

TENEX DIRECTORY STUFF

;USEFUL SYMBOLS

NFDIB= # PAGES IN <SYSTEM>DIRECTORY.;1 (40 DIRS / PAGE)
 NSUBI= # SUBINDICES IN <SYSTEM>INDEX.;1
 NDIRHT= SIZE OF HASH TABLE IN <SYSTEM>INDEX.;1 SUBINDEX 0
 NTRACK= # DISK CYLINDERS ON WHOLE SYSTEM (MISNAMED)
 NBWTK= # WORDS TO REPRESENT ALL PAGES ON A CYLINDER WITH A BIT (MISNAMED)
 DSKBSZ= SIZE OF ENTIRE DISK BIT TABLE (BOTH PARTS) IN WORDS
 NPGTK= # PAGES ON A CYLINDER (MISNAMED)

FDFTF/ 0 => NEW FORMAT FILE SYSTEM, -1 => OLD FORMAT FILE SYSTEM
 DIDSCI/ 1ST DISK ADR OF SYSTEM (NEAR SWAPPING TRACKS)
 DIDSCA/ DISK ADR OF INDEX BLOCK OF <SYSTEM>INDEX.;1
 DIDFN/ OFN OF INDEX BLOCK OF <SYSTEM>INDEX.;1
 FDLOFN/ OFN OF PAGE TABLE TABLE OF LONG FILE <SYSTEM>DIRECTORY.;1
 PFDOFN/ TABLE OF POINTERS TO TABLE OF OFNS FOR EACH FILE SYSTEM (JUST 1)
 FDOFN/ TABLE OF OFNS FOR INDEX BLOCKS FOR <SYSTEM>DIRECTORY.;1
 SBIDTB/ TABLE OF 200 7 BIT BYTE SUBINDEX PTRS INDEXED BY ASCII
 CODE OF 1ST CHAR OF DIRECTORY NAME
 ESBIDT/ HASH TABLE IN <SYSTEM>INDEX.;1 SUBINDEX 0
 DSKFCT/ FREE PAGE COUNT PART OF DISK BIT TABLE MAPPED <SYSTEM>DSKBITTBL.;1
 DSKBTB/ BIT TABLE PART OF DISK BIT TABLE MAPPED <SYSTEM>DSKBITTBL.;1
 DBTJFN/ JFN FOR <SYSTEM>DSKBITTBL.;1 (USE JFNDFN TO GET OFN)

DB DEFINITIONS

USE	TPC,0	; Pc to define offsets into ddb
BLOCK 1		; Header not referenced symbolically
DDBNAM:!BLOCK 1		; Lh ==> pointer to password string block
		; Rh ==> pointer to name string block
DDBMAX:!BLOCK 1		; Maximum disk storage for this directory
DDBLOG:!BLOCK 1		; Time and date of last login
DDBPRV:!BLOCK 1		; Privilege bits
DDBMOD:!BLOCK 1		; Mode bits
DDBRES:!BLOCK 1		; Special resource information
DDBNUM:!BLOCK 1		; RH ==> directory number
		; LH ==> pointer to special information block
		; 0 means no special info
DDBDAT:!BLOCK 1		; Date and time of last login
DDBGRP:!BLOCK 1		; Groups to which this user belongs
DDBLEN:!		

;USER DIRECTORY DEFINITIONS

● RTPC==750000 ; Directory origin
USE DIRTPC

DIRORG: BLOCK 0 ; Directory origin
DIRLCK: BLOCK 1 ; Directory lock
DIRUSE: BLOCK 1 ; Directory use indicator
DIRNUM: BLOCK 1 ; Number of this directory
SYMBOL: BLOCK 1 ; Symbol table bottom
SYMTOP: BLOCK 1 ; Symbol table top
DIRFRE: BLOCK 7 ; Free storage header
FRETOP: BLOCK 1 ; Current top of free storage area
DIRDPW: BLOCK 1 ; Default file protection word
DIRPRT: BLOCK 1 ; Directory protection word
DIRDBK: BLOCK 1 ; Default backup specification
DIRGRP: BLOCK 1 ; Groups having access to this directory
DIRSAV: BLOCK 1 ; A place to put pointers to be qc'ed
DIRDSK: BLOCK 1 ; LH = max allocation this directory,
; RH = current allocation
DIREXL: BLOCK 1 ; -1 PREVENTS EXPUNGE
SPARE: BLOCK 1 ; Locations for additional variables
DIRLOC: BLOCK 1 ; Temp used for saving location in the directory
DIRINP: BLOCK 1 ; Temp to save pointer to input for lookup
DIRINC: BLOCK 1 ; Temp to hold increment for searching
DIRMSK: BLOCK 1 ; Temp to hold mask for lookups
DIRSCN: BLOCK 1 ; Temp to save pointer to pointer
DIRFREE: BLOCK 0 ; Beginning of free area

;SUBINDEX 0 IN INDEX.;1 DEFINITIONS USING ABOVE ALSO

DIRHTD==DIRDPW ; Directory number hash table origin
DIRHTL==DIRPRT ; Directory number hash table length
FDADR==:DIRDBK ; Disc address of fd ib
; DEFINED LSTDND
LSTDND==DIRGRP ; LAST ASSIGNED DIRECTORY NUMBER

SBIDTB: BLOCK <200+4>/5 ; Directory subindex dispatch
ESBIOT:

