2 are set as shown above. If the buffer is empty, the next field is obtained and, if the field is legal, control is returned with CC1 = 0, R1 and R2 set as above, and the buffer in the CRPT marked full. If the field is not legal, control is returned with CC1=1.

UTS TECHNICAL MANUAL

ID

OPNSTARF - Open star file (*SYSIDX)

PURPOSE

To open the specified system file through the M:X1 DCB.

USAGE

Calling sequence

BAL, SR4 OPNSTARF

Input registers

R2 - function (IN, OUT, INOUT, or OUTIN)

R3 - organization (CONSEC or KEYED)

R4 - system file ID letter (A, B, D, G, L, or T)

R5 - address of JIT

D3 - error address or zero, to be put in the DCB if opened

D4 - abnormal address or zero, to be put in the DCB if opened

Output registers

R1 - address of M:X1 DCB

R2 - function

SR3 - 0 if file opened; I/O error or abnormal code if not opened

INPUT

M:X1 - address of M:X1 DCB

SYSID - displacement in JIT for current system ID

OUTPUT

TSTACK - temp stack. Used to set up file name address

INTERACTION

Module used:

OPNSTARF - open file (CPNF)

ERRORS

If the file is not opened, an error return is made with SR3 containing the I/O error or abnormal code.

DESCRIPTION

The system file name is constructed using the current job id from word 0 of the JIT

UTS TECHNICAL MANUAL

and the system file id letter in R4. The the OPNF routine is called which sets up the FPT and calls the Monitor open routine.

If the function specified is INOUT and the file does not exist, the file is opened in the OUT mode.

The system file name constructed is a 3-byte name, the first two bytes being the current job id and the third byte being file id letter (A, B, D, G, L, or T). If the current job id were X'0002', then the Assign file name would be X'030002C1'.

File id letters have the following meaning:

- A = file created from the ASSIGN commands
- B = file created for the Loader by CCI containing object modules input via the 'BI' device
- D = file containing debug information
- L = file created for the Loader containing the information from the LOAD and TREE commands
- G = GO file
- T = file containing the names of all temporary files to be released when the job terminates

UTS TECHNICAL MANUAL

ID

OPNF - Open file routine

PURPOSE

To open the specified DCB to the specified file

USAGE

Calling sequence

BAL, SR4 OPNF

Input registers

- R1 - address of DCB
- R2 - function (IN, INOUT, OUT, or OUTIN)
- R3 - organization (CONSEC or KEYED)
- R4 - address of file name
- R5 - address of JIT
- R7 - current temp stack pointer
- D1 - address of account number or zero
- D2 - address of password or zero
- D3 - error address or zero, to be put in the DCB if opened
- D4 - abnormal address or zero, to be put in the DCB if opened

Output registers

- R2 - function
- SR3 - 0 if file opened; I/O error or abnormal code if not opened

OUTPUT

TSTACK - temp stack. Used for setting up Open FPT.

INTERACTION

Monitor services used (handwritten CALs and FPTs):

- M:OPEN - open DCB
- M:SETDCB - enter caller-supplied error and abnormal addresses in DCB

ERRORS

If the file is not opened, an error return is made with SR3 containing the I/O error or abnormal code.

DESCRIPTION

The open FPT is set up and the Monitor open routine is called. If the function

specified is INOUT and the file does not exist, the file is opened in the OUT mode.

If the file is opened, an M:SETDCB call is made to store the caller-supplied error and abnormal addresses in the DCB.

UTS TECHNICAL MANUALID

JOBR - Process JOB control command

PURPOSE

To process the JOB control command and output time, date, ID, and account number.

USAGE

Calling sequence

BAL, SR4 JOBR

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - contains error code

INPUT

CSL - displacement in CRPT for length of field

PLB - displacement in CRPT of a buffer for containing the name and account

OUTPUT

ACCN - displacement in JIT for account number. Set to account number on JOB command.

J:TIME - set to time converted to minutes binary

UNAME - displacement in JIT for user's name. Set to name field if JOB command is valid; if invalid, first word is set to 0.

INTERACTION

Modules used:

CHARROUT - Get next character string (CHSTSCAN), compare current character (CHARSCAN), and scan hexadecimal number (HEXSCAN)

LIST - Set listing flags in the JIT (SCCLF) and print job identification (LIST).

SUBR - Convert hexadecimal to EBCDIC (HEXBCD), convert decimal

UTS TECHNICAL MANUAL

to binary (DECCNV), output top-of-form on LL device (TOPPAGE), scan to end of control command. (EOCCSCAN), get account number (GETACCN), and list date and time (LISTDATE).

- LOGRT - Open the logon file, :USERS.
- RBLOG - Convert work station ID to RBID and store it in JIT. (PB.C2)

Monitor services used (procedures):

- M:TIME - get time
- M:PRINT - space printer

Monitor services used (handwritten CALs and FPTs):

- M:OPEN - open M:PO DCB
- M:DEVICE(DRC) - set direct option for M:PO
- M:DEVICE(BIN) - set binary mode for M:PO
- M:GP - get page for data buffer
- M:WRITE - punch logotype card on the PO device
- M:FP - free page used for data buffer
- M:DEVICE(BCD) - set BCD mode for M:PO

SUBROUTINES

- HERALD This routine calls on subroutines PRNTSID and PRNTACNT to print the system ID and account number. Routine LISTDATE is called to print the date and time.
- PRNTSID This routine gets the system ID from the JIT and calls on BLKPRT to format and print the information.
- PRNTACNT This routine is called by HERALD and used routine BLKPRT to format and print the account number.
- BLKPRT This routine performs required conversion and formatting for system ID and job account and prints the information.

ERRORS

An error exit is made to (SR4) if a syntax error is encountered on the JOB command, or if an error exit is made from LOGRT. SR3 is set as described above. Normal exit is to (SR4) + 5.

DESCRIPTION

The current job id is updated in the JIT and the Assign flags are reset in the JIT. A logotype card is punched on the PO device and a top of form is output on the LL device. The time and date are output on the LL device.

UTS TECHNICAL MANUAL

The JOB command is processed and the parameters are stored in the JIT. If no priority is specified on the JOB command, the job priority in the JIT is set to 1. If a work station ID is present on the !JOB card, RBLOG (PB.02) is called to put the RBID in JIT. The JOB command is output on the AL, DO, PO, SL, LL, and OC devices. If an error is found in processing the JOB command, or the job account is not a valid one SR3 is set as shown above and the running status of this job is set up such that this job will be aborted and the job stream flushed.

After JOB control command syntax checking and account validation an attempt has been flagged for deletion-by-operator. If so, the job will be aborted and the job stream flushed.

In order to cope with the multi-partition operational mode of the system, a job is allowed to specify its resource requirements (such as CORE, spindle, tape drives) on the RES and/or LIMIT control command. The collection of information (along with others, such as priority and time) is pre-examined by the input symbiont and supplied to the multi-batch scheduler for job selection and execution.

In policing resource requests and allocation, three aspects have to be contended with:

- a) resource values specified by the job
- b) resource authorization values specified in the :USERS file
- c) system default resource values

It is possible for conflicts to arise between the three aspects. This conflict is resolved by CCI and will result either in the abortion of the job or release of resources.

The mechanics of resource checking are done in the routine LOGRT and description found in section PA.15.

UTS TECHNICAL MANUALID

LIMR - Process LIMIT control command

PURPOSE

To process the LIMIT control command and store values into the JIT

USAGE

Calling sequence

BAL, SR4 LIMR

Error return

Normal return

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return is taken, contains error code

INPUT

PLB - displacement in CRPT for buffer containing keyword

OUTPUT

MRT

MPPO

MPO

MDPO

MUPO

PRDCRM

PRDPRM

TMDCRM

TMDPRM

displacements in the JIT for limit values. Set to values specified for TIMF LO, PO, DO, UO, TSTORE, PSTORE, respectively.

INTERACTION

Modules used:

LIST - set listing flag in the JIT for the LL device (SCCLF).

SUBR - get value following Keyword (GETDECVAL), search word table (WDTBLSRH), and scan to end of control command (EOCCSCAN).

UTS TECHNICAL MANUAL

CHARROUT - compare current character (CHARSCAN) and get keyword from command (NAMSCAN)

ERRORS

Errors detected in syntax, keywords, or specified values cause an error exit to be taken with registers SR3 set as described above.

DESCRIPTION

The LIMIT control command is processed and output on the LL device. The values obtained are stored in the JIT. If an error is detected in processing the command, SR3 is set as shown above and the error exit is taken.

If PSTORE (and/or TSTORE) parameter is specified, the value is compared against user's remaining permanent (and/or temporary) secondary storage space. If the specified value is greater than the authorized value, the error exit is taken and SR3 is set up to contain the appropriate error code.

The LIMIT control command must be in the control command buffer on entry. Exit is made to (SR4) + 1 for normal return; otherwise exit is to (SR4).

UTS TECHNICAL MANUAL

ID

MESSR - Process MESSAGE control command

PURPOSE

To process the MESSAGE control command and output the message and the command on the OC device.

USAGE

Calling sequence

BAL, SR4 MESSR

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

INPUT

CBUF - displacement in CRPT for word containing address of control command buffer.

CCP - displacement in CRPT for word containing current character position in control command buffer.

OUTPUT

TSTACK - temp stack of which 22 words are used temporarily for output buffer.

INTERACTION

Module used:

LIST - list message on OC device (LIST), set list flag in the JIT for LL device (SCCLF), and list control command on LL device (CCLIST).

SUBROUTINE

MESSR20 This routine moves the message from the control command to the temporary buffer for output. The output is formatted as a TEXTC string.

DESCRIPTION

The message portion of the command is output on the OC device preceded by !!MESSAGE. The command itself is output on the LL device.

UTS TECHNICAL MANUAL

The MESSAGE control command must be in the control command buffer on entry.
Exit is made to (SR4) + 1.

UTS TECHNICAL MANUALID

TITLR - Process TITLE control command

PURPOSE

To process TITLE control command and move title to buffer.

USAGE

Calling sequence

BAL, SR4 TITLR

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

INPUT

ATITLE - displacement in JIT for address of title buffer

INTERACTION

Module used:

LIMR - move title to title buffer (MESSR20 in routine MESSR)

DESCRIPTION

The portion of the TITLE control command following the command field is moved to the title buffer whose address is contained in the JIT. The number of bytes moved is stored into the first byte of the title buffer. The command is output onto the LL device.

The TITLE control command must be in the control command buffer on entry. Exit is made to (SR4) + 1.

UTS TECHNICAL MANUAL

ID

POOLR - Process POOL control command

PURPOSE

To process the POOL control command and store values into the JIT.

USAGE

Calling sequence

BAL, SR4 POOLR

Error return

Normal return

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return is taken, contains error code.

OUTPUT

JB:NFPOOL byte in JIT for number of FPOOLS.

JB:NIPOOL byte in JIT for number of IPOOLS.

INTERACTION

Modules used:

LIST - set listing flag in the JIT for the LL device (SCCLF).

SUBR - get value following keyword (GETDECVAL), and scan to end of control command (EOCCSCAN).

CHARROUT - compare current character (CHARSCAN), and compare character strings (QUOTSCAN).

ERRORS

Errors detected in syntax, keywords, or specified values cause an error exit to be taken with registers SR3 set as described above.

UTS TECHNICAL MANUAL

DESCRIPTION

The POOL control command is processed and the values obtained are checked against a lower bound of 2 and an upper bound of 31, and if found within bounds are stored in the JIT. The command is output on the LL device. If an error is detected in processing the command, SR3 is set as shown above and the error exit is taken.

The POOL control command must be in the control command buffer on entry. Exit is made to (SR4) + 1 for normal return; otherwise exit is to (SR4).

UTS TECHNICAL MANUALID

SWITCHR - process SWITCH control command

PURPOSE

To process a SWITCH control command and set the specified sense switches in the JIT.

USAGE

Calling sequence

BAL, SR4 SWITCHR

Error return

Normal return

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return is taken, contains error code.

INPUT

PLB - displacement in CRPT containing current keyword.

CSL - displacement in CRPT containing number of characters in value field.

OUTPUT

SS - displacement of sense switch word in the JIT. Bits are set or reset as specified on command.

INTERACTION

Modules used:

- LIST - set listing flag in the JIT for the LL device (SCCLF).
- CHARROUT - compare current character (CHARSCAN), get decimal field (DECSCAN), and get keyword from command (NAMSCAN).
- SUBR - scan to end of control command (EOCCSCAN).

ERRORS

Errors detected in syntax, keywords, or specified values cause an error exit to be taken with registers SR3 set as described above.

UTS TECHNICAL MANUALDESCRIPTION

The SWITCH control command is processed and the pseudo sense switches in the JIT are set as specified. If a sense switch is specified on the SET and RESET options, the switch setting will be that of the last option specified. The command is output on the LL device. If an error is detected in processing the command, SR 3 is set as shown above and the error exit is taken.

The SWITCH control command must be in the control command buffer on entry. Exit is made to (SR4)+ 1 for a normal return; otherwise exit is to (SR4).

UTS TECHNICAL MANUALID

ASSGR - Process ASSIGN control command

PURPOSE

To process an ASSIGN control command. The assign-merge record for the user is read, updated, and written back to RAD storage.

USAGE

Calling sequence

BAL, SR4 ASSGR

Error return

Normal return

Input registers

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return, contains error code

INPUT

- CJOB - contains address of JIT
- CSL - displacement in the CRPT for current string length
- J:AMR - contains 0 if no assign-merge record exists and CCI must create one; contains non-zero value if record exists and is to be updated
- M:BI - DCB for binary input
- M:C - DCB for control input
- M:LL - DCB for control command output
- M:LO - DCB for listing output
- M:X1 - DCB used for reading and writing assign-merge record
- PLB - displacement in CRPT for buffer containing current keyword or value

OUTPUT

- CCBEF - displacement in JIT for certain bit flags. Bit 8 is set to indicate that a control command has been read but not yet processed.
- CHTBL - address of character table. Entry for hyphen is set and reset to allow use of hyphen as a non-delimiter.
- DELFLAG - flag in data area set to -1 if assign-merge table is not to be updated. Flag is set if no options are present on an ASSIGN command.
- FLAGS - displacement in CRPT for flag word. Buffer full flag and blank out flag are set and reset.

UTS TECHNICAL MANUAL

- J:ASSIGN - Bit 22 of this word is set after the updated assign-merge record has been written.
- J:JIT - address of JIT. The file extension bit is reset.
- TSTACK - address of temp stack which is used for saving registers.

INTERACTION

Modules used:

- SUBR - Test for legal expiration date (CALENDTE), shift character string (CHSTSHFT), scan to end of command (EOCCSCAN), get account number (GETACCN), get value following keyword (GETDECVAL), get password (GETPASSW), get serial number (GETSN), and search word table (WDTBLSRH).
- LIST - Set control command listing flag in the JIT (SCCLF).
- CHARROUT - Compare current character (CHARSCAN), scan decimal number (DECSCAN), get name (NAMSCAN), get next active character (NXACTCHR), and compare two character strings (QUOTSCAN).
- CCIR - Read next control command (CCREAD).

Monitor services used (handwritten CALs and FPTs):

- M:GP - get a page for data area. One page is obtained to construct the PLIST for the ASSIGN command; a second page is obtained for reading in the assign-merge record.
- M:FP - free page obtained by M:GP above.
- RAMR - read assign-merge record.
- WAMR - write assign-merge record.

Monitor services used (procedures):

- M:CLOSE - close M:X1.
- M:SETDCB - enter error and abnormal addresses in M:X1.
- M:TFILE - declare temporary GO file.

SUBROUTINES

- MLPOOP This routine moves the DCB name from the ASSIGN PLIST to the Adjust DCB PLIST. Entered with R5 containing numbers of words to be moved, R2 containing 'to' address, and R3 containing 'from' address.
- MPLISTP This routine moves the ASSIGN PLIST to the Adjust DCB PLIST. On entry, R5 contains the number of words in the PLIST, R2 contains the 'to' address, and R3 contains the 'from' address.
- MPLISTP2 This routine moves the ASSIGN QLIST (device parameters) to the assign merge record. On entry, R5 contains a 7 (the maximum number of words to move) and registers R2 and R3 are set as described above under MPLISTP.

UTS TECHNICAL MANUAL

- VARLIST This routine moves a variable list parameter from the ASSIGN PLIST to the Adjust DCB PLIST. Input registers are set as described above under MLPOOP.
- SETHYPH This routine resets the entry for hyphen in the character table to make it a delimiter.

ERRORS

Errors detected in keywords, syntax, or specified parameters cause an error exit to be taken with register SR3 containing the error code.

DESCRIPTION

A data page for the Assign Parameter List (PLIST) is obtained with the M:GP call. Then the ASSIGN control command is processed and the parameters are stored into the Assign PLIST. The command is output on the LL device. As the command is processed and each keyword is obtained from the command, it is used to search a keyword table. The position in the keyword table is used as index into the branch table to go to the appropriate code to process the keyword. The value associated with the keyword is obtained and stored in the Assign PLIST. If the ASSIGN command is for M:LL, M:C, M:BI, or M:LO, the information is also merged into the corresponding DCB in the assign merge record; for other DCBs the M:X1 DCB is assigned.

The DCB is closed before the merge is done. If the ASSIGN command is for M:OC, M:SL, or M:AL or if M:C is being assigned to a file, the error exit is taken. If the ASSIGN command is for M:GO, it must be assigned to a file or the error exit is taken.

