

SERIES-III PL/M-86 V2.0 COMPILATION OF MODULE TS
 OBJECT MODULE PLACED IN :F1:TSTRUC.OBJ
 COMPILER INVOKED BY: PLM86.86 :F1:TSTRUC.P86 OPTIMIZE(3) XREF SET(F1) DEBUG

\$TITLE('TCL STRUCTURES 10/26 14:55')
 \$COMPACT DEBUG

*** WARNING 10 IN 1 (LINE 2): RESPECIFIED PRIMARY CONTROL, IGNORED

\$IF f7
 \$INCLUDE (:F7:cpyrt.dcp)
 \$ELSE
 \$INCLUDE (:F1:cpyrt.dcp)

= /* Intel Corporation Proprietary Information.
 = This listing is supplied under the terms of a
 = license agreement with Intel Corporaton and
 = may not be copied nor disclosed except in
 = accordance with the terms of that agreement. */

\$ENDIF

1 ts: DO;
 2 1 DECLARE w WORD;

\$IF f7
 \$INCLUDE (:F7:TCLGBL.INC)
 \$ELSE
 \$INCLUDE (:F1:TCLGBL.INC)

/* Global Literals */
 04/15/82 */

3 1 /* TCL Global Literals
 DECLARE
 max\$send\$seg LITERALLY '07H', /* max no of back-to-back segs that one connection */
 /* can send at a time */
 tcl\$header\$len LITERALLY '20', /* bytes in tcl header */
 /* ETHERNET-SPECIFIC VALUES */
 dll\$header\$len LITERALLY '14', /* bytes in dll header */
 min\$pkt\$len LITERALLY '46', /* minimum total pkt len - bytes */
 max\$seg\$data\$len\$lit LITERALLY '1480', /* (1480) max no. of client bytes in seg */
 tcl\$protocol\$code LITERALLY '5001H', /* DLLCONNECT user type field */
 tcl\$protocol\$code\$rev LITERALLY '0150H', /* packet header user type field */
 /* Misc values */
 tcl\$mip\$port LITERALLY '4', /* mip port for IP\$IN\$MBX */
 log\$rb\$mip\$port LITERALLY '5', /* debugging: mip port for logging */
 mip\$echo\$port LITERALLY '7', /* mip port of on-bd tcl echo server */
 tcl\$version\$lit LITERALLY '101H', /* Version of this TCL for seg header */
 def\$net\$id\$lit LITERALLY '1', /* default Network ID: "this network" */
 on\$bd\$tcl\$echo\$port LITERALLY '7', /* TCL port of on-board tcl echo server */
 true LITERALLY 'OFFH',

```
= false          LITERALLY '0',
= forever        LITERALLY 'WHILE true',

= Timeout$increase$state LITERALLY '1', /* In this state the retransmission timeout
=                                     is rapidly increased */
= Timeout$steady$state   LITERALLY '0'; /* In this state the timeout is
=                                     slowly decreased. This should not be
=                                     changed, it is the initial state since
=                                     a cdb is initialised to zero */

$ENDIF
$SUBTITLE('      TCLCDB.INC      Connection Data Base')
```


TCLCDB.INC

Connection Data Base

```
= pending$rcv$data WORD, /* 56 = 86t no. of undelivered bytes in last pkt */
= rcv$buf$rej$cnt WORD, /* 58 = 88t no. of times pkt rejected due to no rcv buf */
= ayt$count WORD, /* 5A = 90t no. of times ayt pkts weren't answered */
= /* The Alarm control Blocks */
= data$alarm$cb(2) WORD, /* 5C = 92t Data Alarm Control Block header */
= data$acb$irbtype BYTE, /* 60 = 96t Type byte positioned same as in irb */
= data$acb$flag BYTE, /* 61 = 97t ACB Flag byte (running, expired, clrd) */
= data$acb$rem(10) BYTE, /* 62 = 98t remainder of data acb */

= ctl$alarm$cb(2) WORD, /* 6C =108t Control Alarm Control Block header */
= ctl$acb$irbtype BYTE, /* 70 =112t Type byte positioned same as in irb */
= ctl$acb$flag BYTE, /* 71 =113t ACB Flag byte (running, expired, clrd) */
= ctl$acb$rem(10) BYTE) /* 72 =114t rest of ctl acb */
= /* 7C =124t Total Length */

$ENDIF
;
```

\$SUBTITLE('