;FDB DEFINITIONS

```
      USE      TPC,0      ; Pc to use to define fdb offsets

      BLOCK 1      ; The header, not referenced symbolically
FDBCTL:!BLOCK 1      ; Lh ==> control bits (see below)
                        ; Rh ==> location of file name block
FDBEXT:!BLOCK 1      ; Lh ==> location of extension block
                        ; Rh ==> pointer to other extensions
FDBADR:!BLOCK 1      ; The file address & class field
FDBPRT:!BLOCK 1      ; File protection word
FDBCRE:!BLOCK 1      ; Creation date & time of version 1
FDBUSE:!BLOCK 1      ; Lh ==> last writer directory number
                        ; Rh ==> use count (+1 for each indirect pointer
                        ;   and saved environment)
FDBVER:!BLOCK 1      ; Lh ==> version number
                        ;   this is job number for temp files
                        ; Rh ==> pointer to other versions
FDBACT:!BLOCK 1      ; Account infor for charging
                        ;   + for location of string block
                        ;   - for number
FDBBYV:!BLOCK 1      ; 0-5 ==> number of version to retain
                        ; 6-11 ==> last byte size
                        ; Rh ==> count of actual pages in file
FDBSIZ:!BLOCK 1      ; Length of file in bytes
FDBCRV:!BLOCK 1      ; Creation date and time of this version
FDBWRT:!BLOCK 1      ; Date & time of last write
FDBREF:!BLOCK 1      ; Date & time of last reference
FDBCNT:!BLOCK 1      ; Lh ==> count of writes
                        ; Rh ==> count of references
FDBBCK:!BLOCK 5      ; Words for backup system
FDBUSW:!BLOCK 1      ; User settable word
FDBLEN:              ; Length of fdb

; Bits in lh of fdbctl

FDBTMP==400000      ; File is temporary
FDBPRM==200000      ; File is permanent
FDBNEX==100000      ; No extension for this fdb yet
                        ;   the file does not really exist
FDBDEL==040000      ; File is deleted
FDBNXF==020000      ; File does not exist (first write not complete)
FDBLNG==010000      ; Long file
FDBSHT==004000      ; Compressed page table
FDBENV==002000      ; Environment file
FDBSUB==001000      ; Subroutine file
FDBUND==000400      ; Undeletable file
FDBEPH==000001      ; Ephemeral subsys file
```

<SYSTEM>DSKBTTBL.;1

DBTJFN: JFN FOR FILE

MAPPED INTO ADR SPACE AS FOLLOWS:

DSKFCT:	FREE PG CNT CYL 0	!	!	INITED TO NPGTK
	FREE PG CNT CYL 1	!	!	
	FREE PG CNT CYL 2	!	NTRACK	
	.	!	DSKBSZ!	
	.	!	V	
	.	!	V	
DSKBTB:	1 BIT PER PAGE FOR CYL 0	!	!	INITED TO BIT=1 UNASSIGNED FOR ALL EXISTANT PAGES
	1 BIT PER PAGE FOR CYL 1	!	!	
	1 BIT PER PAGE FOR CYL 2	!	NTRACK * NBWTK	
	.	!	!	
	.	!	!	
	.	!	V	
	.	!	V	

<SYSTEM>INDEX.;1

		PAGE TABLE	SUBINDEX 0	
DFN--	!			
DIDSCI	!			
V	/			
DIDSCA-->	!	8. WORDS	----->!	DIRLCK (ALSO DIRORG)
	!	8. WORDS	-----\	DIRUSE
	!	8. WORDS	!	DIRNUM: 0
/	!	8. WORDS	!	SYMBOT
	!	8. WORDS	!	SYMTOP
V	!	0	!	DIRFRE HEADER BLOCK
	!	.	!	7. WORDS
	!	.	!	FRETOP
	!	.	!	DIRHTO
	!		!	10 PAGES (4K)
\-- 8. * NSUBI WORDS	!		!	DIRHTL
	!		!	FDADR
	!		!	LSTDND
	!		!	9. UNUSED WORDS
	!		!	SBIDTB (ALSO DIFREE)
	!		!	26. WORDS
	!		!	200 7 BIT BYTES PTING
	!		!	TO SUBINDEX. BYTE
	!		!	INDEXED BY ASCII CHAR
	!		!	ESBIDT (HASH TABLE)
	!		!	NDIRHT WORDS
	!		!	DDB OFFSET,,DIR #
	!		!	0
	!		!	0
	!		!	.
	!		!	.
	!		!	.
	!		!	V
	!		!	-----
	!		!	SUBINDEX 0 < N < NSUBI
	!		!	-----
	!		!	\-->! DIRLCK (ALSO DIRORG)
	!		!	!
	!		!	DIRUSE
	!		!	!

<SYSTEM>DIRECTORY.;1 NEW FORMAT (FDFMTF: 0)

FDLOFN--\

	PTT		PT 0		DIRS	
FDADR-->	1 WD	-->	20 WDS	-->	20 PGS	DIR 0
!	1 WD	/	20 WDS	-->	20 PGS	DIR 1
NFDIB	1 WD	!	20 WDS	-->	20 PGS	DIR 2
!	1 WD	!	20 WDS	-->	20 PGS	DIR 3
V	1 WD	!	20 WDS	-->	20 PGS	DIR 3
---	0	!	20 WDS	-->	20 PGS	DIR 4
	0	!	20 WDS	-->	20 PGS	DIR 5
	0	!	20 WDS	-->	20 PGS	DIR 6
	.	!	.	!	.	
	:	!	:	!	:	
	.	X	.	!	.	

	OFNS		PT 1		DIRS	
FDOFN:	1 WD	!-/ \->	20 WDS	-->	20 PGS	DIR 40
!	1 WD	!-/	20 WDS	-->	20 PGS	DIR 41
NFDIB	1 WD	!	20 WDS	-->	20 PGS	DIR 42
!	1 WD	!	20 WDS	-->	20 PGS	DIR 43
V	1 WD	!	20 WDS	-->	20 PGS	DIR 43
---		!	20 WDS	-->	20 PGS	DIR 44
		!	20 WDS	-->	20 PGS	DIR 45
		!	20 WDS	-->	20 PGS	DIR 46
		!	.	!	.	
		!	:	!	:	
		!	.	!	.	