If any variable length file parameters are present on the ASSIGN control command, they are added to the end of the Assign PLIST in the order in which they are encountered. The format for these parameters is the same as in the M:OPEN parameter list. Space is always allocated for all fixed length parameters.

The ASSIGN control command must be in the control command buffer on entry. The Assign PLIST constructed from the parameters on the command is merged with the assign merge record which is read in from RAD storage. If an assign merge record does not exist for this user (J:AMR=0), one is initialized. Before writing out the assign merge record, the next control command is read to see if it is another ASSIGN command. If so, ASSGR is re-entered at the top, the routine for reading the assign merge record is bypassed, and parameters from this command are merged into the assign merge record. Thus parameters from all successive ASSIGN commands are merged before the assign merge record is written.

If an error occurs during processing, exit is made to (SR4) with SR3 containing the error code. Normal exit is to (SR4)+1.

UTS TECHNICAL MANUAL

The format of the Assign PLIST constructed from the Assign parameters is described in Section VL.

If any variable length file parameters (e.g. FILE, LABEL, PASSWORD, etc) are present on the ASSIGN control command they are added to the end of the Assign PLIST in the order in which they are encountered. The format for these parameters is the same as in the M:OPEN parameter list as described in the UTS/BP Reference Manual. Space is always allocated for all fixed-length parameters.

UTS. TECHNICAL MANUAL

ID

LOADR - Process LOAD and OVERLAY control commands.

PURPOSE

To process a LOAD or OVERLAY control command. The LOAD CC table (LOCCT) and ROM table (ROMT) are built and the address of the LOCCT is entered in the JIT.

USAGE

Calling sequence

BAL, SR4 LOADR

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

R6 - address of LOCCT

SR3 - if error return, contains error code

INPUT

- ACCN - displacement in JIT for account number of job
- CSL - displacement in CRPT containing byte count of current field
- J:BUP - address containing program lower limit
- PLB - displacement in CRPT for buffer containing current parameter or keyword
- SYSID - displacement in JIT for system ID

OUTPUT

- ALOCCT - displacement in JIT which is set to the location of the LOCCT created

TSTACK - temp stack used for saving registers

Load module name and password are save in the Assign-Merge record

INTERACTION

Modules used:

- SUBR - Test for legal expiration date (CALENDTE), shift character string

UTS TECHNICAL MANUAL

- (CHSTSHFT), scan to end of command (EOCCSCAN), get account number (GETACCN), get value following keyword (GETDECVAL), and get password (GETPASSW).
- LIST - Set control command listing flag in the JIT (SCCLF).
- CHARROUT - Scan hexadecimal number (HEXSCAN), get name (NAMSCAN), and compare character strings (QUOTSCAN).

Monitor services used (handwritten CAL and FPT):

- M:GP - Get 2 pages for LOCCT and ROMT

SUBROUTINES

- LLEGALNM Tests for legal alphanumeric name of not greater than 8 characters. NAMSCAN, in module CHARROUT, is called to scan the name and place the length in CSL in the CRPT. LLEGALNM is entered with the CRPT address in R7, SR1 set to zero, and the return address in SR4. On a normal return, CCI is set to 0; otherwise it is set to 1.
- LLEGALPW Tests for legal password of not greater than 8 characters. GETPASSW, in module SUBR, is called to scan the password and place the length in CSL in the CRPT. LLEGALNM is entered with the CRPT address in R7, SR1 set to zero, and the return address in SR4. On a normal return, CCI is set to 0; otherwise it is set to 1.
- CONEBCD Converts a maximum of 8 bytes EBCDIC to hexadecimal. On entry R0 contains the address of the first word to be converted, R1 contains the byte count, and SR4 contains the return address. The converted number is in R3 on exit.
- PLBNMSH Shifts the name in PLB in the CRPT one byte to the right. CHSTSHFT in module SUBR is actually called to perform the shift and to insert the string length (in CSL in the CRPT) into the first byte of the buffer, thus creating a TEXTC format.
- STARNAME This routine constructs a monitor assigned module name, stores the name, job account and zeroed password into the area whose beginning address is in R4. On entry, the JIT address is in R5, return address in SR4, and appropriate alphanumeric character (for SYSDX) in low order byte of D2. The LM name (formed by the system ID in the JIT and the character in D2) is stored into the address pointed to by R4, the second and third words are blanked, and the account number from the JIT is stored into the next two words by calling LACCTST1. A two-word blank entry is stored as the password by calling LACCTST1. STARNAME is called to add these entries to the LOCCT or ROM Table.

UTS TECHNICAL MANUAL

LKEY This routine identifies a keyword. If a keyword is not identifiable or if it duplicates another on the command an error exit is taken. Otherwise the routine branches to the associated processing routine for the keyword.

LACCTST Moves a 2-word account number from the PLB in the CRPT to the address specified by R4.

LACCTST1 Moves a 2-word account number from the address specified by R0 to the address specified by R4.

STARNM1 This routine functions much the same as STARNAME (see above description) except only the job account and a zeroed password are stored into the LOCCT or ROM Table.

TCOMMA Test if current delimiter is a comma by calling CHARSCAN in module CHARROUT to compare characters. CC1 is set to 0 if compare is successful; otherwise CC1 is set to 1.

GETROMAD The next available space in the ROM Table is computed, stored in ROMTAD in the LOCCT, and returned in R4 on exit.

ERRORS

Any errors detected in syntax, keywords, or values cause an error return with the error code in SR3. Error return is to (SR4), whereas a normal return is to (SR4)+3.

DESCRIPTION

The LOAD (OVERLAY) control command is processed and the parameters are stored in the LOCCT and the ROMT. The command is output on the LL device. If an error is detected in processing the LOAD (OVERLAY) command, SR3 is set as shown above and the error exit is taken. As the command is processed and each keyword is obtained from the command, the keyword is used to search the keyword table. The position in the keyword table is used as an index into the branch table to go to the

code to process the keyword.

All module names following the keywords EF and SYSLIB are stored in the ROMT. The ROMT is built starting from the high end of the three-page area. The rest of the parameters are stored in the LOCCT. The LOCCT starts from the low end of the three-page area. The space for the LOCCT and ROMT is obtained via an M:GP call to the Monitor to get three pages.

UTS TECHNICAL MANUAL

The format of the Load Control Command Table (LOCCT) is shown below.

| | 0 | 1 | 2 | 7 | 8 | 11 | 12 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31* |
|---------|--|---|---|---|---|----|----|-------|----|----|--|----|----|----|----|----|----|----|----|----|-----|----|----|-----|
| LOCCT 0 | R | Q | | | | | | SL | P | O | N | M | L | K | J | I | H | G | F | E | D | C | B | A |
| 1 | Displacement from LOCCT to ROMT | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Displacement from LOCCT to TREET | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 0 | | | | | | | SYSid | | | | | | | | | | | | | | | | |
| 4 | REF or BREF count (default 0) | | | | | | | | | | LOAD BIAS (WA) | | | | | | | | | | ** | | | |
| 5 | ** | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | FCOM size | | | | | | | | | | FCOM (DA) | | | | | | | | | | *** | | | |
| 7 | ERSTACK size (default X'A') | | | | | | | | | | ERTABLE size (default X'A') | | | | | | | | | | | | | |
| 8 | TSS size (default X'40') | | | | | | | | | | Number of UNSAT accounts (default 0 if NOSYSLIB; -1(:SYS) if not) | | | | | | | | | | | | | |
| 9 | Number of READ accounts (default 1) | | | | | | | | | | Number of WRITE accounts (default 1) | | | | | | | | | | | | | |
| 10 | Load Module Name, TEXTC followed by blanks. (default three characters SYSid L) | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | User account number | | | | | | | | | | | | | | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | Load Module Password (default 0,0) | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | EXPIRE date; 'mmd bb yy' or 'NEVER' (BPM) | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | 0,0 (UTS) | | | | | | | | | | | | | | | | | | | | | | | |
| 19 | Library password, i. e., (PERM, LIB, password) | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | (default 0,0) | | | | | | | | | | | | | | | | | | | | | | | |
| 8 max. | READ account numbers 2 words per acct. #, (default 'ALL bbbb ') | | | | | | | | | | | | | | | | | | | | | | | |
| 8 max. | WRITE account numbers 2 words per acct. #, (default 'NONE bbbb ') | | | | | | | | | | | | | | | | | | | | | | | |
| | UNSAT account numbers and passwords 4 words per (acct #,pass). (default password 0,0) | | | | | | | | | | | | | | | | | | | | | | | |
| TREET → | Total number of words in Tree Tables | | | | | | | | | | | | | | | | | | | | | | | |
| | Tree Tables 1 Table per Segment 11 words per table (see Figure 12 for format) | | | | | | | | | | | | | | | | | | | | | | | |
| ROMT → | ROM Tables 1 Table per ROM 7 words per table (see Figure 11 for format) | | | | | | | | | | | | | | | | | | | | | | | |

UTS TECHNICAL MANUAL

Footnotes for the LOCCT diagram:

- *NOTE:
- | | |
|-------------------------------|---|
| A = 1, MAP specified | K = 1, GO specified |
| B = 1, NOSYSLIB specified | L = 1, BI specified |
| C = 1, REF specified | M = 1, CSECT was specified |
| D = 1, PERM specified | N = 1, NOTCB specified |
| E = 1, LIB specified | O = 1, XMEM in effect (set by the Loader in IN2) |
| F = 1, M10 specified | P = 1, MAP, VALUE specified |
| G = 1, M100 specified | = 0, MAP, NAME specified |
| H = 1, FCOM specified | Q = 1, BREF specified |
| I = 1, ABS specified | R = 1, EF specified |
| J = 1, Assigns Read | |
| SL = Sev. Level (default = 4) | |

**BPM-UTS differences in the LOCCT Tables:

| <u>Word</u> | <u>BPM</u> | <u>UTS</u> |
|-------------|--|--|
| 4 | LOAD BIAS field Default = 0 | LOAD BIAS Default = background lower limit WA |
| 5 | Background lower limit | ----- |
| *** 6 | Passed to the Loader in Reg. D4 (D4) = FCOM size | |

UTS TECHNICAL MANUAL

The format of the Relocatable Object Module Table (ROMT) is shown below:

Entry Format

| | |
|------|------------------|
| word | |
| 0 | EF NAME IN |
| 1 | TEXTC FORM |
| 2 | FLAG |
| 3 | ACCOUNT |
| 4 | NUMBER (2 words) |
| 5 | PASSWORD |
| 6 | (2 words) |
| 7 | LINK |

where

FLAG is 0 if this ROM is last ROM entry in table; X'40' if this is not last ROM entry in table.

LINK is address of the ROMT entry that contains the name of the next ROM in the segment, or zero if this is last entry in chain. This field is filled in by the TREER routine which processes the TREE command. All modules allocated to a particular segment are linked together in the ROMT.

UTS TECHNICAL MANUAL

ID

TREER - process TREE control command

PURPOSE

To process a TREE control command and build the Tree table.

USAGE

Calling sequence

BAL, SR4 TREER

Error return

Normal return

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1- current character or zero

Output registers

SR3- if error return, contains error code.

INPUT

ALOCCT - displacement in JIT containing address of the LOCCT built by the preceding LOAD or OVERLAY command.

CSL - displacement in the CRPT containing current string length.

PLB - displacement in the CRPT for buffer for current string.

INTERACTION

Modules used:

CHARROUT - compare current character (CHARSCAN), and scan name field (NAMSCAN).

LIST - set list flag in JIT for LL output (SCCLF).

SUBR - change name to TEXTC format (CHSTSHFT) and scan to end of command (EOCCSCAN).

SUBROUTINES

RTR This routine is a recursive routine. It is initially called by the TREER routine when the character combination '-(' is encountered in processing the TREE command. This routine calls itself each time it encounters the

UTS TECHNICAL MANUAL

character combination '-' in processing the command, and exits each time it encounters a ')' character. Each time a new segment is found on the command, a new entry in the Tree Table is allocated on the end, and the segment name is stored in the entry. The segment name is the name of the first ROM in the segment.

The ROMs associated with each segment are linked together in the ROMT and to the TREE Table.

If an error condition is encountered, SR3 is loaded with an error code and the error exit is taken.

Registers on entry are:

- R1 - address of current tree table entry
- R2 - address of current back link tree table entry
- R3 - address of current overlay link tree table entry
- D1 - base address of ROM Table
- D2 - base address of TREE Table
- D3 - end address of ROM Table

See TREE Table format below.

GETROMNAME This routine is called by TREER to get the ROM name from the TREE control command. The next field is obtained from the TREE command being processed, and checked if it is a legal ROM name. Then the ROMT is searched for that name. If found, the ROM in the ROMT is linked to the other ROMs in the segment. If the ROM is the first ROM in the segment, the ROM's name is stored in the current Tree table entry. If the ROM name is not a legal ROM name (one to eight alphanumeric characters) or is not found in the ROMT, SR3 is loaded with an error code and the error exit is taken.

Registers on entry are the same as for RTR.

ERRORS

Any errors encountered during processing cause a return to (SR4) with the error code in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

The TREE control command must immediately follow a LOAD or OVERLAY control command. Therefore, the LOCCT and ROMT will have been built when the TREE routine builds the entry for the first segment (root segment) in the Tree table. This is done by processing the first part of the TREE command which specifies the names of ROMs (Relocatable Object Modules) that make up the root segment of the program. Processing the root segment portion of the TREE command is terminated when

the two characters '-(' are encountered; the RTR routine is called to process the remainder of the TREE command.

As the Tree table is built, the ROMs in the ROMT associated with each segment are linked together and to the entry in the Tree table with which they are associated.

The TREE command is output on the LL device. If an error is encountered in processing the TREE command, SR3 is set as shown above and the error exit is taken.

The TREE Table format is shown below.

Each overlay segment in the program has one entry in the Tree table. The first word of the table contains the total size of the table in words. If no TREE command follows the LOAD or OVERLAY command, a default Tree table is built containing only one entry since the program consists of only a root segment with no overlays. The segment name is the load module name.

Entry Format

| | | |
|------|-----------------|-------|
| word | SEGMENT NAME IN | |
| 0 | | |
| 1 | TEXTC | |
| 2 | FORMAT | |
| 3 | RLINK | BLINK |
| 4 | SUBLINK | OLINK |
| 5 | LLINK | |
| 6 | 0 | |
| 7 | 0 | |
| 8 | 0 | |
| 9 | 0 | |
| 10 | 0 | |

where

RLINK is index into the ROMT to entry that contains name of first ROM in segment.

BLINK is back link — index to the Tree table entry for which this entry is a sublink.

SUBLINK is index to the Tree table entry that is the first sublink of this segment.

OLINK is overlay link — index to the Tree table entry that overlays this segment. If several segments overlay each other, they are chained through this entry. The last segment has zero OLINK.

LLINK is index into the ROMT to the entry that contains the name of the last ROM in the segment.

ID

PTREER - process PTREE control command

PURPOSE

To process the PTREE control command and read the first record from the file specified on the command.

USAGE

Calling sequence

BAL, SR4 PTREER

Input registers

R5 - address of .JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return, contains error code.

INPUT

- ALOCCT - displacement in JIT containing address of the LOCCT built by the preceding LOAD or OVERLAY command.
- ACCBUF - displacement in JIT containing address of command buffer.
- CREAD - address of read routine in CCIR which is set as an address in the CRPT for reading continuation records.
- CSL - displacement in the CRPT containing current string length.
- M:X1 - DCB through which the TREE command is read.

OUTPUT

- CBUF - displacement in CRPT for address of buffer used for reading commands. Set and reset for reading TREE command from file.
- CCP - displacement in CRPT for current character position in command buffer. Initialized to 1 before reading TREE command.
- CCRSZ - displacement in CRPT for word containing size of TREE command record.
- CONTR - displacement in CRPT for address of routine to read continuation records. Initialized to RDTR before reading TREE command. Reset to CREAD.
- OUTR - output routine address which is set to the address of CCLIST.

UTS TECHNICAL MANUAL

- RUNFLAG - displacement in JIT for run flag. Bit 10 is set or reset.
 TSTACK - address of temp stack which is used for saving registers.

INTERACTION

Modules used:

- CHARROUT - compare current character (CHARSCAN) and get name from command (NAMSCAN).
 SUBR - format file name to TEXTC format (CHSTSHFT), scan to end of command (EOCCSCAN), get account number (GETACCN), and get password (GETPASSW).
 OPNSTARF - open file containing TREE command (OPNF).
 TREER - process TREE command (TREER).

Module services used (handwritten CALs and FPTs):

- M:READ - read TREE record through M:X1 DCB.
 M:CLOSE - close M:X1 DCB.

ERRORS

Any errors encountered in processing cause an error return to (SR4) with the error code contained in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

The PTREE control command is processed and checked for errors. If the control command is not in error, the file specified on the command is opened and the first record is read into the control command buffer. The address of the read routine in the CRPT is changed to that of the read routine for reading the file; then the TREER routine is called to process the TREE control command being read from the file.

After return from the TREER routine, the file is closed and the CRPT is restored. If an error occurs in processing the PTREE command or the TREE command in the specified file, an error return is given, with SR3 set as shown above.

UTS TECHNICAL MANUALID

TELESCOPE - Pack the LOCCT, Tree Table and ROMT into one table.

PURPOSE

To compact the LOCCT, Tree Table, and ROM Table by eliminating unused space.