<SYSTEM>DIRECTORY.;1 OLD FORMAT (FDFMTF: -1)

F0LOFN--\

	PTT		PT 0		DIRS	
FDADR-->	1 WD	----->	10 WDS	----->	10 PGS	1ST HALF DIR 0
?		/				
!	1 WD	!	10 WDS	----->	10 PGS	1ST HALF DIR 1
NFDIB	=====	!				
!	1 WD	\	10 WDS	----->	10 PGS	1ST HALF DIR 2
!		!				
V	1 WD	!	10 WDS	----->	10 PGS	1ST HALF DIR 3
---		!				
	0	!	10 WDS	----->	10 PGS	1ST HALF DIR 4
		!				
	0	!	10 WDS	----->	10 PGS	1ST HALF DIR 5
		!				
	0	!	10 WDS	----->	10 PGS	1ST HALF DIR 6
		!				
	.	!	.	!	.	
	:	!	:	!	:	
	.	X	.	!	.	

	GFNS		PT NFDIB/2		DIRS	
FDOFN:	1 WD	!-/ \->	10 WDS	----->	10 PGS	2ND HALF DIR 0
?		/				
!	1 WD	!	10 WDS	----->	10 PGS	2ND HALF DIR 1
!	=====	!				
NFDIB	1 WD	!-/	10 WDS	----->	10 PGS	2ND HALF DIR 2
!		!				
V	1 WD	!	10 WDS	----->	10 PGS	2ND HALF DIR 3
---		!				
		!	10 WDS	----->	10 PGS	2ND HALF DIR 4
		!				
		!	10 WDS	----->	10 PGS	2ND HALF DIR 5
		!				
		!	10 WDS	----->	10 PGS	2ND HALF DIR 6
		!				
		!	.	!	.	
		!	:	!	:	
		!	.	!	.	

USER DIRECTORY

```

DIRORG: !-----!
! DIRLCK !
!-----!
! DIRUSE !
!-----!
! DIRNUM !
!-----!
! SYMBOT !
!-----!
! SYMTOPT !
!-----!
! DIRFRE HEADER BLOCK !
! 7. WORDS !
!-----!
! FRETOP !
!-----!
! DIRDPW ! 20 PAGES (8K)
!-----!
! DIRPRT !
!-----!
! DIRDBK !
!-----!
! DIRGRP !
!-----!
! DIRSAV !
!-----!
! DIREXL !
!-----!
! SPARE !
!-----!
! DIRLOC !
!-----!
! DIRINP !
!-----!
! DIRINC !
!-----!
! DIRMSK !
!-----!
! DIRSCN !
!-----!
! DIRGRP !
!-----!
! DIFREE FREE STORAGE !
! FDBS AND STRING BLOCKS !
!-----!
! BLOCK TYPES: !
! 400100,,LEN = FDB !
! 400001,,LEN = NAME !
! 400002,,LEN = EXT !
! -1,,LEN = ACCT !
!-----!
! FDB !
!-----!
! !
! 400100,,FDBLEN !
!-----!

```



```

;ASSEMBLY OF SENSITIVE PAGE HARDWARE ADRS FOR ASSIGNMENT TO ERROR FREE AREA
;ON CDC WINCHESTER DRIVES
;CYL 0, HEADS 0 AND 1 ARE AVAILABLE.
;HEAD 1 IS ASSUMED SAFER THAN HEAD 0 (LESS CHANCE OF HUMAN ERROR CLOBBERING IT)
;CYL 0, HEADS 0 AND 1, SECTOR 0 ARE ASSUMED UNSAFE (MUCH CHANCE OF HUMAN ERROR)

```

```

;LAYOUT
;* DENOTES MAY NOT BE PRESENT (USUALLY ONLY FOUND ON PACK 0 OF A STRUCTURE)

```

```

;CYLINDER 0, HEAD 0 (TRACK 0)

```

								SECTOR #					
0	1	2	3	4	5	6	7						
!	CORE	!	HOMBAT	!	HOME	!	BAT	!	MONITOR	!	CORE	!	CORE
!	DMP 768	!	FILE 2	!	BLOCK 2	!	BLOCK 2	!	.SAV	!	DMP 256	!	DMP 512
!	XB *	!	XB	!		!		!	XB	!	XB *	!	XB *

```

;CYLINDER 0, HEAD 1 (TRACK 1)

```

								SECTOR #					
0	1	2	3	4	5	6	7						
!	CORE	!	HOMBAT	!	HOME	!	BAT	!	MICRO	!	MICRO	!	BOOT
!	DMP 1024	!	FILE 1	!	BLOCK 1	!	BLOCK 1	!	CODE	!	DIAG	!	STRAP
!	XB *	!	XB	!		!		!	XB	!	XB	!	XB

HOME BLOCKS

NOTE: ALL DISK ADDRESSES ARE IN HARWARE FORMAT SPECIFIED BY HOMHWO.
EACH 9 BIT FIELD SPECIFIES # BITS IN RESPECTIVE FIELDS OF
HARDWARE FORMAT DISK ADR. CURRENTLY 6.,13.,,8.,8.

* => ADR MAY NOT BE ON THIS PACK (PTS TO ANOTHER PACK IN STRUCTURE)

HOMNAM	'HOME'	
HOMID	'ID'	
HOMVER	VERSION ,, LENGTH (WDS)	-->
HOMHWO	UNIT BITS, CYL BITS ,, SRF BITS, SEC BITS	
HOMHW1	# CYLS THIS PACK ,, # SURFACES THIS PACK	
HOMHW2	# SECTORS / TRACK ,, # WORDS / SECTOR	
HOMHW3	# SECTORS / THIS PACK ,, # SECTORS / CYL	
HOMHW4	UNUSED	
HOMHB1	PRIMARY HOMBAT XB ADR	
HOMHB2	SECONDARY HOMBAT XB ADR	
HOMMDA	MICRO DIAG XB ADR	
HOMMDF	MICRO DIAG FLAGS	
HOMMCA	MICROCODE XB ADR	
HOMMCF	MICROCODE FLAGS	
HOMBTA	BOOTSTRAP XB ADR	
HOMBTF	BOOTSTRAP FLAGS	
HOMMNA	MONITOR XB ADR	
HOMMNF	MONITOR FLAGS	
HOMCOA *	CORDMP.256 XB ADR	
HOMCOF	CORDMP.256 FLAGS	
HOMC1A *	CORDMP.512 XB ADR	
HOMC1F	CORDMP.512 FLAGS	
HOMC2A *	CORDMP.768 XB ADR	
HOMC2F	CORDMP.768 FLAGS	