USAGE

Calling sequence

BAL, SR4 TELESCOPE

Input register

R5 - address of JIT

INPUT

ACCN - displacement in JIT for job account number
 ALOCCT - displacement in JIT for address of LOCCT
 SYSID - displacement in JIT for system ID

OUTPUT

TSTACK - temp stack for saving registers

SUBROUTINES

TBLSHFT This routine moves a table entry. Entered with word count in D2, source address in D3, and destination address in D1.

STARNAME This routine constructs a monitor-assigned module name from the system ID and a 'G' or 'B' supplied on entry (i. e., SYSIDG or SYSIDB). This name, job account, and zeroed password are stored at a specified location. On entry, the destination address is in R4, the JIT address is in R5 (system ID and account are taken from the JIT), and the appropriate character to append to the name ('G' for GO or 'B' for BI) is in the low order byte of D2.

LACCTST1 This routine stores a two-word value (account or password) in the specified location. Entered with source address in R0 and destination address in R4.

DESCRIPTION

The LOCCT is condensed by eliminating any unused entries in the space allocated for the READ and WRITE accounts. The Tree table is packed next to the LOCCT. The ROMT is moved from the upper end of the two-page area to the area following the Tree table. When the ROMT is moved, it is ordered and sorted by segment.

Each segment in the Tree table has a pointer to the chain of ROM (Relocatable Object Module) names in the ROMT which make up the segment. The ROMT is moved by going through the Tree table segment by segment, and moving the ROMs that make up the segment.

See Section PA.09 for the format of the original LOCCT and ROMT. The format of the Tree table is shown in Section PA. 10.

UTS TECHNICAL MANUAL

ID

RUNR - Process RUN control command

PURPOSE

To process the RUN control command and build the RUN table

USAGE

Calling sequence

BAL, SR4 RUNR.

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return, contains error code.

INPUT

ACCN - displacement in JIT which contains job account number.

CSL - displacement in CRPT for byte length of current string.

PLB - displacement in CRPT for buffer containing current parameter.

OUTPUT

ALOCCT - displacement in JIT which is set to the location of the RUN Table.

RUNFLAG - displacement in JIT for word in which bits 20-23 are set to the XSL value specified on the run command.

TSTACK - address of temp stack which is used for saving registers. A portion of this area is used for building the RUN table.

INTERACTION

Modules used:

SUBR - Scan to end of command (EOCCSCAN), get account number (GETACCN), get address value (GETAVAL), get value following keyword (GETDECVAL), get hexadecimal field (GETHEXVAL), get password (GETPASSW), format LM name into TEXTC string (GETTEXT)

UTS TECHNICAL MANUAL

and get name and value fields of START option (GETLOC2 in GETLOC).

- LIST - Set list flag in JIT for output to LL device (SCCLF).
 CHARROUT - Compare current character (CHARSCAN), get name from command (NAMSCAN), and compare two character strings (QUOTSCAN).

ERRORS

Any errors detected in processing cause a return to (SR4) with the error code in SR3. Normal return is to (SR4) + 4.

DESCRIPTION

The RUN command must be in the control command buffer on entry. The RUN control command is processed and the parameters are stored in the RUN table. The command is output on the LL device. If an error is detected in processing the RUN command, SR3 is set as shown above and the error exit is taken.

As the command is processed and each keyword is obtained from the command, it is used to search the keyword table. The position in the keyword table is used as an index into the branch table to go to the code to process the keyword.

The RUN table is built in a portion of the temp stack (TSTACK) with space allocated for all of the parameters. After the RUN command is processed, the table is compacted by eliminating any unused space for the START parameter and the real-time parameters RBLOCK and TIME.

If there is no LMN parameter on the RUN command, the name is obtained from an entry in the ASSIGN-MERGE RECORD. This entry contains the name of the last load module formed for the current job. If no load module was formed during the job, then the JITLMN entry in the JIT is zero. If a zero load module name is put in the RUN table, an error will occur when the open request is made for the load module with a zero name. An error message will be output and the job will be aborted.

CCI requests the Monitor to load the program specified in the RUN table by exiting to the Monitor and giving the address of the RUN table in the ALOCCT word in the JIT.

UTS TECHNICAL MANUAL

Format of RUN Table

| Word | 0 | 1 | 7 | 8 | 15 | 16 | 23 | 24 | 31 | | | | | | | |
|------|--|----------------|----------------|----------------|----------------|----|----------------|----|-----|----------------|----------------|--|--|--|------|----|
| 0 | F ₁ | X ₁ | | | N ₁ | | N ₂ | | XSL | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | BIAS | 31 |
| 2 | F ₂ | | | | | | | | | | | | | | | |
| 3 | LOAD MODULE NAME | | | | | | | | | | | | | | | |
| 4 | (LMN) IN | | | | | | | | | | | | | | | |
| 5 | TEXTC FORMAT | | | | | | | | | | | | | | | |
| 6 | ACCOUNT | | | | | | | | | | | | | | | |
| 7 | NUMBER | | | | | | | | | | | | | | | |
| 8 | PASSWORD | | | | | | | | | | | | | | | |
| 9 | | | | | | | | | | | | | | | | |
| A | SYMBOLIC START ADDRESS IN TEXTC FORMAT (1 to 16 WORDS) | | | | | | | | | | | | | | | |
| 1A | ± HEX START ADDRESS ADDEND OR ZERO | | | | | | | | | | | | | | | |
| 1B | INTERRUPT OR CLOCK ADDRESS | | | | | | | | | | | | | | | |
| 1C | P ₁ | P ₂ | P ₃ | P ₄ | P ₅ | | | | | P ₆ | P ₇ | | | | | |
| 1D | 0 | | | | | | | | | | | | | | | |
| 1E | RBLOCK | | | | | | | | | | | | | | | |
| 1F | TIME VALUE | | | | | | | | | | | | | | | |
| 20 | IOD | | | | | | | | | | | | | | | |

} zero if no password

where

- F₁ is 1 if real-time parameters are in table.
- X₁ is relative position of real-time parameters in table.
- N₁ is number of SNAP, SNAPC, IF, AND, OR, and COUNT records in System Debug File.
- N₂ is number of MODIFY records in System Debug File
- XSL is value of XSL parameter.
- BIAS is value of BIAS parameter.

UTS TECHNICAL MANUAL

- F₂ is 0 if System Debug File is not present; 1 if System Debug File is present.
P₁ is 1.
P₂ is 0 if no RBLOCK value is present; 1 if RBLOCK value is present.
P₃ is 0 if no TIME value is present; 1 if TIME value is present from CLOCK or TIME option.
P₄ is 0 if no IOD value is present; 1 if IOD value is present.
P₅ is 0 if TIME value is associated with CLOCK option; 1 if TIME value is associated with TIME option.
P₆ is 0 if DISABLE option not specified; 1 if DISABLE option specified.
P₇ is 0 if DIRECT option not specified; 1 if DIRECT option specified.

Note: Words 1B - 20 are unused in UTS-A00 and A01.

UTS TECHNICAL MANUALID

DEBUGR - process INCL and debug control commands

DESCRIPTION

This module contains the routines for the processing of the INCL control command and the debug control commands: MODIFY, PMD, PMDI, PMDE, SNAP, SNAPC, IF, AND, OR and COUNT. In the processing of the debug commands the information is stored in a table and written to the system debug file which is a keyed file known as star 'D' file. The file name is a three-character name consisting of the current job ID from the JIT followed by the letter D. Each table of information (record) is written with a unique key. The key consists of the segment name from the Tree table to which the debug control command applies, with a digit appended. Before any debug commands are processed, the byte following the last character of each segment name in the Tree table is set to zero and the byte count for the name is incremented by one. As the different debug commands are processed, the last byte of the segment name in the Tree table is incremented each time a record of debug information is written into the debug file for that segment.

When debug control commands follow the RUN command, the Tree table from the load module to be loaded is read in before any debugs are processed.

The MODIFY debug command is the only debug command which may follow the LOAD and TREE commands.

UTS TECHNICAL MANUAL

ID

INCL - Process the INCL control command.

PURPOSE

To process the INCL control command and store each ROM name in the ROM Table.

USAGE

Calling sequence

BAL, SR4 INCL

Error return

Normal return

Input registers

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output registers

SR3 - if error return, contains error code.

INPUT

ALOCCT - displacement in JIT for address of LOCCT.

PLB - displacement in CRPT for buffer containing ROM name.

CSL - displacement in CRPT for word containing byte length of ROM name.

INTERACTION

Modules used:

SUBR - change ROM name to TEXTC format (CHSTSHFT), and scan to end of command (EOCCSCAN).

CHARROUT - get ROM or segment name (NAMSCAN), and check for comma (CHARSCAN).

SUBROUTINES

SEGSRCH This routine gets the segment name from the control command, searches the tree table for the segment name and checks for a comma following the segment name. On entry, R1 contains the address of the tree table, R7 the address of the CRPT, SR1 the current character or zero, and D4 the end address of the tree table. On exit, R1 contains the address of the tree table entry if found (CC1 = 0) or SR3 contains the error code if

entry is not found (CC1 = 1).

ERRORS

Any errors detected in processing cause a return to (SR4) with the error code in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

The INCL control command to be processed must be in the control command buffer. SEGSRCH is called to search the tree table for the segment name in the command. Each ROM name in the command is stored in the ROMT and linked to the other ROMs in the segment. If an error is detected in processing the command, the registers are set as shown above and the error exit is taken.

UTS TECHNICAL MANUALID

MODFYR - Process the MODIFY control command

PURPOSE

To process a MODIFY control command and build the Modify table.

USAGE

Calling sequence

BAL, SR4 MODFYR

Error return

Normal return

Input registers

R3 - contains 2, code for MODIFY command

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - if error return, contains error code

OUTPUT

TSTACK - temp stack used for building Modify table and for saving registers.

INTERACTION

Modules used:

SUBR - get modify location and relocation name (GETLOC2 and GETLOC1 in GETLOC), get hexadecimal field (GETHEXVAL), and scan to end of command (EOCCSCAN).

CHARROUT - check for comma or '+' (CHARSCAN).

DEBUGR - write debug record (WRTDEBUG).

ERRORS

Any errors detected cause return to (SR4) with the error code in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

The MODIFY control command to be processed must be in the control command buffer. The two top words of the stack used by CCI must contain the address and the address

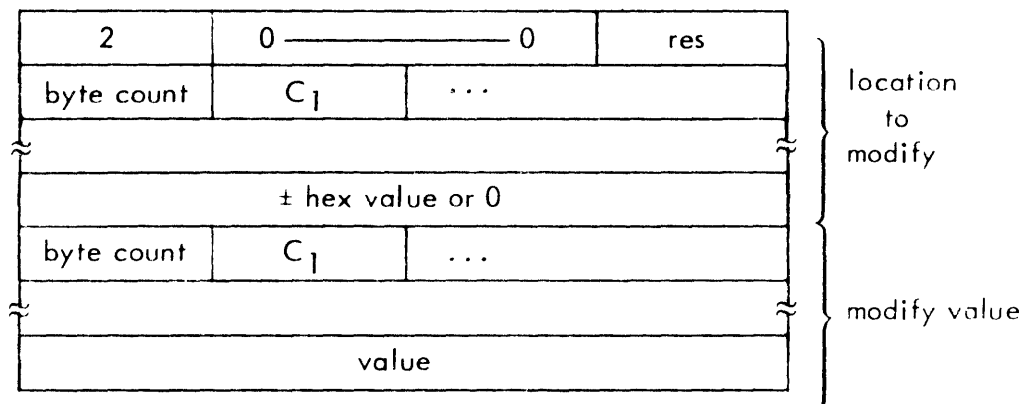
UTS TECHNICAL MANUAL

plus one at which the Modify table is to be built when the command is processed.

The MODIFY control command is processed and the information is stored in the Modify table. There is one Modify table for each Modify value on the command, and each Modify table is written as one record into the system debug file by subroutine WRTDEBUG. If an error is detected in processing the command, SR3 is loaded with an error code and the error exit is taken.

Format of Modify Table

The Modify Table is written into the system debug file for the current job.



where

res = 0 if byte resolution; = 1 if halfword resolution; = 2 if word resolution; = 3 if doubleword resolution. The resolution applies to the name specified with the modify value.

value is value to be given to the location to be modified. It has the form value + res (name) name is the name of an external definition and value is a 1-digit to 8-digit hexadecimal number.

UTS TECHNICAL MANUALID

PMDR - process PMD, PMDE, and PMDI control commands.

PURPOSE

To process a PMD, PMDE, or PMDI control command and build the PMD table.

USAGE

Calling sequence

BAL, SR4 PMDR

Error return

Normal return

Input registers

R3 - code for debug command: 3 for PMD, 4 for PMDE, 5 for PMDI

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - if error return, contains error code

INPUT

- CSL - displacement in CRPT for word containing current field length.
 PLB - displacement in CRPT for buffer containing current field of command.

OUTPUT

- FLAGS - displacement of flag word in CRPT. Bit 2 is set to indicate parameter buffer is full before scanning for decimal value.
 ALOCCT - displacement in JIT for address of LOCCT or RUN table. Count of dumps in RUN table is incremented for each PMD type command.
 TSTACK - temp stack used for building PMD table and for saving registers.

INTERACTION

Modules used:

- CHARROUT - check for expected delimiters (CHARSCAN) and scan decimal number (DECSCAN).
 SUBR - get name and value from command (GETLOC2 in GETLOC), convert EBCDIC to hexadecimal (BCDHEX), scan to end of command (EOCCSCAN), and get hexadecimal field (GETHEXVAL).
 DEBUGR - write debug record (WRTDEBUG).

UTS TECHNICAL MANUAL

ERRORS

Any errors detected cause return to (SR4) with the error code in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

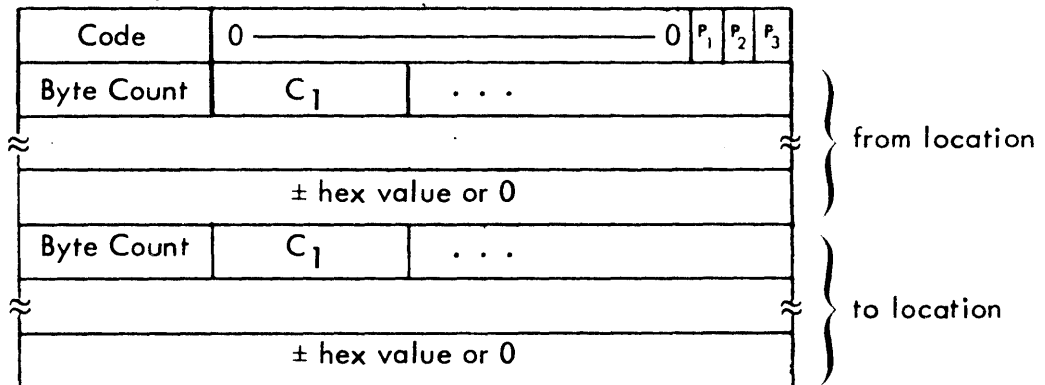
The PMD (PMDI, PMDE), control command must be in the control command buffer. The two top words of the stack used by CCI must contain the address and the address plus one at which the PMD table is to be built when the command is processed.

The PMD (PMDI, PMDE) control command is processed and the information is stored in the PMD table. There is one PMD table for each 'from - to' specified on the command and each PMD table is written as one record into the system debug file by subroutine WRTDEBUG.

If an error is detected in processing the command, SR3 is loaded with an error code and the error exit is taken.

Format of PMD, PMDE, PMDI Table

The PMD, PMDE, PMDI Table is written into the system debug file for the current job.



where

Code is 3 for PMD command, 4 for PMDE command, and 5 for PMDI command.

P₁ = 1 for dump protection type 00.

P₂ = 1 for dump protection type 01.

P₃ = 1 for dump protection type 10.

from location is beginning dump location

to location is ending dump location

UTS TECHNICAL MANUALID

SNAPR - Process SNAP and SNAPC control commands

PURPOSE

To process a SNAP or SNAPC command and build the SNAP table.

USAGE

Calling sequence

BAL, SR4 SNAPR

Error return

Normal return

Input registers

R3 - code for debug control command: 6 for SNAP, 7 for SNAPC.

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - if error return, contains error code

INPUT

- PLB - displacement in CRPT for buffer containing flag or comment field of command
- CSL - displacement in CRPT for word containing byte length of comment.

OUTPUT

- TSTACK - temp stack used for building SNAP table and for saving registers.

INTERACTION

Modules used:

- CHARROUT - get flag or comment field (CHSTSCAN) and check for expected delimiter (CHARSCAN).
- SUBR - get SNAP location or 'to' or 'from' address (GETLOC2 in GETLOC) and scan to end of command (EOCCSCAN).
- DEBUGR - write debug record (WRTDEBUG).

ERRORS

Any errors detected cause return to (SR4) with the error code in SR3. Normal return is to (SR4) + 1.