HOMC3A *	CORDMP.1024 XB ADR	
HOMC3F	CORDMP.1024 FLAGS	
HOMSNM	'STRUCTURE NAME'	
HOMSFG	STRUCTURE FLAGS	
HOMSPK	TOTAL STR PACKS ,, THIS PACK #	
HOMPFG	THIS PACK FLAGS	
HOMSWP	1ST SWAPPING CYL ,, LAST SWAPPING CYL	
HOMNX1 *	INDEX.;1 XB ADR	
HOMNX2 *	INDEX.;2 XB ADR	
HOMDR1 *	DIRECTORY.;1 XB ADR	
HOMDR2 *	DIRECTORY.;2 XB ADR	
HOMSPR	SPARE	
	.	
	.	
	.	
	SPARE	
HOMCOD	707070	
HOMSLF	THIS HOME BLOCK ADR	V

BAT BLOCKS

NOTE: ALL DISK ADDRESSES ARE IN HARWARE FORMAT SPECIFIED BY BATHWO.

SINCE THE UNIT FIELD OF THE DISK ADR IS MEANINGLESS, SECTOR COUNT OCCUPIES THAT FIELD.

BATNAM	'BAT'	
BATID	'ID'	
BATVER	VERSION ,, LENGTH (WDS)	-->
BATHWO	CNT BITS, CYL BITS ,, SRF BITS, SEC BITS	
BATHB1	PRIMARY HOMBAT XB ADR (PACK RELATIVE)	
BATHB2	SECONDARY HOMBAT XB ADR (PACK RELATIVE)	
BATFRE	FREE COUNT	
BATCNT	INITIAL COUNT ,, ADDED COUNT	
BATSPR	SPARE	
BATDAT	SECTOR CNT ! START ADDRESS (W/D UNIT)	
	SECTOR CNT ! START ADDRESS (W/D UNIT)	
	.	
	.	
	.	
BATCOD	SECTOR CNT ! START ADDRESS (W/D UNIT)	
	0	
	0	
BATSLF	606060	
	THIS BAT BLOCK ADR (PACK RELATIVE)	V

;DSKPAR GTTAB TABLE - USED BY CHECKDISK,BLKLUK,SWPBAT, MAYBE OTHERS

DSKPAR: LOTRK ;0 = FIRST CYLINDER ON STRUCTURE
HITRK ;1 = LAST CYLINDER+1 ON STRUCTURE
NSECTK ;2 = # SECTORS / CYLINDER
NWSEC ;3 = # WORDS / SECTOR
NTKUN ;4 = # CYLINDERS / PACK
NPACKS ;5 = # PACKS ON STRUCTURE
SWPTK ;6 = OTH SWAPPING TRACK ON A PACK
SWPTKH ;7 = LAST + 1 SWAPPING CYLINDER ON A PACK
NSURFS ;10 = # SURFACES OR HEADS / PACK
NSECS ;11 = # SECTORS / TRACK (NOT PER CYLINDER)
NSECPG ;12 = # SECTORS / PAGE
NBWTK ;13 = # BITTABLE WORDS / CYLINDER
DSKBSZ ;14 = SIZE OF BITTABLE (COUNT + BIT AREAS)
NMINFP ;15 = DSKASN FREE CHOICE PARAMETER
-DSKMSK,, -1 ;16 = MASK OF BITS IN SOFTWARE ADR
<NUNTB>B8+<NCYLBT>B17+<NSRFBT>B26+<NSECBT>B35 ;17 = 9 BIT HDWR SPEC
POINT NUNTB,0,UNTLB ;20 = UNIT PART OF HDWR DISK ADR
POINT NCYLBT,0,CYLLSB ;21 = CYL PART OF HDWR DISK ADR
POINT NSRFBT,0,SRFLSB ;22 = SURFACE OR HEAD PART OF HDWR DISK ADR
POINT NSECBT,0,SECLSB ;23 = SECTOR PART OF HDWR DISK ADR

NDSKPR==.-DSKPAR

;NUMBER OF WORDS IN THIS GTTAB TABLE

;HOMBAT PARAMETER TABLE
 ;SPECIFIES WHERE VARIOUS HOMBAT RELATED PAGES LIVE
 ;BO MEANS ON ALL PACKS, OBO MEANS ON PRIMARY PACK OF STRUCTURE ONLY

HOMBAT: VERHOM,,HOMLEN ;LATEST HOME BLOCK VER,,LEN KNOWN TO MONITOR
 VERBAT,,BATLEN ;LATEST BAT BLOCK VER,,LEN KNOWN TO MONITOR
 FREBAT ;INITIAL FREE COUNT FOR VERBAT VERSION OF BAT BLK
 BATSCT ;POINTER TO SECTOR COUNT FIELD FOR VERBAT
 BATADR ;POINTER TO HDWR DSK ADR FIELD FOR VERBAT
 PRMUNT ;PRIMARY UNIT OF A STRUCTURE
 1BO!<CYLALL>B<CYLLSB>!<SRFXB1>B<SRFLSB>!<SECXB1>B<SECLSB> ;XB1
 1BO!<CYLALL>B<CYLLSB>!<SRFHM1>B<SRFLSB>!<SECHM1>B<SECLSB> ;HM1
 1BO!<CYLALL>B<CYLLSB>!<SRFBB1>B<SRFLSB>!<SECBB1>B<SECLSB> ;BB1
 1BO!<CYLALL>B<CYLLSB>!<SRFXB2>B<SRFLSB>!<SECXB2>B<SECLSB> ;XB2
 1BO!<CYLALL>B<CYLLSB>!<SRFHM2>B<SRFLSB>!<SECHM2>B<SECLSB> ;HM2
 1BO!<CYLALL>B<CYLLSB>!<SRFBB2>B<SRFLSB>!<SECBB2>B<SECLSB> ;BB2
 1BO!<CYLALL>B<CYLLSB>!<SRFMIC>B<SRFLSB>!<SECMIC>B<SECLSB> ;MIC
 1BO!<CYLALL>B<CYLLSB>!<SRFMCD>B<SRFLSB>!<SECMCD>B<SECLSB> ;MCD
 1BO!<CYLALL>B<CYLLSB>!<SRFBOT>B<SRFLSB>!<SECBOT>B<SECLSB> ;BOT
 1BO!<CYLALL>B<CYLLSB>!<SRFMON>B<SRFLSB>!<SECMON>B<SECLSB> ;MON
 OBO!<CYLALL>B<CYLLSB>!<SRFCD0>B<SRFLSB>!<SECCD0>B<SECLSB> ;CDO
 OBO!<CYLALL>B<CYLLSB>!<SRFCD1>B<SRFLSB>!<SECCD1>B<SECLSB> ;CD1
 OBO!<CYLALL>B<CYLLSB>!<SRFCD2>B<SRFLSB>!<SECCD2>B<SECLSB> ;CD2
 OBO!<CYLALL>B<CYLLSB>!<SRFCD3>B<SRFLSB>!<SECCD3>B<SECLSB> ;CD3

NHOMBA==.-HOMBAT

TYMOP JSYS DOCUMENTATION

TYMOP JSYS 654

TYMnet utility Operator; performs a variety of TYMNET utility functions.

Accepts in 1: 0-8 option bits
9-17 function
18-35 TYMNET tty designator (400000+N)

2 & 3: function dependent arguments

TYMOP

Returns + 1: Unsuccessful, with error number in 1, and if supervisor error, supervisor error number in 2
2: Successful: 1,2 & 3 possibly updated

NOTES ON CARRIER OFF INTERRUPTS:

Carrier off interrupts for tymnet lines will be issued as usual via the terminal interrupt word for the job. Controlling terminals for jobs will result in terminal code 30 ints when zapped. Non-controlling terminals will result in terminal code 32 ints when zapped. That means aux circuits on the initiating end and terminating end will both get terminal code 32 ints unless those ends wind up being controlling ttys for some job in which case their zapping will result in terminal code 30 interrupts.


```

MESSAGE (ETM,IGNORE,B,1,1,1,1) ;23 enter 2741 transparent mode
MESSAGE (LTM,IGNORE,B,1,1,1,1) ;24 leave 2741 transparent mode
MESSAGE (LOS,BLACK,C,1,1,1,1) ;25 lost ball, data has been lost from
;buffers. the data filed may tell how
;many were lost
MESSAGE (SUP,IGNORE,C,1,0,1,0) ;26 supervisor request (aux circuits)
MESSAGE (SUR,SUPRSP,C,0,0,1,0) ;27 supervisor response (aux circuits)
MESSAGE (AXC,IGNORE,C,0,0,0,1) ;30 supervisor string character
MESSAGE (TSP,IGNOR2,F,0,0,0,0) ;31 test pattern probe
MESSAGE (TSR,IGNOR2,F,0,0,0,0) ;32 test pattern response
MESSAGE (SAD,IGNOR2,F,0,0,0,0) ;33 host sad
MESSAGE (ECN,IGNORE,B,1,1,1,1) ;34 echo on
MESSAGE (ECF,IGNORE,B,1,1,1,1) ;35 echo off
MESSAGE (TCS,IGNORE,D,1,1,1,1) ;36 term characteristics, first data
;byte indicates which characteristics
;second data byte indicates
;value to set to
MESSAGE (TCP,IGNORE,C,1,1,1,1) ;37 term characteristics probe, data byte
;indicates which terminal characteristic
;were requested
MESSAGE (TCR,IGNORE,D,1,1,1,1) ;40 term characteristics response, data
;is just like tcs, comes in response
;to a probe; also is reflected by remote
;when terminal characteristics are sent
MESSAGE (HSI,BAD,C,0,0,0,0) ;41 host up and answering with # of
;ports in port byte, and host # in data
;byte
MESSAGE (CLP,BAD,A,0,0,0,0) ;42 SUPER CLOCK INFO REQUEST
MESSAGE (CLR,IGNOR2,F,0,0,0,0) ;43 SUPER CLOCK INFO
MESSAGE (BKD,BAD,H,0,0,0,0) ;44 START BLK OUTPUT
MESSAGE (BOC,TYMBOC,B,0,0,0,0) ;45 BLK OUPUT COMPLETE
MESSAGE (BIN,BAD,H,0,0,0,0) ;46 START BLK INPUT
MESSAGE (INB,TYMINB,D,0,0,0,0) ;47 BLK INPUT DONE, OUT OF BUFFER
MESSAGE (INE,TYMINE,D,0,0,0,0) ;50 BLK INPUT DONE, COMPLETED
MESSAGE (INT,TYMINT,D,0,0,0,0) ;51 BLK INPUT DONE, TIMEOUT
MESSAGE (IRQ,BAD,B,0,0,0,0) ;52 RQST BLK INPUT TERMINATION
MESSAGE (IHR,TYMIHR,D,0,0,0,0) ;53 BLK INPUT DONE, HOST REQUEST
MESSAGE (ORQ,BAD,B,0,0,0,0) ;54 RQST BLK OUTPUT TERMINATION
MESSAGE (OHR,TYMOHR,B,0,0,0,0) ;55 BLK OUTPUT DONE, HOST REQUEST
MESSAGE (NEG,IGNORE,C,0,0,0,0) ;56 BLK IO PORT NEGOTIATION
MESSAGE (BRK,IGNORE,B,0,0,0,0) ;57 BREAK CHR RECVD
MESSAGE (PSR,IGNORE,C,0,0,0,0) ;60 PORT STATUS
MESSAGE (EAD,BAD,B,0,0,0,0) ;61 ENTER ALT OUTPUT DEVICE MODE