UTS TECHNICAL MANUAL

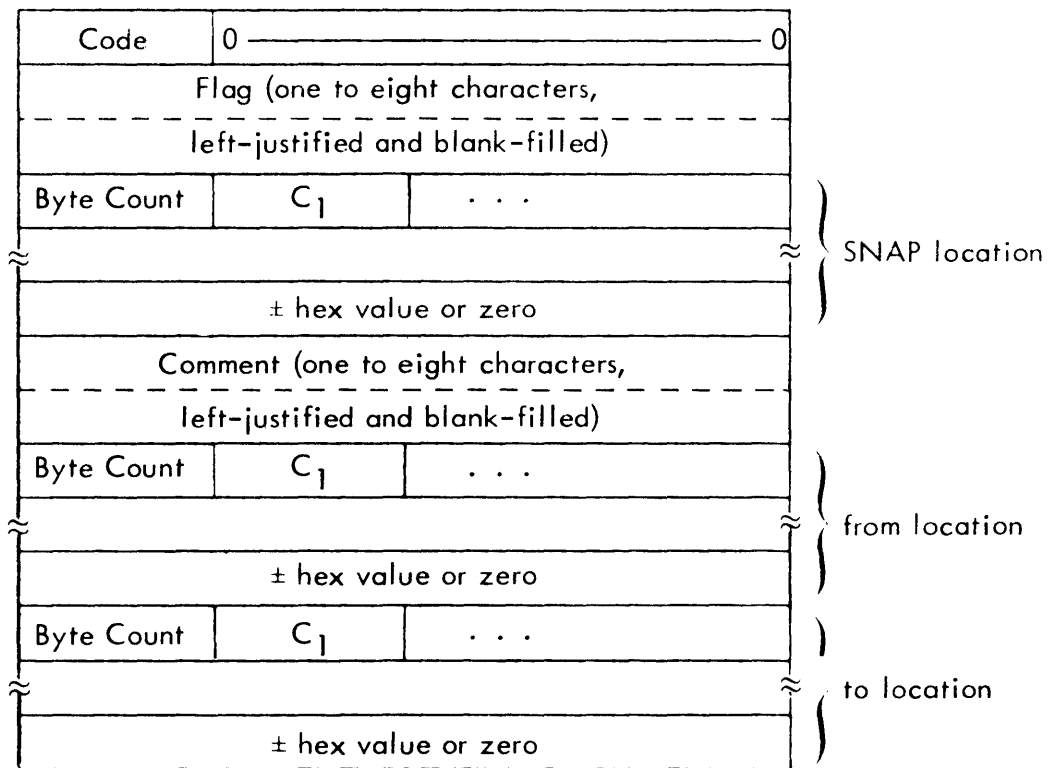
DESCRIPTION

The SNAP (SNAPC) control command to be processed must be in the control command buffer. The top two words of the stack used by CCI must contain the address and the address plus one at which the SNAP table is to be built when the command is processed.

The SNAP (SNAPC) control command is processed and the information is stored in the SNAP table. There is one SNAP table for each 'from - to' specified on the command and each SNAP table is written as one record into the system debug file by subroutine WRTDEBUG. If an error is detected in processing the command, SR3 is loaded with an error code and the error exit is taken.

Format of the SNAP, SNAPC Table

The SNAP, SNAPC Table is written into the system debug file for the current job.



where

- Code is 6 for SNAP command, 7 for SNAPC command
- SNAP location is location at which SNAP is to be done
- from location is beginning dump location
- to location is ending dump location

ID

IFR - Process IF, AND, OR, and COUNT control commands

PURPOSE

To process an IF, AND, OR, or COUNT command and build the appropriate table for the command.

USAGE

Calling sequence

BAL, SR4 IFR

Error return

Normal return

Input registers

R3 - code for debug control command: 8 for IF, 9 for AND, 10 for OR, 11 for COUNT

R5 - address of JIT

R7 - address of CRPT

SR1 - current character or zero

Output register

SR3 - if error return, contains error code

INPUT

CSL - displacement in CRPT for word containing byte length of field.

PLB - displacement in CRPT for buffer containing current field of command.

OUTPUT

FLAGS - displacement in CRPT for flag word. Bit 2 is set to indicate character buffer is full.

TSTACK - temp stack used for building debug tables and for saving registers.

INTERACTION

Modules used:

CHARROUT - get flag from command (CHSTSCAN), check for expected delimiter (CHARSCAN), get decimal value (DECSCAN), and get relationship (NAMSCAN).

SUBR - get location (GETLOC2 in GETLOC), scan to end of command (EOCCSCAN), and get decimal value (GETDECVAL).

DEBUGR - write debug record (WRTDEBUG).

UTS TECHNICAL MANUAL

ERRORS

Any errors detected cause return to (SR4) with the error code in SR3. Normal return is to (SR4) + 1.

DESCRIPTION

The IF (AND, OR, COUNT) control command to be processed must be in the control command buffer. The two top words of the stack used by CCI must contain the address and the address plus one at which the IF, AND, OR, or COUNT table is to be built when the command is processed.

In processing the IF, AND, OR commands, the information is stored in the IF, AND, OR table. The information from the COUNT command is stored in the COUNT table. The processing of the last three fields of the COUNT command is processed by a separate block of code labeled COUNTR since the format of the COUNT command is slightly different from the IF, AND, OR commands. The IF, AND, OR and the COUNT tables are written as separate records into the system debug file by subroutine WRTDEBUG. If an error is detected in processing the command, SR3 is loaded with an error code and the error exit is taken.

Format of IF, AND, OR Table

This table is written into the system debug file for the current job.

| | | | |
|---|----------------|---------------------|------------------|
| Code | | r | |
| Flag(one to eight characters, left-justified and blank filled) | | | |
| Byte Count | C ₁ | ... | } LOC |
| ± hex value or zero | | | |
| Byte Count | C ₁ | ... | } L ₁ |
| * | | ± hex value or zero | |
| | b ₁ | x ₁ | } L ₂ |
| Byte Count | C ₁ | ... | |
| * | | ± hex value or zero | } |
| | b ₂ | x ₂ | |

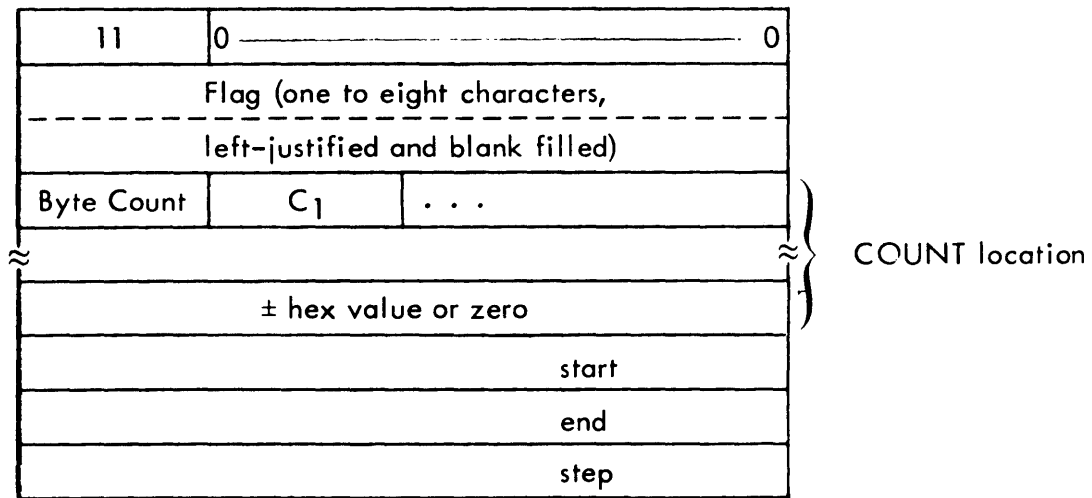
UTS TECHNICAL MANUAL

where

| | |
|---------------------------------|--|
| Code | is 8 for IF command, 9 for AND command, 10 for OR command. |
| r | is 0 for GT, 1 for LT, 2 for EQ, 3 for GE, 4 for LE and 5 for NE. |
| b ₁ , b ₂ | is 0 for byte, 1 for halfword, 2 for word and 3 for doubleword. |
| x ₁ , x ₂ | is 0 through 7. |
| * | specifies an indirect address (for L ₁ or L ₂). |
| LOC | specifies the location at which the test is to take place. |
| L ₁ , L ₂ | specify the locations to be compared as specified by 'r'. |

Format of Count Table

The Count Table is written into the system debug file for the current job.



UTS TECHNICAL MANUAL

ID

WRTDEBUG - Write out Debug Table

PURPOSE

To write the different debug tables into the system debug file SYSIDD.

USAGE

Calling sequence

BAL, SR4 WRTDEBUG

Input registers

- R2 - address of debug table to be written
- R3 - end address plus one of the debug table to be written
- R4 - address of key (segment name in Tree table) with which the debug table is to be written.
- R5 - address of JIT

INPUT

CCLTFLGS - displacement in JIT for word containing flag (bit 15) which if set causes CCI to bypass incrementing counters in the RUN table.

OUTPUT

- ALOCCT - displacement in JIT for address of RUN table. Counters are incremented for MODIFY, SNAP, SNAPC, IF, AND, OR, or COUNT records.
- DBUGF - displacement in JIT for word in which PMD flag (bit 14) is set if a PMD type record is written.

INTERACTION

Monitor services used (handwritten CALs and FPTs):

- M:SETDCB - enter error and abnormal addresses in DCB.
- M:WRITE - write debug table.

DESCRIPTION

The specified debug table is written into the system debug file through the M:X1 DCB and the last byte of the key (segment name) is incremented by one. If the debug command being processed follows the RUN command, the appropriate debug counter in the RUN table is incremented. If the debug command is a PMD, PMDI, PMDE, the PMD flag is set in the JIT. If an error or abnormal condition occurs in trying to write the record into the file, the normal exit is taken and the record is lost.

UTS TECHNICAL MANUAL

ID

READBI - Read the BI device and write the information into the system BI file.

PURPOSE

To process the BI option on a LOAD or OVERLAY command and perform the I/O.

USAGE

Calling sequence

BAL, SR4 READBI

Input register

R5 - address of JIT

Output register

SR3 - if error return, contains error code

INPUT

M:BI - address of DCB through which the BI file is read.

ALOCCT - displacement in JIT for address of LOCCT through which the Tree table address is obtained.

M:C - address of M:C DCB.

OUTPUT

CCBEF - displacement in JIT for word in which buffer full flag is set (bit 8) if BI and C have same assignment.

INTERACTION

Modules used:

OPNSTARF - open file *SYSIDB.

SUBR - get DCB assignment (GETDCBA).

Monitor services used (handwritten CALs and FPTs):

M:OPEN - open M:BI DCB.

M:SETDCB - set error and abnormal addresses in M:BI DCB.

M:READ - read record through M:BI.

M:WRITE - write record through M:X1.

M:CLOSE - close M:BI and M:X1.

ERRORS

If an error occurs in reading, writing or opening, exit is made to (SR4) with the

error code in SR3. Normal exit is to (SR4) + 1.

DESCRIPTION

After a LOAD or OVERLAY control command has been processed, the READBI routine is called if the BI option was specified on the command. The BI device is read through the M:BI DCB. Whatever device the DCB is assigned to is the device read. The BI device is read until two consecutive EODs are read or until an end-of-file is encountered. The information is written into the system BI file. The name of the system BI file is a three-character name consisting of the current job id from the JIT with the letter B appended. If an error is encountered in opening, reading or writing, SR3 is set as shown above and the error exit is taken.

ID

LOGRT - Job initiation and validation routine

PURPOSE

To open the logon file, check resource allocation requirements, store information from the logon record into the JIT and write the Assign-Merge record header.

USAGE

Calling sequence

BAL, SR4 LOGRT

Output register

SR2 - if error, contains the appropriate error code

SR3 - contains 0 if no error or violation has occurred
contains 1 if job account is invalid

contains 2 if resource violation has occurred

INPUT

M:EO - DCB used for accessing the logon file

M:X1 - DCB used for writing the Assign-Merge record

J=JIT + JACCN - two-word entry in the JIT which is set to a secret account for reading the logon file. Original account is then restored.

J=JIT + JUNAME - three-word entry in JIT used to test for manager account's name.

OUTPUT

JB: PRIV - byte item in JIT which is set to contain the privilege level from the logon record.

J: ASSIGN - bit 3 of this word entry in JIT is set if the logon record indicates that files under this account are not to be read by any other account.

PRDCRM - word entry in JIT which is set to contain the permanent RAD storage space as specified in the logon record or the default already established in the JIT.

PRDPRM - permanent DISK storage space, usage analogous to PRDCRM.

TMDCRM - word entry in JIT which is set to contain the temporary RAD storage space as specified in the logon record or the default value already established in the JIT.

TMDPRM - temporary DISK storage space, usage analogous to that of TMDCRM

UTS TECHNICAL MANUAL

- J: JIT + JPUF - word entry in JIT bits 28-31 of which are set to contain the peripheral use flags from the logon record.
- J:XP - word entry in JIT the left halfword of which is used to contain the default retention period (for files in the account) as specified in the logon record and the right halfword the maximum retention period; if not specified in logon record, default values already established in JIT are used.

Modification of the following items in the JIT are possible due to the resource checking process.

- JB: MNPA - byte item in JIT which is set to contain the core size
- JB: M9T - byte item in JIT which is set to contain the number of 9 track drives allocated to this job
- JB: M7T - byte item in JIT which is set to contain the number of 7 track tape drives allocated to this job
- JB: MSP - byte item in JIT which is set to contain the number of disk spindles allocated to this job

INTERACTION

Monitor services used (handwritten CALs and FPTs)

- M: OPEN - open the logon file
- M: READ - read the logon file
- M: CLOSE - close the logon file
- WAMR - write the Assign-Merge record
- M: WRITE - write logon record for management account
a release CAL to release the excess resources.

ERRORS

If logon file cannot be accessed with user's account and name, and a file is not being created for management, SR2 and SR3 are set and an exit is made.

If resource checking results in violation, SR2 and SR3 are set and an exit is made.

DESCRIPTION

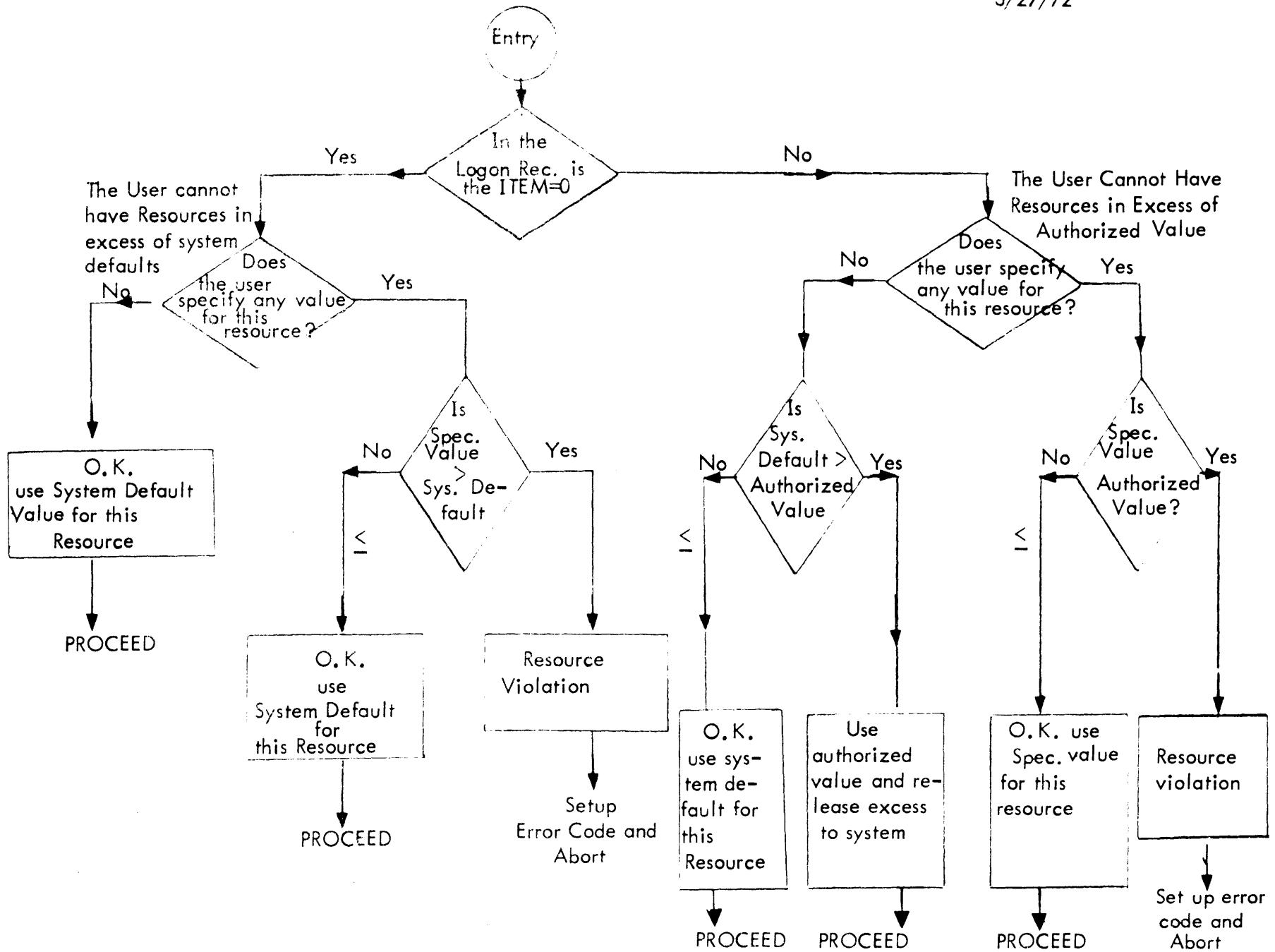
LOGRT is called by JOBR to set up the JIT from the values present in the user's logon record, perform resource checking and write the Assign-Merge header information. The logon file, :USERS, is opened and a keyed record is read. The key for this record is formed from the user's account and name present in the JIT. If the keyed record exists, values are extracted from the record, appropriate resource checking is performed and the values are entered into the JIT. The

Assign-Merge leader information (such as pointers, start date and time, user name, system ID, extended accounting information, rate structure, permanent RAD and DISK storage space) is prepared and the Assign-Merge record is written.