```

NOTES ON AUX CIRCUIT QUEUEING:

GLOSSARY:

DJOB - destination job for which aux circuit is queued
(a negative DJOB usually means SYSNDX slot #)

SJOB - source job which queued aux circuit

EDBITS - internal status bits associated with DJOB

UDBITS - user settable status bits associated with DJOB

ESBITS - internal status bits associated with SJOB

USBITS - user settable status bits associated with SJOB

SYSNDX - index into SYSNDX table. system jobs enter themselves into predetermined SYSNDX slots to give aux circuit queuers a handle on those system jobs. this allows for listeneing socket style communications with system jobs similar to the way the arpanet does things.

HOST TO HOST PROTOCOL - talked by local and remote hosts while establishing aux circuits.

The HOST TO HOST protocol looks like 200,F,D0,D1,D2,D3,200

where F = function (0=EXEC, 1=QUEUE)

Dn = data bytes (only hi 4 bits of D3 significant)

F=0 (EXEC) : D0,D1,D2,hi D3 = IGNORED, should be 0

F=1 (QUEUE): D0,D1,D2,hi D3 = UDBITS,,DJOB right justified
neg DJOB => SYSNDX

When an aux circuit is established, the HOST TO HOST protocol is expected as the first bytes down the pipe. Anything that does not look like the HOST TO HOST protocol is passed thru and the function is defaulted to EXEC. Note that at least one character is needed to make the determination as to whether or not the user at the initiating end is talking the HOST TO HOST prototcol. If the initiating host does not talk the HOST TO HOST PROTOCOL and the target host does, it is recommended that a garbage character be sent as the first character down the pipe to allow for this determination. A rubout will be gobbled and not considered as part of the data stream. Any other character will be passed through to an exec. The rubout can be considered an escape character only if it is the first character down the pipe.

SCENERIO OF TYPICAL USE OF AUX CIRCUIT QUEUEING:

User on HOST A wants to talk to system process occupying SYSNDX slot N HOST B via an aux circuit. (Such as a file transfer server).

ON HOST A:

User executes AUXBLD function of TYMOP jsys to establish aux circuit to HOST B. User specifies F=QUEUE function and provides SYSNDX # N as =N job # in data bytes. Ignore UDBITS for now. The user then hangs on a bin on the aux circuit waiting for a hello from the desired system process.

ON HOST B:

HOST B starts up a job for that aux circuit just created which talks the HOST TO HOST PROTOCOL as part of its initialization. The job sees the F=QUEUE function desired and uses the QUEAUX function of TYMOP to queue the aux circuit for the indicated SYSNDX slot associated job. On success the job logs out leaving the aux circuit flapping, sitting in the queue tables. Eventually the target system process comes around and claims (via the CLMAUX function of TYMOP) the aux circuit and does whatever it is supposed to with it. Normally the system process would send some initial hello down the aux circuit which the user on HOST A would see since he is hanging on a bin on that aux circuit. At this point the communication pipe to the system process is complete.

Either end may release the aux circuit with the AUXRLS function of TYMOP breaking the connection with zaps etc.

PRETTY PICTURES:
INTERNAL MONITOR TABLES OF INTEREST TO PROGRAMMERS

TABLE OF SYSNDX SLOTS AND CORRESPONDING JOB #'S:
INDEXED BY SYSNDX #

SYSNDX:	!	JOB #	!	;OCCUPIED
	!	-1	!	;FREE

length is NSYSND

TABLE OF DESTINATION JOB AUX CIRCUIT QUEUE ENTRIES:
INDEXED BY LINE #

AUXDJQ=DUXDJQ=TYMTTL

	9 bits	9 bits	18 bits		
DUXDJQ:	!	EDBITS !	UDBITS !	JOB #	!
	!		-1		!

length is NTYMTT

EDBITS: 1b0 = job # obtained from SYSNDX
UDBITS: user settable

TABLE OF SOURCE JOB AUX CIRCUIT QUEUE ENTRIES:
INDEXED BY LINE #

AUXSJQ=DUXSJQ=TYMTTL

	9 bits	9 bits	18 bits		
DUXSJQ:	!	ESBITS !	USBITS !	JOB#	!
	!		-1		!

length is NTYMTT

ESBITS: 1b0 = job logged out since queuing
USBITS: user settable

TABLE OF TIMEOUT AUX CIRCUIT QUEUE ENTRIES:
INDEXED BY LINE #

AUXTMQ=DUXTMQ=TYMTTL

DUXTMQ:	!	todclk at which timeout occurs	!	;OCCUPIED
	!	-1	!	;FREE

length is NTYMTT

TABLE OF HOST TO HOST PROTOCOL ENTRIES:
INDEXED BY LINE #

AUXCTL=DUXCTL=TYMTTL

	8 bits	8 bits	8 bits	8 bits	4 bits							
DUXCTL:	!	F	!	D0	!	D1	!	D2	!	D3	!	;OCCUPIED
	!					-1					!	;FREE

length is NTYMTT

NOTE: only hi 4 bits of D3 retained

NOTES ON BLOCK IO ***** not debugged yet *****

● BLOCK IS 8 BIT BYTES PACKED 4 PER WORD LEFT JUSTIFIED
A BLOCK WILL NOT BE MORE THAN 512. WORDS LONG (1 PAGE)
A BLOCK MUST START ON PAGE BOUNDRY
THE COUNT DOES NOT INCLUDE 0TH WORD (1 <= COUNT <= 511.)

;0TH WORD OF BLOCK IS OVERHEAD WORD.
;COUNT IS 1 TRHU 777 AND DOES NOT INCLUDE THE OVERHEAD WORD.
;RETURNED COUNT < DESIRED COUNT IS NOT CONSIDERED AN ERROR AND
;MUST BE CHECKED BY THE USER.

WHAT A BLOCK LOOKS LIKE:

```

-----
+0  !                               HEADER WORD                               !
-----
+1  !          32 BITS (4 8 BIT BYTES)          !  IGNORED 4 BITS !
-----
+2  !          32 BITS (4 8 BIT BYTES)          !  IGNORED 4 BITS !
-----
+3  !          32 BITS (4 8 BIT BYTES)          !  IGNORED 4 BITS !
-----
      :
-----
●76 !          32 BITS (4 8 BIT BYTES)          !  IGNORED 4 BITS !
-----
+777 !          32 BITS (4 8 BIT BYTES)          !  IGNORED 4 BITS !
-----

```

OVERHEAD WORD (0TH WORD OF AN IO BLOCK):

AT TIME OF CALL:

```

-----
!          18 BITS          ,,          9 BITS          !          9 BITS          !
-----
!          0                ,,          0                !  DESIRED COUNT  !
-----

```

RETURNED (UNLESS INHIBITED)

```

-----
!          18 BITS          ,,          9 BITS          !          9 BITS          !
-----
!  XFER ERROR BITS OR 0    ,,  RETURNED COUNT !  DESIRED COUNT !
-----

```

```

;HEADER WORD ERROR BITS FOR XFER ERRORS
●KERB==400000          ;OUT OF BUFFER ERROR
●LKERT==200000        ;BASE TIMEOUT ERROR

```

FUNCTIONS:

1 - BUILD AUXILIARY CIRCUIT: (AUXBLD)

ACCEPTS:

1/ option bits,1,,0

where option bits are: B0 - Wait for AUXBLD process if busy
B1 - F,DATA IS IN 3 FOR HOST TO HOST PR.
B2 - DON'T SEND ANY HOST TO HOST PROT.
B3 - USE USERNAME PROVIDED IN 2
B4 - USE JOB'S ENTRY TYMNET USERNAME

0B3 & 0B4 => USE SYSTEM DEFAULT TYMNET USERNAME

1B3 & 1B4 => ERROR, CONFLICTING OPTION BITS

2/ Asciz string pointer to signon string in user space

3/ F,DATA where F=0 for EXEC and data is don't care

F=1 for QUE for DJOB using udbits in data

for F=1: data is UDBITS,,DJOB right justified
neg DJOB => SYSNDX

Contents of 3 are specification of optional Host to Host Protocol. If not specified, default F,DATA=0 is used.

F is left justified 8 bits, DATA is rest of 36 bit word.

RETURNS:

+1 Error code in 1, possibly supervisor error code in 2

+2 Success, TTY designator in 1

If b3 is not lit, any user name provided by the user in the signon string is replaced with:

0b4: the contents of AUXNAM in the JSB if AUXNAM has been initied.

1b4: the system default aux circuit building username.

AUXNAM is initied when a tymnet connection initiates a job.

Only wheels and tymox priv may light b3 without error.

When b3 lit, full user provided signon string is used.

2 - GET TYMNET LINE INFO: (GETLNS)

ACCEPTS:

1/ option bits,2,,tty designoator

where option bits are: NONE

RETURNS:

+1 error code in 1

+2 success, 2/ login info (TYNODS)

3/ status bits (TYMLNS)

3 - SET INPUT BACK PRESSURE: (SETIBP)

ACCEPTS:

1/ option bits,3,,tty designoator
where option bits are: NONE

RETURNS:

+1 error code in 1
+2 success, input back pressure set in TYMLNS (IBPOF)

4 - GET TYMNET LINE INTERRUPT INFO: (GETINS)

ACCEPTS:

1/ option bits,4,,tty designoator
where option bits are: NONE

RETURNS:

+1 error code in 1
+2 success, 2/ ball code mask (TYMCMM)
3/ terminal code (TYMCMCI)

5 - SET TYMNET LINE INTERRUPT INFO: (SETINS)

ACCEPTS:

1/ option bits,5,,tty designoator
where option bits are: NONE
2/ ball code mask (TYMCMM)
3/ terminal code (TYMCMCI)

RETURNS:

+1 error code in 1
+2 success

6 - QUEUE AN AUX CIRCUIT TTY FOR DJOB (QUEAUX)

ACCEPTS:

1/ option bits,6,,tty designator
2/ 0,USBITS,,0
3/ 0,UDBITS,,<DJOB (or SYSNDX # if negative)>

where option bits are: NONE

RETURNS:

+1 1/ error code
+2 success, aux circuit queued

7 - CLAIM AUX CIRCUIT QUEUED FOR US WHEN "AND" OF UDBITS NON=0 (CLMAUX)

ACCEPTS:

1/ option bits,7,,0
2/ 0,<UDBITS or 0>,,<DJOB (or SYSNDX if neq) if option bit 1B1 on>

where option bits are:

1B1 = DJOB (or SYSNDX if neg) in RH 2 !REQUIRES WHEEL!
1B3 = Listen if no jobs found

NOTE: 1B1 OFF => (RH 2 <= JOBNO)

NOTE: UDBITS IN 2 WILL BE USED ONLY IF NON=0
(SINCE 0 COULD NEVER WIN AND)

RETURNS:

+1 error code in 1

+2 success = updated ACS in 1,2,3 (3 gets SJOB stuff)

10 - DEQUEUE AUX CIRCUIT (DEQAUX)

ACCEPTS:

1/ option bits,10,,tty designator
2/ 0,<UDBITS or 0>,,<DJOB (or SYSNDX if negative)>
3/ SJOB

where option bits are:

1b1 = ignore RH 2 and dequeue all jobs queued by SJOB
1B3 = SJOB in 3 !REQUIRES WHEEL!
1B4 = tty designator in 1 (ignore 2)

NOTE: UDBITS IN 2 WILL BE USED ONLY IF NON=0
(SINCE 0 COULD NEVER WIN AND)

RETURNS:

+1 error code in 1

+2 success, ACS 1,2,3 updated

11 - SET SYSNDX ENTRY (SETSYS) !REQUIRES WHEEL!