If an error return occurs on either the open or the read, the error exit is taken from LOGRT. If an abnormal return occurs and the file is being updated, another attempt is made to open it. If the file does not exist and a management account is specified, a logon file is created. In either case, the Assign-Merge record header information is written.

Resource checking is performed on four resource items -- core, 9 track tape drive (9T), 7 track tape drive (7T) and disk pack spindle. The job control command streams and gathers information (from RESOURCE, LIMIT and JOB cards) necessary for job selection by the multi-batch scheduler. The scheduling information is stored in the symbiont input tables until job selection. When a job is selected and CCI takes control, system default values are still in the JIT associated with the job and they are saved upon initial entry of a job to CCI. After CCI makes its first access to the input symbiont file, the secondary information regarding core, 9T, 7T and SP are transferred to the user's JIT for control. In the respective bytes for these resource items, a 1 in the high order bit indicates that the user has not specified any request for that particular resource. The system default values, the pre-processed resource information and the user's authorization values in the logon record are then examined to determine if any resource violation results or any excess resources should be released to the system pool.

The flow-chart of the resource checking process is outlined on the following page:



ID

TEL - Terminal Executive Language

OVERVIEW

The Terminal Executive Language processor provides the interface between the terminal user and the UTS system and exerts executive control of all terminal activities. Functionally and operationally, TEL commands are described in the following five categories:

- I. Subsystem calls
 - A. Explicit subsystem calls
 1. Calls that require no intermediate attention from TEL
 2. Calls that require intermediate attention from TEL
 - B. Implicit subsystem calls
 - C. Load module (user program) execution
- II. On-line session checkpoint-and-restart facility
- III. Interrupting, resuming and terminating execution
- IV. Input, Output and Format control
- V. Miscellaneous services
 - A. Account security
 - B. File reliability
 - C. Job status information
 - D. Job cancellation
 - E. Terminal status inquiry
 - F. Terminal type change
 - G. Operator - console communication
 - H. On-line session summary
 - I. System parameters display
 - J. DCB information assignments
 - K. On-line session termination

UTS TECHNICAL MANUAL

PURPOSE

The various TEL commands, each dictating some specific actions, are grouped and outlined under the aforementioned categorization. In command formats, square brackets are used to indicate optional characters. The absence of such optional characters therefore clearly indicates the acceptable abbreviated form of each respective command.

- 1A. Explicit subsystem calls¹
Control is transferred to the subsystem requested without further attention from TEL except perhaps a rudimentary syntax check. The subsystem calls are grouped according to the processors they invoke.

Language Processors:

!BASIC
!FORT4²
!META²
!COBOL

Execution Control Processor/Linkage Processor

!DELTA
!LINK

Service Processors:

!ANLZ
!BATCH
!E[DIT]
!PCL

System Management Processors³

!CONTROL
!DRSP
!ERR:LIST
!ERR:SUM
!FILL
!RATES
!SUMMARY
!SUPER
!UTSPM

¹This is intended to be a general, but in no sense complete, list. For example, calls to application and user processors are not included.

²These two commands may require intermediate attention from TEL.

³Access to most system management processors may require special privileges.

IB. Implicit subsystem calls

The command implies a subsystem action without explicitly requesting the appropriate processor.

| | |
|-----------|--|
| !B[UILD] | implicitly invoking the EDIT subsystem |
| !C[OPY] | implicitly invoking the PCL subsystem |
| !D[ELETE] | implicitly invoking the PCL subsystem |
| !L | implicitly invoking the PCL subsystem |
| !RUN | implicitly invoking the LINK subsystem |

IC. Load Module (user program) execution⁴

The command initiates execution of a load module

!S[TART].....
!lmm.

II. On-line session checkpoint-and-restart facility

During interruptions of execution, core images of programs may be saved on secondary storage for subsequent recall and continuation.

!SAVE
!GET
!RESTORE

III. Commands for interrupting, resuming and terminating execution⁵.

Upon interruption of execution, control is returned to TEL. The interrupted operation may then be resumed by one of the following commands:

!CONTINUE
!GO
!PROCEED

The interrupted operation may be given up completely by the following commands:

!END
!Q[UIT]
!STOP

⁴Intermediate attention from TEL is generally required for these commands.

⁵For a complete list and description of commands that can be used after interruption, refer to the section TEL in the UTS reference manuals.

IV. Commands for input, output and format control

| | |
|--|--|
| <p>ICOMMENT ILIST IOUTPUT IPLATEN IPRINT ITABS IDONT COMMENT IDONT LIST IDONT OUTPUT</p> | <p>specifies the destination of error commentary specifies the destination of listing output specifies the destination of ROM output specifies the width and page length for terminal output causes output for the line printer and card punch to be placed in the queue at once. specifies the simulated tab stop for a terminal. counter acting</p> |
|--|--|

V. Miscellaneous services

Various commands are available under TEL to supplement the integration of the terminal user to the system.

| Functions | Commands |
|-----------------------------------|------------------|
| A. account security | !PASSWORD |
| B. file reliability | !BACKUP |
| C. job status examination | !JOB |
| D. job cancellation | !CANCEL |
| E. terminal status inquiry | !TERMINAL STATUS |
| F. terminal type change | !TERMINAL |
| G. operator-console communication | !M[ESSAGE] |
| H. on-line session summary | !ST[ATUS] |
| I. system parameters display | !DI[SPLAY] |
| J. DCB information assignment | !SET |
| K. on-line session termination | !OFF |
| | !BYE |

USAGE

User oriented information and instruction as to commands, their syntax and parameters may be obtained from the UTS/TS Reference Manual 90 09 07.

INPUT

Input is the command line type by the terminal user. The command line is read into J:TELBUF; the command line record size is in byte zero of J:PUF. Input information is also placed in the JIT by LOGON (system ID, account, password, privilege, etc).

OUTPUT

Output is described in the UTS/TS Reference Manual command descriptions and individual command writeups.

DATA BASES

Data contained in the monitor area which is accessed by TEL includes:
Job Information Table - JIT (Section VA)

Two JIT words which need explanation not to be found in Section VA are JTEFLGS and JOPT. Certain logical states are indicated within TEL by flag bits set in JTEFLGS. The bits have the following associations:

| <u>Bit</u> | <u>Meaning When Set</u> |
|------------|---|
| 11 | Unrecognized load module being processed |
| 12 | Under DELTA is implied for next command |
| 13 | Implied quit flag |
| 14 | Allow no following message text (trailing garbage flag) |
| 15 | Auto-processor association (Set by LOGON) |
| 16 | Auto-get associations (Set by LOGON) |
| 17 | LMN was detected by START command (SYSERR process) |
| 18 | Current buffer image is to be processed |
| 19 | 'DONT' adverb in effect |
| 20 | 'LIST', 'OUTPUT', 'COMMENT' command in progress |
| 21 | Unique 'COMMENT' command indicator |
| 22 | Default for \$ ROM in progress |
| 23 | Load under FDP required |

UTS TECHNICAL MANUAL

| <u>Bit</u> | <u>Meaning When Set</u> |
|------------|---|
| 24 | Load under DELTA required |
| 25 | 'OVER' specified as command modifier |
| 26 | 'ON' specified as command modifier |
| 27 | SI has been assigned - chain remaining FIDs |
| 28 | Do not release SI entries in A/M table |
| 29 | Parse phase flag, 0 = Build GO DCB; 1 = Build LO DCB |
| 30 | Y ^c (Break) received (set by monitor) |
| 31 | Job Step flag, 0 = not at job step; 1 = within job step |

TEL and certain system processors utilize a cell in JIT to communicate a subset of available options. TEL sets a defined bit prior to the load of a processor and this indicates an assignment has been made for that option (DCB). The bit definitions for the cell 'JOPT' are as follows:

| <u>Bit</u> | <u>DCB</u> | <u>Bit</u> | <u>DCB</u> |
|------------|------------|------------|------------|
| 23 | DO | 28 | CI |
| 24 | GO | 29 | CO |
| 25 | SO | 30 | BO |
| 26 | LL | 31 | LO |
| 27 | SI | | |

In addition, the GO, LO, and DO bits may be set and reset during the course of a job execution, by the receipt of the OUTPUT, LIST and COMMENT (optionally preceded by adverb 'DONT') command verbs.

Data Control Blocks

TEL communicates through the DCBs represented in the JIT:

- a. M:UC - *DW:BUF *DW:ERA *DW:ABA
*DW:PRI *DW:BLK
- b. M:XX - *DW:ERA *DW:ABA *DW:FCD

Fields which are modified are preceded by an asterisk.

Assign/Merge Table (Section VL)

Information merged into DCB's prior to I/O operations is kept in the A/M table. This table is created for each user by LOGON. Entries may be added and deleted and default entries may be altered by the SET command.

SUBROUTINES

TEL subroutines are described individually in Section PB.02. Each is mentioned by name in the TEL table of contents.

ERRORS

The error messages that TEL issues are:

1. 'ASSIGN LIMIT EXCEEDED' issued by subroutine ASSIGN when there is not enough room in an A/M entry for another parameter. No more assignments allowed.
2. 'UNABLE TO READ A/M TABLE' issued by subroutine GET when no virtual pages are available; by READAM if a read error is detected when trying to bring A/M table in or if no buffer is available. No user correction available.
3. 'INPUT ERROR - RETRY' issued by main routine PROMPT section if a read error is detected while reading the user terminal. Reenter command.
4. 'ON FILE' issued by main routine when an error in using ON as a modifier is detected. Used in conjunction with LEGER to construct a message reading ON FILE ILLEGAL. Enter correct command.
5. 'FILE: ME' issued by main routine when ME is specified illegally for output. Used to construct a message reading 'FILE: ME ILLEGAL'. Enter correct command.
6. 'ILLEGAL' issued in conjunction with various other error messages.
7. 'NOT FOUND' issued in conjunction with other error messages.
8. 'START WHAT' issued by main routine when an attempt is made to START a non-existent load module. Enter correct LMN.
9. 'IMPROPER FORMAT FOR SET CMD' issued in the SET command section upon detection of data after termination should occur, or for an illogical command sequence. Correct syntax error and reenter command.
10. 'CONTINUE WHAT' issued by continue section when the CONTINUE command has been given at job step. No correction needed.
11. 'COMMAND LEGAL AT JOB STEP ONLY' issued by SET if an attempt is made to use the command out of job step. Enter QUIT command to force job step.
12. 'BAD PLIST - RESPECIFY DCB' issued by EXPAND when an error in PLIST structure is detected. Reenter SET command.
13. 'BATCH WHAT' issued by BATCH when there is no FID associated with the command.
14. 'CANNOT ACCESS THE FILE' used by the PASSWORD when it cannot open the :USERS file, when there is an error reading the user's logon record, or when there is an error in writing the logon record. Reenter command.
15. 'FILE DOES NOT EXIST' issued by BACKUP when trying to BACKUP

- a file which does not exist.
16. 'OPTION ILLEGAL FOR DEVICES' issued by main routine when using an incorrect option for device PLIST. Reenter command with legal device option.
 17. 'OPTION ILLEGAL FOR FILE' issued by main routine when using an improper option for a file PLIST. Reenter command with legal file option.
 18. 'QUIT WHAT' issued when a QUIT command is given when there is no break condition in progress. No correction needed.
 19. 'PROCESS NOW ACTIVE: QUIT OR CONTINUE' issued when a command other than QUIT or CONTINUE is given at break time. Enter QUIT or CONTINUE command.
 20. 'ILLEGAL REMOTE WORK STATION ID' issued when remote work station ID on PRINT command exceeds characters.
 21. 'INSUFFICIENT ASSIGN/MERGE ENTRY SIZE' issued by PENT when a parameter won't fit into the A/M table and by FILENT when there is insufficient room for an entry in the table. No correction available.
 22. 'DCB NOT ASSIGNED' issued by SET when there is an attempt made to update a specification for a DCB which is not present in the A/M table. Assign DCB with a SET command.
 23. 'TERMINAL TYPE NOT VALID' issued when processing the TERMINAL command if illegal terminal type is input. Reenter command with recognized terminal type.
 24. 'GET WHAT' issued by GET when there is no FID associated with the command. Reenter command with correct FID.
 25. 'SAVE WHAT' issued by SAVE when an attempt is made to use the command at a job step or when no FID is given in conjunction with the command. Reenter command with correct FID or enter QUIT command.
 26. 'CONFLICT WITH DELTA - TRY LATER' issued by GET when the M:XX DCB is already open which indicates that DELTA is using it. Wait and reenter command.
 27. 'IMPROPER FILE' issued by GET when some file characteristic prevents its being used in the context of an on-line environment, i. e., it probably was not SAVED first.
 28. 'BAD DCB' issued in conjunction with IMPROPER FILE message when the DCB being used is not correct. No user correction available.
 29. 'BAD JIT' issued in conjunction with 'IMPROPER FILE' message when the JIT is not correct. No user correction available.
 30. 'EXPANDED INPUT EXCEEDS 80 CHARACTERS' issued when processing a short form command (E, B) which exceeded 80 characters when expanded.

UTS TECHNICAL MANUAL

31. 'BAD LIMITS' issued in conjunction with 'IMPROPER FILE' message when the LIMITS are not correct. No user correction available.
32. 'CORE EXCEEDED' used by TEL when no pages are available for temporary storage. No user correction available.

INTERACTIONS

If a TEL command results in a subsystem call, that request is made to STEP in the form of an interpretive exit. CAL1,9 1 with the following register settings:

| <u>Registers</u> | <u>Contents</u> |
|------------------|--------------------------------------|
| 6, 7, 8 | Load module name in TEXTC format |
| D2, D3 | Account number |
| SR3, SR4 | Password |
| 0, 1 | 'UNDER' name, i. e., DELTA, FDP or 0 |

An interrupted job may be aborted by entering a QUIT, STOP or END command which results in an abortive exit: CAL1,9 3.

An interrupted job may be continued by entering a CONTINUE or PROCEED command which results in a restorative exit: CAL1,9 1, R6 = 0.

A monitor detected error in a user program results in a return to TEL with an abort CODE/SUBCODE in J:ABC and J:ERO. This code is used to read the error message file (ERRMSG) and output the corresponding error message. If the error was on an ASSIGN/MERGE read or write, TEL does an interpretive exit to LOGON after printing the error message.

DESCRIPTION

TEL consists of 2 modules: TEL and ERRMSG. Both modules are entirely reentrant pure procedure (CSECT1) and are loaded in the special shared processor area (bias: X'1C200'). Special JIT access is provided so that TEL may have read/write access to JIT.

Initial entry to TEL is from STEP subsequent to the logon procedure (exception: an auto-call processor is specified). Reentries result when subsystems issue normal, abort or error CALs or if the user depresses Y^C or the break key.

Upon entry into TEL, the environment is initialized and dynamic storage is requested from the index buffer pool (J:IPOOL). If none is available, the user is informed and an abort CAL is issued. Otherwise, the abort flag, J:ABC is checked. If set, the system error message file is read and the error message is sent to the user. If reset, the job step flag is set, a prompt character (!) is typed and a terminal read is issued. The command is read into J:TELBUF and the command record size is in byte zero of J:PUF.

After the read, the command line is scanned for the command field.

If the scan is terminated by a period, the command is assumed to be a load module name in the user's account. This results in the appropriate interpretive exit. Otherwise the command field is compared to a table of command verbs recognized by TEL. The table entry number is used as an index to a command vector. If further processing is required control is transferred to the appropriate routine. If not, the processor name is returned and TEL executes an interpretive exit. Subsequent processing is described in the individual command writeups.

TEL commands

For ease of reference, TEL commands are listed in alphabetical order. Obvious subsystem calls, however, are not described. These commands signal a request for a standard system processor and TEL, as a result, performs an interpretive exit which invokes the correct UTS monitor action. Such commands include !BASIC, !DELTA, !LINK, !ANLZ, !BATCH, !E[DIT], !PCL, !CONTROL, !DRSP, !ERR:LIST, !ERR:SUM, !FILL, !RATES, !SUMMARY, !SUPER and !UTSPM.

Generally, any information following such a command on the same command line, if meaningful, has to conform to the format of usage as established by various processors. TEL, therefore, is relieved of such a responsibility except in the case of a !FORT4 or !META command since some intermediate action is required before making the interpretive exit call. (These two commands are described in this section).

UTS TECHNICAL MANUAL

ID

!BACKUP - save a file on the system backup tape

DATA BASES

J:TELBUF
J:JIT+JACCN
M:XX
:BACKUP file
ERRMSG file

INTERACTIONS

| | | |
|---------|----------|--|
| CAL1, 8 | GETPG | Get one dynamic page as a buffer for BACKUP record |
| CAL1, 1 | OPENBKUP | Open :BACKUP file |
| CAL1, 1 | READBKUP | Read BACKUP record |
| CAL1, 1 | WRITERC | Write BACKUP record |
| CAL1, 1 | CLOSEBC | Close :BACKUP file |
| CAL1, 8 | PGDROP | Free one dynamic page |
| CAL1, 6 | BKUPCAL | Wake the FILL ghost |
| CAL1, 1 | WRITE | Error notification |

SUBROUTINES

SCAN obtain FID elements
WRITERR obtain error message from system error message file

ERRORS

A dynamic page is not available. The Backup record is larger than one page.

DESCRIPTION

The system account is temporarily inserted in the users JIT and the :BACKUP file is opened. The BACKUP record is read, altered to include file name, account, password and written back to the :BACKUP file.

Finally the :BACKUP file is closed, an initiate job CAL is issued to initiate or wake up the Backup ghost.

ID

IB[UILD] - create a keyed file on secondary storage

DESCRIPTION

An interpretive exit to the EDIT processor is executed. The whole command is retained in the J:CCBUF buffer in JIT.

ID

IBYE - terminate a terminal session

DESCRIPTION

Same as the !OFF command. An interpretive exit to the LOGOFF processor is executed.

ID

ICANCEL - delete a job waiting in the input symbiont

INTERACTIONS

| | | |
|---------|----------|--|
| CAL1, 1 | CANCL | a JOBENT CAL to delete a job |
| CAL1, 2 | SENDCNCL | send message 'ID--- CANCELED' if job is successfully deleted |
| CAL1, 1 | WRITE | issue message to the user |

SUBROUTINES

| | |
|---------|-------------------------------------|
| SCAN | obtain JOB ID following the command |
| HEX2BIN | Convert EBCDZC to binary |

ERRORS

Attempting to delete job submitted under accounts other than the LOG-IN account. Message 'NOT YOUR FILE' is issued.

The job has either been completed or not yet been input. Message 'COMPLETED OR NOT INPUT' is issued.

DESCRIPTION

The job ID (i. e., system ID) specified in the command argument field is converted to binary form and a JOBENT CAL is issued to delete the job. If the job is successfully deleted, a message is issued to the operator console. If the job is currently running, it is aborted. Otherwise, error return is taken and appropriate message is issued to the user. In all cases, control is returned to the user via TEL.

UTS TECHNICAL MANUALID

ICOMMENT - direct the output of the M:DO DCB

DATA BASES

J:TEFLGS

J:JIT + JOPT

A/M table

SUBROUTINES

SCAN obtain modifier, device mnemonic, or FID

READAM Read A/M table

ERRORS

User not at job step. Modifier other than 'ON' or 'OVER'. Illegal device mnemonic.

DESCRIPTION

The M:DO DCB is assigned either to a file, the line printer or the terminal as specified by arguments on the command line.

Set bits 21 and 29 in J:TEFLGS; set bit 23 in J:OPT

ID

ICONTINUE - continue an interrupted process

DESCRIPTION

TEL determines if the user is at job step (bit 31, J:TEFLGS). If so, an exit CAL to the monitor is issued with R6 = 0. Otherwise an error message is sent.

Equivalent to the !GO and !PROCEED command.

ID

IC[OPY] - copy a file or device input to the specified file or device

DESCRIPTION

An interpretive exit to the PCL processor is executed.

UTS TECHNICAL MANUAL

ID

!D[ELETE] - delete the specified files

DESCRIPTION

An interpretive exit to the PCL processor is executed.

ID

!DI[SPLAY] - display system load parameters

PURPOSE

To list certain system parameters that are useful to the user. The parameters displayed are: number of users, execution multiplication factor, 90 percent response time and permanent secondary storage space available.

DATA BASES

J:JIT+PRDCRM
J:JIT+PRDPRM
J:DWSK

INTERACTIONS

CAL1, 8 DISFPT get display parameters from the monitor
CAL1, 1 Write output to the terminal

SUBROUTINES

BINDECBCD convert binary value to EBCDIC form

DESCRIPTION

An M:DISPLAY CAL (CAL1, 8) is issued to the monitor which returns execution multiplication factor, 90 percent response time and number of users in registers 5, 6 and 7 respectively. Permanent secondary storage space available is obtained from the JIT (J:JIT+PRDCRM, J:JIT+PRDPRM); the value displayed is the sum of permanent RAD and PACK space available to the user.

ID

DONT { COMMENT
LIST
OUTPUT

PURPOSE

To suspend output from the DCB implied by the command.

UTS TECHNICAL MANUAL

DESCRIPTION

Set bit 19 in J:TELFLG and proceed according to the respective command verb.

ID

!E[DIT] - calls EDIT processor

DESCRIPTION

An interpretive exit to the EDIT processor is executed.

ID

!END - to abort a currently active process to allow the initialization of a new process.

DATA BASES

J:IPOOL

DESCRIPTION

A check is made to determine if the user is at job step. If so, an inquiry message 'QUIT WHAT' is sent to user. Otherwise, the user's dynamic storage is returned to the index buffer pool (J=IPOOL) and an abort CAL is issued.

ID

!FORT4 - interface with FORTRAN IV processor

DATA BASES

Assign - Merge table

SUBROUTINES

| | |
|----------|--------------------------|
| READAM | read Assign-Merge table |
| GETFIELD | special entry to SCAN |
| FILENT | create short PList entry |

DESCRIPTION

The command argument field is scanned for source input, binary output and listing output specifications. Appropriate Assign-Merge table entries are made if default or current assignments are to be changed. An interpretive exit to the FORTRAN IV processor is then executed.

UTS TECHNICAL MANUAL

ID

IGET - restore user's environment

PURPOSE

To restore the users environment that was previously checkpointed by a SAVE command.

DATA BASES

J:JIT+TELFLGS

M:XX

M:FPOOL

M:BUF

J:EUP

J:PUL

J:JIT+JB:CMAP

J:JAC

J:CUL

J:PUL

J:CPPO

J:ASSIGN

J:JIT+JB:LMAP

J:JRNST

J:CLMN

J:CLMP

J:GST

J:IST

J:GST

J:IST

J:CPROCS

J:CFLAGS

J:INTENT

J:USENT

J:EXTENT

J:VLH

J:PLL

J:PUL

J:DDUL

J:DLL

JB:TDP

J:DDLL

SUBROUTINES

FLOP Open file

UTS TECHNICAL MANUALINTERACTIONS

| | | |
|---------|--------------|-------------------------------|
| CAL1, 8 | =X'84000001' | Get virtual page |
| CAL1, 1 | 'READAD | Read save record |
| CAL1, 1 | TYPE | Messages |
| CAL1, 8 | SMPRTPP | Set memory protection |
| CAL1, 8 | =X'08000002' | Free pages |
| CAL1, 8 | 0 | Set memory protection to read |

ERRORS

FID specified does not exist or no FID specified.

No virtual page available.

User not at job step.

File is not a SAVEd file.

DESCRIPTION

First a check is made to determine if the user is at job step. If not, an error message is sent. Next, an attempt is made to open the file named in the command line. If the file does not exist or cannot be accessed, an error message is sent. Otherwise the following environment is restored to the user: Context data, dynamic pages, common pages, DCBs, processor and flags. Reference SAVE command.

ID

IGO - continue an interrupted process

DESCRIPTION

Equivalent to commands !CONTINUE and !PROCEED.

See description for the !CONTINUE command.

UTS TECHNICAL MANUALID

!JOB - check status of batch jobs

INTERACTIONS

| | | |
|---------|--------|----------------------|
| CAL1, 1 | JOBCAL | check job status |
| CAL1, 1 | WRITE | write status message |

SUBROUTINES

| | |
|-----------|--------------------------|
| SCAN | Obtain job ID |
| HEX2BIN | Convert EBCDIC to binary |
| BINDECBCB | Convert binary to EBCDIC |

DESCRIPTION

The job ID (i. e. system ID) specified in the command argument field is converted to binary and a M:JOB CAL is issued. The code returned is used to output the appropriate status message.

ID

!L - list file names and optionally, attributes from the account dictionary, tape or disc pack

DESCRIPTION

An interpretive exit to the PCL processor is executed.

ID

!LIST - direct the output of the M:LO DCB.

DATA BASES

J:TELFLGS
J:JIT+JOPT
A/M table

SUBROUTINES

| | |
|--------|--|
| SCAN | obtain modifier, device mnemonic, or FID |
| READAM | read A/M table |

UTS TECHNICAL MANUAL

ERRORS

User not at job step.
Modifier other than 'ON' or 'OVER'.
Illegal device mnemonic.

DESCRIPTION

Set bit 29 in J:TELFLGS; set bit 31 in J:OPT.

UTS TECHNICAL MANUAL

ID

!lmn - initiates execution of a load module

DESCRIPTION

The command argument field is scanned for account name and password for the load module. If specified, registers 13 & 14 are set up to contain the account number and register 10 & 11 the password. Further scanning is done for possible source input, binary output and listing output specifications. Appropriate Assign-Merge table entries are then made. An interpretive exit with the load module name in TEXTC format in registers 6, 7 and 8 is then executed.

ID

!M[ESSAGE] - send a message to the operator console.

DATA BASES

J:TELBUF

INTERACTIONS

CAL1,2 SENDMSG send message to OC device

ERRORS

If message exceeds 50 characters, the truncated message is sent.

DESCRIPTION

Two TAB characters are inserted into the message specified in the command argument field and the message is written to the OC devices.

ID

!META - interface with Metasymbol processor

DATA BASES

A/M table

SUBROUTINES

| | |
|----------|--------------------------|
| READAM | read A/M table |
| GETFIELD | special entry to SCAN |
| FILENT | create short Plist entry |

UTS TECHNICAL MANUAL

DESCRIPTION

The command argument fields are scanned for optional source input, binary output and listing output specification. Appropriate A/M table entries are made if default or current assignments are to be changed. An interpretive exit to the Metasymbol processor is executed.

ID

!OFF - terminate a terminal session

DESCRIPTION

An interpretive exit to the LOGOFF processor is executed.

ID

!OUTPUT - direct the output of the M:GO DCB.

DATA BASES

J:TELFLGS
J:JIT+JOPT
A/M table

SUBROUTINES

SCAN obtain modifier, device mnemonic, or FID
READAM read A/M table

ERRORS

User not at job step.
Modifier other than 'ON' or 'OVER'.
Illegal device mnemonic.

DESCRIPTION

Set bit 24 in J:OPT

UTS TECHNICAL MANUALID

!PASSWORD - password a user's account

DATA BASES

J:DWSK
J:JIT+JACCN
J:JIT+JUNAME
M:XX

INTERACTION

| | | |
|---------|-------|---------------------------|
| CAL1, 1 | OPLOG | Open :USERS file |
| CAL1, 1 | *R1 | Read user record |
| CAL1, 1 | *R1 | Write updated user record |

SUBROUTINES

| | |
|--------|------------------------------|
| SCAN | Obtain new password |
| CONCAT | Concatenate name and account |
| SETALC | Set and close file |

DESCRIPTION

The command field is checked to determine if a password is present. If not, password removal is assumed. The :USERS file is opened and the users name and account are concatenated to form a key to read the user's record. At that point, the specified password is inserted or the current password is removed. The record is written and the :USERS file is closed with SAVE option.

ID

PLATEN - define the terminal output page size

DATA BASES

JB:PCW

JB:LPP

SUBROUTINES

SCAN obtain platen length and width

DECBIN convert EBCDIC to binary

DESCRIPTION

The command argument field is scanned for platen width and/or platen length. The width value is placed in JB:PCW and the length value is placed in JB:LPP.

ID

!PRINT - send output to be placed in the symbiont output queue

DESCRIPTION

If no command argument is present, print output will be sent to the line printer and punch output to the punch, both located at the central cite. If a work station name is present as the argument field and valid, the output is assigned to the specified work station by calling RBLOG (PB.02). If not valid, an error message 'QUIT' is issued and the command is essentially ignored.

ID

!PROCEED - continue an interrupted process

DESCRIPTION

Same as !CONTINUE and !GO commands
See description for the !CONTINUE command.

UTS TECHNICAL MANUAL

ID

!Q[UIT]- terminate the current job step

DESCRIPTION

Same as the !END and !STOP commands.
See descriptions for the !END commands.

ID

!R[ESET] - reset all DCB assignments to their system default value.

DATA BASES

J:ABUF
J:TELFLGS
J:JIT+JSTDOPT
J:JIT+JOPT

ERRORS

Operation allowed at job step time only. If not at job step error message 'COMMAND LEGAL AT JOB STEP ONLY' is issued.

SUBROUTINES

READAM read assign-merge table
WRITAM write assign-merge table

DESCRIPTION

The assign-merge table is read. The assigned and SI chains in the assign-merge table are linked to the available chain thoroughly removing all current DCB assignments. Standard options bits from J:JIT+JSTDOPT are then stored into J:JIT+JOPT (i. e., non-standard options no longer effective).

UTS TECHNICAL MANUAL

ID

!RESTORE - restore the previously saved core image.

DESCRIPTION

Same as the !GET command
See description for the GET command

ID

!RUN - link one or more ROMs and initiate program execution

DESCRIPTION

An interpretive exit to the LINK processor is executed.

UTS TECHNICAL MANUAL

ID

!SAVE - to check-point and save a user's environment for later use

DATA BASES

J:JIT+TELFLGS
M:XX
J:FPOOL
J:BUF
J:EUP
J:PUL
J:JIT+JB:CMAP
J:JAC
J:CUL
J:PUL
J:CPPO
J:ASSIGN
J:JIT+JB:LMAP
J:JRNST
J:CLMN
J:CLMP
J:GST
J:IST
J:CPROCS
J:CFLAGS
J:INTENT
J:USENT
J:EXTENT
J:VLH
J:PLL
J:PUL
J:DDUL
J:DLL
JB:TDP
J:DDLL

INTERACTIONS

| | | |
|---------|---------------|--|
| CAL1, 4 | = X'02000000' | Put associated processor and flag into JIT |
| CAL1, 9 | 6 | Super-close |
| CAL1, 1 | WRITERAD | Write to RAD |
| CAL1, 1 | WRITE | Write error messages |

UTS TECHNICAL MANUALSUBROUTINES

| | |
|----------|--------------------------------|
| GETFIELD | Special entry to SCAN |
| SETALL | Close any files currently open |
| FLOP | Open file to save environment |
| FIDER | Break complex FID |
| TRUNK | Truncate DCBs |

DESCRIPTION

First, a check is made to determine if the user is at job step. If so, an error message is sent. Otherwise the following environment is saved as a RAD file: context data, dynamic pages, common pages, program, DCBs, processor and flags. Values and limits are found in the user's JIT, which contains a dynamic profile of the user's environment.

UTS TECHNICAL MANUALID

!SET - create or extend A/M table entries

DATA BASES

J:TELFLGS

J:ABUF

A/M table

SUBROUTINES

SCAN obtain DCB name and information
CHKDCBN check for legal DCB name
NFND convert to TEXTC
READAM read A/M table
ASSIGN construct a new A/M entry
REBIT reset option bit when a DCB is released
WRITEAM write A/M table
TRUNDLE compact skeletal Plist
PENT merge pre-formed parameter words and flags in skeletal Plist
BITS turn off file extension bits
EXPAND expand a compacted A/M entry
SHIFTY determine value for presence bit

DESCRIPTION

First, a check is made to determine if the user is at job step. If not, an error message is sent. Next the A/M table is read into core and the command argument is scanned for the DCB name. If the scan is terminated by a semicolon, the command is assumed to be an update request and the command line is further scanned to determine the updates specified. The updates are merged into the appropriate Plist entries and the A/M table is written to RAD. If the scan is terminated by a comma or blank, the appropriate Plist is constructed and entered in the A/M table. The A/M is written to RAD. If the scan was terminated by a blank and no further arguments are present, the DCB entry is released.

ID

!S[TART]- initiate the execution of a load module

DESCRIPTION

The first argument field of the command is assumed to be the load module name and is converted to TEXTC format for an interpretive exit. The command is further examined for an UNDER debugger. If present, the debug processor name is placed in registers zero and one. Finally an interpretive exit is executed.

UTS TECHNICAL MANUALID

!S[TATUS]- list user status

PURPOSE

To list user status, CPU time, connect time, number of interactions and charge units.

DATA BASES

:RATE file
 J:UTIME
 J:PTIME
 J:TUIOT
 J:TPIOT
 J:INTER
 J:CALCNT
 J:NSUT
 J:NSVT
 J:CIC
 J:CPO
 J:CPPO
 J:CUPO
 J:CDPO
 J:TIME
 J:RATE

INTERACTIONS

| | | |
|---------|-------|----------------------|
| CAL1, 8 | TIMER | get current time |
| CAL1, 1 | ORATE | open :RATE file |
| CAL1, 1 | RRATE | read :RATE file |
| CAL1, 1 | CRATE | close :RATE file |
| CAL1, 1 | TYPE | write output message |

SUBROUTINES

| | |
|----------|------------------------------------|
| TIMEVERT | Convert time to minutes from 12:00 |
| BINDCB | Convert binary to EBCDIC |

DESCRIPTION

Charge units are computed from execution time (J:UTIME), overhead time (J:UTIME+1), processor overhead time (J:PTIME+1), processor execution time (J:PTIME), user core-time factor (J:TUIOT), processor core-time factor (J:TPIOT), console interactions

UTS TECHNICAL MANUAL

(J:INTER), elapsed time, number of I/O CALs (J:CALCNT), number of scratch tapes (JB:TMTS), number of 7T tapes (JB:M7T), number of 9T tapes (JB:M9T), number of packs mounted (JB:PMTS), number of spindles allocated (JB:MSP), number of cards read (J:CIC), number of cards punched (J:COP), number of processor pages out (J:CPPO), number of user pages out (J:CUPO) and the number of diagnostic pages out (J:CDPO). Elapsed time is computed by subtracting current time from connect time (J:TIME). CPU time is computed from J:UTIME and J:PTIME. Terminal interaction are computed from J:INTER. The user's charge class is taken from J:RATE and the :RATE file in opened, read and closed. The charge class number serves as an index into the rate table from which the charge units are computed. The computed values are converted to EBCDIC and printed.