ACCEPTS:

1/ option bits,11,,0
2/ SYSNDX
3/ JOB #

where option bits are: 1B0 = JOB # in 3

RETURNS:

+1 1/ ERROR CODE

+2 ENTERED

12 - CLEAR SYSNDX ENTRY (CLRSYS) !REQUIRES WHEEL!

ACCEPTS:

1/ option bits,12,,0
2/ SYSNDX

where option bits are: NONE

RETURNS:

+1 1/ error code

+2 cleared, 1/ previous contents just clobbered

13 - GET SYSNDX ENTRY (GETSYS)

ACCEPTS:

1/ option bits,13,,0

2/ SYSNDX

where option bits are: NONE

RETURNS:

+1 1/ error code

+2 1/ contents, -1 if free

14 - RELEASE AUX CIRCUIT (AUXRLS)

ACCEPTS:

1/ option bits,14,,tty designator

where option bits are: NONE

RETURNS:

+1 error code in 1

+2 success

15 - GET LAST RECVD BALL CODE (GETCMD)

ACCEPTS:

1/ option bits,15,,tty designator

where option bits are: NONE

RETURNS:

+1 error code in 1

+2 2/ ball code, subtype, data in bytes 0,1,2

16 - SEND BALL CODE (SNDCMD)

ACCEPTS:

1/ option bits,16,,tty designator

2/ ball code, subtype, data in bytes 0,1,2

RETURNS:

+1 error code in 1

+2 success

17 - SET BLOCK MODE INPUT FOR LINE (SETBKI)

ACCEPTS:

1/ option bits,17,,tty designoator
where option bits are:
B0 IF BLOCK MODE DESIRED, OFF IF NOT

RETURNS:

+1 error code in 1
+2 success, BLK MODE FLAG set/cleared in TYMLNS (BLKMOD)
NODE NOTIFIED APPROPRIATELY

20 - SET BLOCK MODE OUTPUT FOR LINE (SETBKO)

ACCEPTS:

1/ option bits,20,,tty designoator
where option bits are:
B0 IF BLOCK MODE DESIRED, OFF IF NOT

RETURNS:

+1 error code in 1
+2 success, BLK MODE FLAG set/cleared in TYMLNS (BLKMOD)
NODE NOTIFIED APPROPRIATELY

21 - GET ID BLOCK FOR LINE (GETBLK)

ACCEPTS:

1/ option bits,21,,tty designoator
where option bits are:
B0: 1 - POKE TERMINAL INT 33 ON COMPLETION
0 - WAIT FOR COMPLETION
B1: 1 - DON'T RETURN STATUS IN BLOCK HEADER WRD
0 - ALLOW RETURN OF STATUS IN BLOCK HEADER WRD
2/ START ADR OF BLOCK (ON PAGE BOUNDRY)

RETURNS:

+1 error code in 1
+2 success, UPDATED OTH WORD IN BLOCK IF APPROPRIATE

22 - SEND ID BLOCK FOR LINE (SNDBLK)

ACCEPTS:

1/ option bits,22,,tty designoator
where option bits are:
B0: 1 - POKE TERMINAL INT 34 ON COMPLETION
0 - WAIT FOR COMPLETION
B1: 1 - DON'T RETURN STATUS IN BLOCK HEADER WRD
0 - ALLOW RETURN OF STATUS IN BLOCK HEADER WRD
2/ START ADR OF BLOCK (ON PAGE BOUNDRY)

RETURNS:

+1 error code in 1

+2 success, UPDATED OTH WORD IN BLOCK IF APPROPRIATE

ERROR CODES FOR TYMOP JSYS:

TYMXX1=601200	;AUX CIRCUIT BUILDING MECHANISM BUSY
TYMXX2=601201	;AUX CIRCUIT BUILDING MECHANISM TIMED OUT
TYMXX3=601202	;NO FREE LINE AVAILABLE
TYMXX4=601203	;LINE MYSTERIOUSLY ZAPPED PREMATURELY
TYMXX5=601204	;SUPERVISOR REPORTS ERROR
TYMXX6=601205	;ILLEGAL RECEIVE BITS REQUEST
TYMXX7=601206	;ILLEGAL TYMOP FUNCTION CODE
TYMXX8=601207	;ILLEGAL TYMNET TTY
TYMXX9=601210	;JOB DOES NOT OWN TTY
TYMX10=601211	;INSUFFICIENT CAPS
TYMX11=601212	;ILLEGAL JOB #
TYMX12=601213	;ILLEGAL SYSNDX
TYMX13=601214	;SYSNDX SLOT ALREADY OCCUPIED
TYMX14=601215	;SYSNDX SLOT NOT CURRENTLY OCCUPIED
TYMX15=601216	;ILLEGAL AUX TTY OR IN BAD STATE
TYMX16=601217	;AUX TTY NOT QUEUED BY JOB
TYMX17=601220	;NO JOBS QUEUED
TYMX18=601221	;BAD AUX CIRCUIT FUNCTION SPECIFIED
TYMX19=601222	;AUX CIRCUIT NOT IN BLOCK I/O MODE
TYMX20=601223	;AUX CIRCUIT NOT IN NON-BLOCK MODE
TYMX21=601224	;I/O BLOCK DOES NOT START ON PAGE BOUNDARY
TYMX22=601225	;ILLEGAL I/O BLOCK WORD COUNT
TYMX23=601226	;TTY IS NOT ASSOCIATED WITH AN AUX CIRCUIT
TYMX24=601227	;BLK IO XFER ERROR, ERROR BITS IN LH OF BLOCK

hil