ID

!STOP - terminates the current job step

DESCRIPTION

Same as the END and QUIT commands.
See description for the END command.

UTS TECHNICAL MANUALID

!TABS

PURPOSE

To enter simulated tab stops.

DESCRIPTION

A Plist is constructed with tab stop entries as specified in the command argument fields and an M:DEVICE CAL (UC device) is issued.

ID

!TERMINAL - specify user terminal type or get terminal operational status

DESCRIPTION

The terminal type specified in the command argument field is compared against a table of recognized terminal types. If terminal type is not found, an error message is sent. Otherwise a M:CT CAL is issued. If STATUS is specified in the command argument field, a M:CT CAL is issued and the operational status of the terminal is listed.

ID

!U - implies 'UNDER DELTA' for the next command

DATA BASES

J:TELFLGS

DESCRIPTION

BIT 12 in J:TELFLGS is SET.

UTS · TECHNICAL MANUALID

ABNRET - Abnormal errors reading terminal

PURPOSE

To handle the occurrence of an abnormal or error condition incurred during the reading of a user terminal.

USAGE

Input register

SR3 = ERROR code

B ABNRET

EXIT

PROMPT

REGISTERS ON EXIT

R1, R2, SR3 = DESTROYED

DESCRIPTION

An '05' error code (end-of-data) is ignored and a return is made to PROMPT. Other errors result in the output of an error message and a return to prompt.

UTS TECHNICAL MANUAL

ID

ASSIGN - Assign/Merge Table manipulator

PURPOSE

To provide a central subroutine which performs all of the manipulations of the A/M table.

USAGE

Input registers:

R6 - R7 = DCB name in TEXTC format

R4 = address of A/M table

BAL, SR4 ASSIGN

Creates a new entry position for the given DCB and releases all other like - DCB entries. R5 returns the address of the specified entry.

BAL, SR4 ASSIGN + 1

Locates the current entry for the given DCB and returns the address in R5.

BAL, SR4 ASSIGN + 2

Chains the given DCB to the unique SI DCB chain.

BAL, SR4 ASSIGN + 3

Releases the matching entry (if present) and returns the space to the available chain. The in-use chain is re-linked around the removed entry.

ERROR EXIT

B SYN2

If there is no room in the A/M table for the new entry, the error message 'ASSIGN LIMIT EXCEEDED' is printed and this error exit taken.

UTS TECHNICAL MANUAL

ID

BITS - reset file extension bits

PURPOSE

To turn off file extend bits and turn on DCB assigned bits for a DCB.

USAGE

R4 = A/M entry address
BAL, SR4 BITS

REGISTERS ON EXIT

R5, D1, R0, R7, SR2 = Destroyed
Others as upon entry

DESCRIPTION

A check is made on the DCB type. If the DCB is not a system label, no action is taken. For system DCBs, the file extend bits are reset if necessary.

UTS TECHNICAL MANUAL

ID

CHKDCBN - Validate DCB name

PURPOSE

To test the validity of a DCB name.

USAGE

Input registers

D1 - D2 = DCB name
R7 = Character count
BAL, SR4 CHKDCBN

ERROR EXIT

B SYNTAX

If there are more than seven characters in the DCB name or the name does not start with M: or F:, or if the name is blanks, UC, OC, or XX then this exit is taken.

REGISTERS ON EXIT

R8 - R6 Same as on entry
R7 Destroyed

UTS TECHNICAL MANUAL

ID

DECBIN

PURPOSE

To convert an EBCDIC decimal character string to binary.

USAGE

Input registers:

R7 = Number of characters

R3 = Word address of first character

BAL, SR4 DECBIN

ERROR EXIT

B SYNTAX

Numbers too large to be representable in 32 bits cause this exit to be taken.

REGISTERS ON EXIT

R7 = Result

R8 - R6 = Same as upon entry

UTS TECHNICAL MANUALID

EXPAND

PURPOSE

To expand a compressed A/M entry and put it in the normal skeletal form.

USAGE

Input registers:

R4 = Address of entry to be expanded
BAL, SR4 EXPAND

ERROR EXIT

B SYN1

Any error detected in PLIST format results in printing an error message which states that the PLIST is bad, and this exit is taken.

REGISTERS ON EXIT

Same as upon entry.

DESCRIPTION

This subroutine is called when it is necessary to insert new entries into a compressed PLIST or if current entries must be altered. The method used is to move the entire PLIST to the bottom of the entry area and move parameters upward one word at a time. Words of zero are added where necessary to pad the parameter out to the full space allocated for a skeletal list.

SUBROUTINES

SHIFTY - Used to determine the values of a presence bit and return its relative position within a TEL skeletal PLIST. This is done by a repeated left shift and counting the number of shifts required to bring a one into position zero of the word.

UTS TECHNICAL MANUAL

ID

FID

PURPOSE

To obtain a complex File Identification

USAGE

Input registers:

D1 - D3 = File name

R2 = Field starting position (as for SCAN)

R1 = Remaining Size of Input Message (as for SCAN)

BAL, SR4 FID

ERROR EXIT

B SYNTAX

Presence of a \$ for file name causes an exit to SYNTAX

REGISTERS ON EXIT

D1 - D3 = File name

SR2 - SR3 = Account

R7 - SR1 = Password

R6 = Destroyed

SUBROUTINES

SCAN - Used to obtain account and password from JIT.

DESCRIPTION

FID sets SR2 - SR3 to the proper account and R7 - SR1 to contain the password. If no account is given in the command line, the account from the JIT (user's account) is used; if an account is given then it is used. If a password is not given on the command line then zeroes are inserted into R7 - SR1; if a password is given then it is used.

ID

FILENT

PURPOSE

To create a 'standard' short form Plist in the A/M table.

USAGE

Input registers:

R5 = Assign/Merge table entry

D1 - D3 = File Name

SR2 - SR3 = Account

R7 - SR1 = Password

BAL, SR4 FILENT

REGISTERS ON EXIT

Same as upon entry

DESCRIPTION

The data area of TEL contains a 'canned' PLIST at the location symbolized by that name. The first five words of this PLIST are fixed and are moved to the A/M table entry, then the proper account, password and TEXTC name are moved to the A/M table. The entry created is always specified as an OUT mode entry.

SUBROUTINES

NFND

FIELDS MOIFIED

A/M Table PLIST entry.

ID

FLOP

PURPOSE

To provide a general PLIST to an input file using M:XX DCB.

USAGE

Input registers:

D1 - D3 = File name

SR2 - SR3 = Account

R7 - SR1 = Password or Zeroes

BAL, SR4 FLOP

ERROR EXIT

In the case where there is an error when an Open CAL is attempted, R0 contains the error codes right justified. If there is no error, R0 equals zero.

REGISTERS ON EXIT

Same as upon entry.

DESCRIPTION

This routine builds a general PLIST for an input file using the M:XX DCB. The DCB is then opened specifying consecutive and sequential access, and the name, account and password are inserted into the FPT. If the command line specified ON, the mode is IN; if not, the mode is OUT.

SUBROUTINES

NFND

UTS TECHNICAL MANUALID

NAME\$

PURPOSE

To create a unique 4 character file name for \$ files.

USAGE

Input registers:

R5 = C'L' or C'R'

BAL, SR4

REGISTERS ON EXIT

R5 = Completed name

R4 = Destroyed

DESCRIPTION

The user 'ID' is obtained from JH:SYSID and an 'L' or an 'R' is appended to it to form a file name. An L is used for a LMN file and an R is used for a ROM file.

ID

NFND

PURPOSE

To change a load module name to TEXTC format

USAGE

Input registers:

D1 - D3 = Load module name

BAL, SR3 NFND

REGISTERS ON EXIT

R4 - R5 = Destroyed

R6 - SR1 = TEXTC format LMN

UTS. TECHNICAL MANUAL

ID

PENT

PURPOSE

To insert a parameter word and presence flag into a skeletal PLIST.

USAGE

Input registers:

R5 = Positioned parameter presence bit or -1

R4 = A/M entry address

R7 = Index position from top of PLIST

D1 = Parameter word image

If R5 = -1, D2 = Mask for STS

BAL, SR4 PENT

ERROR EXIT

B ASIZER

If there is insufficient room for a parameter to go into the skeletal PLIST then this exit is taken, an error message is printed and SYNTAX is entered at SYN1.

REGISTERS ON EXIT

Same as upon entry.

DESCRIPTION

PENT tests to see if there is room in the A/M for an entry into the OPEN prime PLIST and if there is, it stores the parameter into the appropriate word and sets the proper presence bit. If it is desired to store into only part of a parameter word, then R5 must contain a -1 and the parameter word is stored selectively according to the mask contained in D2.

UTS TECHNICAL MANUAL

ID

RBLOG

PURPOSE

Convert an alphanumeric Remote Batch Work Station ID into an internal Remote Batch ID and store it in JIT.

USAGE

The alphanumeric Remote Batch Work Station ID must be in the KBUF in TEXTC formation in the M:XX DCB. This buffer is DEF'd KEYBUFF.

BAL, 11 RBLOG
return to BAL+1
successful (SR3) = 0
unsuccessful (SR3) \neq 0

INPUT

M:XX DCB must be closed at entry.

OUTPUT

J:ASSIGN inhibit buffer limit checking
J:CASSIN RBID byte

DESCRIPTION

The file :RBLOG. :SYS is read using the TEXTC work station ID as the key. The first byte of the record is the internal Remote Batch ID (RBID). It is loaded directly into JIT at byte one of J:CASSIN.

If the work station ID is 'LOCAL' by definition the RBID is zero.

ERROR CONDITIONS

If either the file :RBLOG. :SYS or the work station ID record does not exist the error return is taken ((SR3) \neq 0).

UTS TECHNICAL MANUAL

ID

READAM

PURPOSE

To bring the Assign/Merge Table into core.

USAGE

BAL, SR4 READAM

REGISTERS ON EXIT

All restored as when entered

FIELDS MODIFIED

Sets J:ABUF = A/M Table location

DESCRIPTION

This routine obtains a buffer from the FPOOL area and reads the A/M table into it.

ID

REBIT

PURPOSE

To reset the associated ABSDCB and option bit when a DCB assignment is released.

USAGE

R4 = Address of the A/M table

R6 - R7 = TEXTC format DCB name

BAL, SR4 REBIT

REGISTERS ON EXIT

Same as upon entry

UTS TECHNICAL MANUAL

ID

SCAN

PURPOSE

To isolate command line fields one at a time and relate positioning information to the caller.

USAGE

Input registers:

R2 = Field starting position within command line

R3 = Address of where field is to be moved. Zero if no move is to take place.

R1 = Remaining size of Input Message (ARS).

BAL, SR3 SCAN

ERROR EXIT

B SYNTAX

Certain syntax errors result in this exit, such as too many characters within a field or unmatched parentheses.

REGISTERS ON EXIT

R6 = Field Delimiter character (Except EOM is not seen, in this case R1 = 0)

R7 = Number of characters in field exclusive of separators

R8 = Destroyed

R5 = Index into terms table (character type that stopped the scan)

R1 = ARS - Number of characters left to be scanned in message

R2 = Byte displacement of start of next field

DESCRIPTION

The SCAN subroutine picks up a field from the input command. It provides positioning information in exit registers which allow processing of the next field. If R3 is non-zero, it moves the field to the address contained in that register. Field terminators are determined by the contents of table TERMS. Leading and trailing blanks are suppressed. All data enclosed within parentheses is ignored. All characters are considered legal. Maximum field size is limited to ten characters.

UTS TECHNICAL MANUAL

ID

SYNTAX

PURPOSE

To provide a necessary general purpose error handling scheme for the entire TEL processor.

CALLING SEQUENCE

B SYNTAX

Entering through this point causes the !EH message to be output with the character position of the unrecognized entry. Processing continues at SYN1.

After entering here, the TEL environment is cleaned up so that conditions are proper for a retry. The option and flag words in the JIT are reset, the stack is reset and any FPOOL buffer is released.

EXIT

B PROMPT

ID

SYSERR

PURPOSE

To provide handling and cleanup for system detected errors.

CALLING SEQUENCE

B SYSERR

EXIT

B XABORT

Where TEL is effectively aborted then re-entered by the monitor.

SUBROUTINES

WRITERR1 - to read error file and print message

DESCRIPTION

This routine resets JTELFLGS break bit, prints an error message, resets error address in M:XX, resets JABC and JRNST.

ID

TRUNDLE

PURPOSE

To compact a free form Plist by eliminating any zero words.

USAGE

R4 = Address of A/M entry to be compacted
BAL, SR4 TRUNDLE

REGISTERS ON EXIT

Same as upon entry

DESCRIPTION

TRUNDLE changes a free form PLIST into one which is usable by OPEN. Zero words are eliminated by moving non-zero words up over them in a continuing sequence down the Plist. Tab words are a special case as they may legitimately contain zero entries.

UTS TECHNICAL MANUAL

ID

WRITEAM

PURPOSE

To write the A/M table out to RAD

USAGE

BAL, SR4 WRITEAM

REGISTERS ON EXIT

Same as on entry

FIELDS MODIFIED

JCPPO = Op labels for file extension

ID

LOGON, OFF - Initiate on-line session, terminate session

PURPOSE

The purpose of LOGON is to receive a terminal user's request for service and to determine if he is authorized to use the UTS system. When the user's id has been determined to be valid, LOGON puts the information from the user's logon record into his JIT and with proper information inserted, writes the assign-merge record. LOGON then passes control to an installation job initiation routine to police the user. Upon return, LOGON determines whether the user has passed the security check. If not, LOGON will automatically log the user off. Otherwise, LOGON proceeds to call TEL or an auto-call processor (See :USERS file, Section VN.01). LOGON also creates the :USERS logon file (if it does not already exist) when the :SYS, LBE user is first logged on.

LOGON is also entered to log users (batch and terminal) off the system. If the user had previously successfully logged on the system, the RATE file is read and the ACCNTSUM subroutine (described in Section PC.01) is called to compute and log the accounting information, update the current in-use RAD and DISK PACK spaces in the USERS file record and release "STAR" files. To provide one more level of control, ACCNTSUM passes control to an installation accounting routine before writing the accounting log for the user. Writing of the accounting log is suppressed should the installation so desire.

USAGE

If the user is not logged on, LOGON is automatically entered when the user dials in if sense switch two is reset. If the switch is set, LOGON is called by hitting the break key after dialing in.

When a terminal user types "OFF" in TEL, LOGON is called to log the user off the system. LOGON is also called by CCI at the completion of a batch job.

When a terminal user dials onto the system, the monitor constructs a dummy JIT prior to calling LOGON as a shared processor. LOGON fills in the name and account fields, the remaining permanent RAD and DISK PACK spaces, and other user information from USERS file.

INPUT

In response to the message LOGON PLEASE, the terminal user should type his account, name, extended accounting information (if any) and password (if any) in the following format:

ACCOUNT, NAME (ext. acct. info.), PASSWORD CR

Blanks or tabs may be substituted for commas. However, comma is not allowed in the extended accounting information field.

If the USERS file exists, LOGON key-reads the file to verify the name and account of the terminal user and to obtain necessary data for the JIT. As part of the logging off process, LOGON reads the RATE file (see Section IC) for use by the ACCNTSUM subroutine.

OUTPUT

If the USERS file does not exist and the user attempting to logon is :SYS, LBE, the file is created with a record for that account containing the password entered by the user (if any). All fields of the logon record are set to zero with the exception of the account, name and password fields.

Once the user name and account have been verified, an Assign/Merge Table is created and the operator is informed that the user has logged on with the message

USER#XX, AAA, UUUUU

Where XX is the line, UUUUU is the user name and AAA is the account.

If the file "MAILBOX" exists in the user's account file directory, the message

CHECK DC/MAILBOX

is output to the user terminal. If the user's file storage limit has been exceeded, the message

FILE STORAGE LIMIT EXCEEDED (RAD) and/or
FILE STORAGE LIMIT EXCEEDED (PACK)

will be typed to inform him that no new file space can be obtained until some file space has been released.

Associated with the process of logging a user off, the following information is displayed on the terminal for the on-line user:

CPU = .XXXX CON = :04 INT = XX CHG = XXXXX

which shows the amount of CPU time (minute), terminal connected time (minute) number of interactions and charge limits for this session.

INTERACTION

LOGON uses the following monitor services:

M:READ reads input from the user terminal and records from the USERS file in the logon process. Used to read the RATE file in the logoff process.

| | |
|--------------|---|
| M:WRITE | outputs to the user terminal and writes the initial record of the USERS file. Also used to output a null line to the terminal in order to zero the BTD field of the M:UC DCB. |
| M:MESSAGE | used to send messages to the operator's console. |
| M:OPEN | used to open the USERS, RATE and MAILBOX files. |
| M:CLOSE | used to close the USERS, RATE and MAILBOX files. |
| M:TIME | obtains the time for the salutation message and for the JIT. |
| M:SETDCB | used to set error and abnormal returns to a routine which causes them to be ignored. |
| M:WAMR | writes the Assign/Merge Table. |
| M:EXIT | when successfully logged on, exits to TEL or other processor. Otherwise, used to exit to the monitor. |
| M:WAIT | used to wait for one second if the USERS file is busy. |
| M:SUPERCLOSE | used in logoff process to close cooperative files. |

DATA BASES

LOGON accesses the JIT (see Section VA) and USERS file records (see Section VN.01) creates an Assign/Merge Table with proper information in the header (see Section VL) and reads the RATE file (see Section IC) for the ACCNTSUM subroutine.

SUBROUTINES

LOGON uses the following subroutines:

| | |
|----------|--|
| TIMEVERT | subroutine within ACCNTSUM which pulls the hour/minute time from the buffer loaded by M:TIME, converts it to binary minutes from midnight and stores the result into a data word as the user's logon time. |
| ACCNTSUM | subroutine which computes and logs the accounting information, prints an accounting summary, updates the USERS file and releases the STAR file (see Section PC.01). |
| BIN2HEX | subroutine within ACCNTSUM which converts a binary byte to its EBCDIC hexadecimal value. |

- PARSE subroutine which places the user's account into the JIT and the name, account, extended accounting information (if any), and password (if any) into temporary storage.
- CONCAT subroutine used to concatenate the name account for a key-record of the USERS file.
- TRIES subroutine which increments logon error tries and aborts the logon process if the count exceeds a threshold (currently five).
- SCAN subroutine which scans the user's id message and provides input to the PARSE subroutine.

ERRORS

LOGON types the following error messages to the terminal user:

- ID? no user name was input.
- XXXXXXXXXX? (where XXXXXXXXXXXX represents the input sequence) - too many characters were in the input account, name or password fields.
- ACCOUNT? too many characters in account field.
- PASSWORD? the user has a password in his logon record but the input password was not present or is not recognized.
- SORRY, UNABLE TO LOG YOU ON - the user is unable to logon after five attempts. If the logon attempt is being aborted because of a system error, message will also appear with an error message.
- BARRED BY INSTALLATION, SEE SOMEBODY - the user is prohibited by the installation to have any further access to the system.
- UNRECOVERABLE I/O ON RAD - an unrecoverable I/O error was encountered during the read of the USERS file record.
- ABNORMAL ERROR ON LOGON FILE - an abnormal error has been encountered while attempting to open the USERS file.
- ACCOUNT/ID AAAA/UUUUU? - user id AAAA, UUUUU was not recognized as being valid.

LOGON sends the following error messages to the operator's console:

ABNORMAL ERROR ON LOGON FILE

ERROR CODE = XX

A user has logged off and encountered an error in USERS file.

UNRECOVERABLE ERROR READING USERS FILE

ERROR CODE = XX

self explanatory

ERROR IN SYSTEM ACCOUNT FILE DIRECTORY

ERROR CODE = XX

An attempt by LOGON to open the USERS or RATE file has resulted in the detection of an error in the :SYS account file directory.

UNABLE TO READ RATE FILE

ERROR CODE = XX

self-explanatory

UNABLE TO OPEN RATE FILE

ERROR CODE = XX

file exists but couldn't be opened.

RESTRICTIONS

If the logon file called :USERS does not exist, the first person to use the system must logon under the name LBE under the :SYS account. This id will be recognized and the USERS file will be created with a record for this name. If a password is entered with the initial logon, it will be placed in this logon record and will be required for subsequent logons with this name.

All context information is kept in the ACCNTSUM module.

DESCRIPTION

LOGON and ACCNTSUM are grouped into one physical load module to perform the functions of logging users on and off the system. The ACCNTSUM subroutine performs almost all of the operations required to log the terminal or batch user off the system.

When LOGON is accessed, it notes the time and determines whether the user is logging on or off. If he is logging off, a branch is taken to the LOGOFF routine; if not, LOGON sets up a temporary storage area and outputs a salutation message. The user responds by typing in his user account, name, extended accounting information (if any) and password (if any).

LOGON reads the user response while checking for syntax errors and concatenates the name to the account to form a key. A detected syntax error results in the appropriate message being sent to the user and the logon message being reissued. After the key has been built, LOGON attempts to open the :USERS file. If this attempt is unsuccessful because the file is busy, LOGON continues to attempt the open every second until successful. If the file doesn't exist and the user is :SYS, LBE, the USERS file is created and the first record written.

After LOGON has successfully opened the USERS file, it performs a key-read of the file which, if successful, indicates the name and account are valid. The password in the logon file record is then compared with the password input by the user and if they are the same, the user is permitted on the system. If the key-read of the USERS file is unsuccessful, the user is given additional tries until a threshold is reached. At this point, the logon attempt is aborted and the user receives a message informing him that he is unable to logon.

LOGON next puts profile information such as billing rate, privileges, core limits, etc. from the logon record into the user's JIT. Also computed and entered into the JIT is the maximum amount of permanent RAD and DISK PACK spaces the user is permitted to accumulate this session. If this amount is negative, the user is notified that his file storage limit has been exceeded. If an auto-call processor is specified, it is set up as the target for an interpretive exit; otherwise TEL is the target.

LOGON next tests for the existence of a "MAILBOX" file. If it exists, the user is notified to check the mailbox. LOGON then writes the Assign/Merge Table, with information such as user name, logon time, rate structure, system ID etc., inserted in the header. LOGON then passes control to the installation job initiation routine to police the job. Lastly, LOGON sets the UBTD field of the M:UC DCB to zero and does an interpretive exit.

LOGOFF, when entered, first determines whether the user had successfully logged on and, if not, exits. If he had, the RATE file (if it exists) is read and the ACCNTSUM subroutine entered to perform the logoff functions. Before writing the accounting record to the accounting log, LOGON passes control to the installation job accounting routine to determine if the writing should be suppressed. Upon return, a Super Close CAL is issued to close the cooperative files and the processor exits.

UTS TECHNICAL MANUAL

ID

ACCNTSUM - Accounting subroutine (synonymous with ACCTSUM).

PURPOSE

The ACCNTSUM subroutine performs those functions necessary to terminate an on-line session or batch job. These functions include closing and releasing the user's temporary and "STAR" files, displaying and logging the user's accounting information, and updating the permanent RAD and DISK PACK spaces field of the user's logon record.

USAGE

ACCNTSUM is called with a return link in register SR4 and returns with the contents of all registers unchanged. Prior to calling ACCNTSUM, the RATE file (if any) must be read into the buffer, RECORD, and a zero stored into RATEFLAG to indicate that the file has been read. If the rate file is not read, RATEFLAG must be set to a non-zero value. (The RATE file is described in Section IC.) The ACCNTSUM subroutine is loaded with both RECOVER2 and LOGON.

INPUT

ACCNTSUM reads the "T" STAR file to determine (from the keys) the names of the temporary files that the user has accumulated.

OUTPUT

ACCNTSUM outputs an accounting record to the :ACCTLG accounting log file, accounting summary to the batch or on-line user, and an updated logon record to the USERS file.

(The accounting record and accounting summary for batch jobs are described in Section IC, and the :USERS file record is described in Section VN.01.)

The format of the accounting summary message output to the terminal user is shown in the following example:

CPU = .2612 CON = 1:20 INT = 213 CHG = 273514

where CPU is the total execution time in minutes, CON is the total connect time, INT is the number of console interactions and CHG is the total charge units for the session.

The message, "OFF", is output to the operator console by ACCNTSUM for all on-line users.

(Both of the above messages are suppressed when ACCNTSUM is called by RECOVER2.)

INTERACTION

The following monitor services are used:

| | |
|-----------|---|
| M:TIME | puts the time into the accounting buffer. |
| M:WAIT | waits one second if the USERS file is busy. |
| M:MESSAGE | output messages to the operator console. |
| M:OPEN | opens :ACCTLG and USERS files, temporary and "STAR" files in order to release them. |
| M:CLOSE | closes :ACCTLG and USERS files, closes and releases temporary and "STAR" files. |
| M:WRITE | outputs to the line printer (if batch) and user terminals, outputs the accounting and the updated USERS file records. |
| M:READ | reads the user's logon file record, reads the "T" STAR file to obtain the names of the temporary files. |
| M:PFIL | positions to the end of the :ACCTLG file. |
| M:SETDCB | changes error and abnormal return address in the DCBs. |

DATA BASES

ACCNTSUM uses one internal table -- STARFTBL. This is a byte table used to determine the names of the "STAR" files to be released. ACCNTSUM also uses the RATE file (Section IC), the JIT (Section VA) and the :USERS file (Section VN.01).

A 100 word patch area is located in the context area and its label, PATCH, is DEFed.

SUBROUTINES

The following subroutines are used by ACCNTSUM:

| | |
|----------|--|
| RELSTARF | releases all temporary and "STAR" files. |
| UPDATE | updates the current in-use RAD and DISK PACK spaces field of the user's logon file record. |
| DISPLY | displays and logs the user's accounting information. |
| BANNER | outputs two lines of "UTS UTS UTS..." to the printer before and after the batch accounting summary is printed. |
| UPSPACE | spaces a number of lines to separate the banner lines from the accounting summary for batch jobs. |
| TIMEVERT | converts current time to binary minutes from midnight. |
| BINDCB | converts a binary field to its decimal EBCDIC equivalent. |
| ZEROBK | changes leading zeros in an EBCDIC field to spaces. A string with all zeros is converted to one zero with leading spaces. |
| DEC2 | subroutine to build and output to the printer one line of the accounting summary. |
| DECV | same as DEC2 except that a binary number is given to the subroutine which is first converted to its decimal EBCDIC equivalent. |

UTS TECHNICAL MANUAL

| | |
|-----------|---|
| DECP | same as DECV except the number is first divided by three to convert it from clock ties to ten-thousandths of a minute. |
| COMPUTECH | computes the total charge units for the user's job. If RATEFLAG is non-zero, indicating that the RATE file has not been loaded, a net charge of zero is returned. |
| HX | outputs a brief accounting summary to the user's terminal. |
| SETMES | outputs accounting summary lines to the printer for batch jobs. |
| APPEND | appends the user name to account to provide a key for reading the user's logon file record. |
| OPNSTARF | opens the "STAR" file identified by parameters contained in the registers when called. |
| OPNF | used by OPNSTARF and RELSTARF to open temporary files. |

ERRORS

ACCNTSUM sends the following error messages to the operator's console:

ERROR IN SYSTEM ACCOUNT FILE DIRECTORY

ERROR CODE = XX

Error detected when trying to open the :ACCTLG file prior to writing the accounting record.

The logoff process continues but the accounting information is lost for the job.

UNABLE TO OPEN :ACCTLG FILE FOR ACCOUNTING

ERROR CODE = XX

Self-explanatory. Same action as above.

UNABLE TO OPEN USERS FILE FOR UPDATE

ERROR CODE = XX

Self-explanatory. The logoff process continues but the current in-use RAD space kept in the USERS file record is not updated. The message does not appear for ghost jobs or for the jobs logged off by recovery.

BAD LOGON RECORD

ERROR CODE = XX

Error detected while reading or writing the user's logon record. Same action as above.

RESTRICTIONS

The parameter P:JIT must be externally defined as the address of the JIT. In LOGON, P:JIT is equated with J:JIT, in GHOST1 (RECOVER2), it is equated to the area where the user's JIT is kept.

Assembly switches have been installed to permit debugging under BTM. When assembling for UTS, the symbol FORBTM must be equated with zero.

UTS TECHNICAL MANUAL

DESCRIPTION

The ACCNTSUM subroutine is loaded with two modules, LOGON and GHOST1. Its purpose is to perform the functions required to log terminal and batch users off the system. The ACCNTSUM subroutine is comprised of three main subroutines: RELSTARF, UPDATE and DISPLY.

RELSTARF

The RELSTARF subroutine is responsible for closing and releasing all user "STAR" and temporary files. The "STAR" files are assigned when the user accesses the system and the temporary files (if any) are accumulated over the duration of the user's session. Their names are formed by appending the appropriate letter to the user's system ID (right halfword of word zero of the JIT). RELSTARF first closes and releases "STAR" files "A", "B", "D", "G", "L", and "R". The function of each of these files is described below:

- "A" contains the ASSIGN data and DCB names.
- "B" contains information about binary input and is present only if BI is specified on a !LOAD card.
- "D" contains debugging information; e.g., PMD and SNAP cards.
- "G" this is the GO file or RUN file in batch.
- "L" contains local information.
- "R" contains a ROM from a compiler or assembler.

RELSTARF next reads the "T" STAR file and closes and releases each of the user's temporary files. The "T" file consists of a number of keys which are the names of the temporary files created during the user's session.

UPDATE

The UPDATE subroutine causes the in-use RAD and PACK spaces fields of the user's logon record to be updated. When called, the routine builds a key by concatenating the user name to the account in order to select the proper record in the :USERS file. The logon file is now opened and the user's record read. The logon record is then updated and written to reflect the net change in permanent file storages.

Finally, the USERS file is closed and the subroutine exits.

If the attempt to open the USERS file fails because the file is busy, UPDATE continues attempting the open until successful.

DISPLY

The DISPLY subroutine performs the necessary accounting tasks for ACCNTSUM. When it is first entered, it outputs the top banner lines (if the user is batch) to the printer. It then computes the elapsed job time and puts the result in the accounting area and into the on-line accounting summary message or, if batch, outputs the elapsed time line.

DISPLY next calls the COMPUTECH subroutine to compute the charge units accrued during the session. If the RATE file has not been read as indicated by a non-zero value in RATEFLAG, COMPUTECH returns a value of zero for the charge units. If RATEFLAG is set to a zero, COMPUTECH assumes that the RATE file has been read and proceeds to compute the charges. The subroutine first determines which table is to be used from the J:RATE word in the JIT. Each RATE table contains a set of billing rates for such items as CPU time, terminal interactions, etc. (See Section ID). These rates are multiplied by the CPU rate from the RATE table in use. The results are summed to obtain the final figure for the charge units. The resultant total is then sent back to the DISPLY subroutine which places the value in the accounting buffer and the on-line message buffer or the batch message.

DISPLY next computes all of the values listed in the example entitled "Accounting Information For Batch Jobs" in Section ID, places the results in the accounting buffer and, if batch, outputs them to the printer with an explanation message as part of the accounting summary.

When all of the computations have been completed, DISPLY prints the bottom banner lines if batch. If on-line, the number of terminal interactions are computed and the result stored in the accounting area and in the terminal accounting summary message. Next, the core-time charges are computed and stored in the accounting buffer. If the job is on-line and RECOVERY is not running, an accounting summary is output to the terminal and the "OFF" message sent to the operator.

The accounting record for the user is now appended to the end of the :ACCTLG account-log file and the subroutine exits. If the :ACCTLG file is busy when the open is attempted, DISPLY continues the open attempt until successful.

ID

Installation Accounting Routines

PURPOSE

In order to provide the installation with capabilities of policing job entries and modifying accounting charges, exits are made at job and terminal session initiation and termination for the inclusion of supplementary account routines (to be supplied by the installation). The degree of control and scope of sophistication are left to the discretion and needs of the installation.

USAGE

The installation may supply a job or terminal session initiation accounting routine whose DEF'ed entry point is M:ACINIT and a job or terminal session termination accounting routine whose DEF'ed entry point is M:ACTERM. Exits are made to these names which are assembled as SREFs at the following places:

M:ACINIT

- (a) Batch job initiation
- (b) Terminal session log-on

Calling Sequence

BAL, D4 M:ACINIT

Input Registers:

R3 contains the address of the image of LOGON record
R5 contains the address of the JIT

Output Registers:

R3 is either zero or non-zero as determined by this M:ACINIT routine. Upon return from this routine, all registers except R3 are expected to remain intact.

M:ACTERM

- (a) Batch job termination
- (b) Terminal session log-off

UTS TECHNICAL MANUAL

Calling sequence

BAL, D4 M:ACTERM

Input registers:

R3 contains the address of the image of accounting record

Output registers:

R3 contains either zero or non-zero as a result of the action of the M:ACTERM routine.

Upon return from this routine all registers except R3 are expected to remain intact.

In each case, the JIT can be found via location X'4F'. An installation wishing to supplement the standard validation and accounting provided by the system then includes ROMs with the above entry names into the appropriate processor Load Modules, during PASS3 portion of SYSGEN by adding the ROM file names (and account, if needed) to the list of element files on the processor LOCCT commands. The initiation routine, M:ACINIT is called by both LOGRT and LOGON and must be included in both the CCI and LOGON Load Modules. The termination routine, M:ACTERM, is called by ACCTSUM and must be included in both LOGON and GHOST1.

DESCRIPTION

Exit is made at job and terminal session initiation to the installation accounting routine M:ACINIT. At this point, syntax checking on the control command to enter the system has already been performed. Relevant items in the JIT have already been set up to reflect the environment in which the job is to be executed. The extended accounting information, as might be specified on the job entry control command has been recorded in the Assign-Merge Table and items such as the remaining permanent secondary storage for the job have also been stored in the table. In addition, for batch entries, resource requests (core, tape drives, pack spindles) have already been indicated. By pooling the information in JIT, user's :USERS-record and Assign-Merge Table, the installation accounting routine can then police the job. If the routine decides that the job is to be denied further access to the system, a zero should be returned in register R3 in which case the system will output an appropriate message to the job and the user will be aborted.

At job and terminal session termination, exit is made to the installation accounting routine M:ACTERM. At this point, the system has already prepared an accounting record image which contains all the information as displayed in Section PC. 01. However, before the record is actually appended to the system accounting file :ACCTLG, the installation accounting routine has the option of modifying the information in the record image. The routine can also determine whether the record should be included in the :ACCTLG accounting file. In the event that the record is not to be written to the file, a zero in register R3 is returned by the routine to the system and the accounting record will be suppressed.