TCLSTA.INC

Status Data Block')

```

5 1 DECLARE TCLSTA BASED w
$IF f7
$INCLUDE (:F7:tclsta.INC)
$ELSE
$INCLUDE (:F1:tclsta.INC)
= /* Definition of the fields returned in a Status request buffer 11/03/81 */
=
= STRUCTURE ( /* fields returned on either type of request */
= tcl$state BYTE, /* state of tcl */
= def$abort WORD, /* default abort timeout for connections */
= def$retran$dw DWORD, /* default retransmit timeout */
= def$persist WORD, /* default persistence value */
= cur$max$cdb$ BYTE, /* max number of cdb$ for which there is space avail */
= num$cdb$ BYTE, /* number of connection data bases now allocated*/
= loc$net WORD, /* ID of our own network */
= loc$host(3) WORD, /* ID of this comm board (the local host ID) */
= tot$pkts$rej WORD, /* total no. of rcv packets rejected by this TCL */
= tot$retran$events WORD, /* total number of times retran timer expired */
= tot$rcv$buf$rej WORD, /* ttl no times there was insufficient buf space */
= rtc$dw DWORD, /* real-time-clock: (read-clock units) */
=
= /* fields returned only for cid <> 0 */
= /* NOTE: locport thru remport must be in this order to agree */
= /* with tclcdb.inc declaration - doing MOVW in status$req */
= loc$port WORD, /* the local port number of this connection */
= rem$net WORD, /* ID of remote network */
= rem$host(3) WORD, /* ID of the remote host */
= rem$port WORD, /* the remote port number for this connection */
= loc$cid WORD, /* local half of the connection id */
= rem$cid WORD, /* remote half of the connection ID */
= conn$abort WORD, /* abort timeout value for this connection */
= conn$retran$to$dw DWORD, /* conn retransmit timeout value */
= conn$persist WORD, /* persistence value for this connection */
= conn$state BYTE, /* state of the connection */
= pending$rcv$data WORD, /* no. of bytes of rcv data w/no rcv buf space */
= rcv$buf$rej$cnt WORD, /* number of times there was insuff rcv buf space */
= cbtq$buf$cnt BYTE, /* Number of RBs in transmit queue */
= pcbq$buf$cnt BYTE, /* Number of RBs in Receive buffer queue */
= loc$credit BYTE, /* my$credit: #pkts we said remote guy could send*/
= rem$credit BYTE, /* his$credit: #pkts remote guy said we can send */
= highest$sent WORD, /* highest seg seq number sent (mod 65k)*/
= my$ack$no WORD, /* highest seg seq number we have acked */
= no$confid WORD, /* approximate number of retries since last ack */
= /* was received; value is inversely proportional*/
= /* to confidence that remote tcl and/or client */
= /* is still alive and healthy */
= last$entry BYTE) /* symbolic ref to end of list: no info here */
$ENDIF
;

```


TCLRBO.INC

Request Block/Open

```

7 1 DECLARE TCLRBO BASED w
$IF f7
$INCLUDE (:F7:tclrbo.INC)
$ELSE
$INCLUDE (:F1:tclrbo.INC)
= STRUCTURE( /* Request Block for the TCL Open request 03/31/81 */
= cmx$ptr POINTER, /* 0 = 0 for cmx/mip to use */
= mip$buf$base POINTER, /* 4 = 04T : */
= mip$length WORD, /* 8 = 08T : */
= mip$ids$cid BYTE, /* A = 10T : MIP fields: required for */
= mip$owner$dev$cid BYTE, /* B = 11T : any comm subsystem */
= owner$process$cid WORD, /* C = 12T : */
= req BYTE, /* E = 14T Code for type of request */
= resp BYTE, /* F = 15T reponse code: ok or error type */
= rtn$mip$skt WORD, /* 10 = 16T return address: kaos mbx or MIP socket*/
= link POINTER, /* 12 = 18t to rb tcl keeps for abort signaling */
= CID WORD, /* 16 = 22t returned by open processing */
= loc$port WORD, /* 18 = 24t local port for connection */
= rem$net WORD, /* 1A = 26t remote net id : */
= rem$host(3) WORD, /* 1C = 28t remote host id : socket */
= rem$port WORD, /* 22 = 34t remote port id : */
= persist WORD, /* 24 = 36t no times to ignore remote connect refusal */
= abort$timeout WORD, /* 26 = 38t optional abort timeout value */
= seq WORD) /* 28 = 40t reserved for TCL */
= /* 2A = 42t Total Length */

$ENDIF
;
```

```

$SUBTITLE(' TCLRBS.INC Request Block/Standard')
```

```

8 1 DECLARE TCLRBS BASED w
    $IF f7
    $INCLUDE (:F7:tclrbs.INC)
    $ELSE
    $INCLUDE (:F1:tclrbs.INC)
    = STRUCTURE( /* Request Block for TCL Standard requests 05/29/81 */
    = contents BYTE, /* 0 flag: sendable data/signals here */
    = credit BYTE, /* 1 receive buf credit for this rb */
    = last$seq WORD, /* 2 seq of last seg in RB */
    = /* above 4 bytes hold KAOS ptr when RB on mbx) */
    = mip$buf$base POINTER, /* 4 */
    = mip$length WORD, /* 8 */
    = mip$ids$id BYTE, /* A = 10T */
    = mip$owner$dev$id BYTE, /* B = 11T */
    = internal$process$id WORD, /* C = 12T for failure handler, not SCL process ID */
    = req BYTE, /* E = 14T Code for type of request */
    = resp BYTE, /* F = 15T reponse code: ok or error type */
    = rtn$mip$skt WORD, /* 10 = 16T return address: CMX mbx or MIP socket*/
    = link POINTER, /* 12 = 18T optional chain to another RB*/
    = CID WORD, /* 16 = 22t returned by open processing */
    = first$seq WORD, /* 18 = 24t reserved for TCL: seq of 1st seg in RB */
    = client$use WORD, /* 1A = 26t Reserved for client Use (SCL) */
    = buf$len WORD, /* 1C = 28t total no of client data bytes */
    = num$blks BYTE, /* 1E = 30t number of data blocks */
    = vb BYTE) /* 1F = 31t start of Variable-length Buffer (this byte */
    = /* used only as symbolic ref for variable ptr ) */
    = /* 20 = 32t Total Length */
    $ENDIF
    ;
9 1 END ts;

```


DEFN	ADDR	SIZE	NAME, ATTRIBUTES, AND REFERENCES
3			DEFNETIDLIT. LITERALLY '1'
3			DLLHEADERLEN LITERALLY '14'
3			FALSE. LITERALLY '0'
3			FOREVER. LITERALLY 'WHILE true'
3			LOGRBMIPPORT LITERALLY '5'
3			MAXSEGDATALENLIT LITERALLY '1480'
3			MAXSENDSEG LITERALLY '07H'
3			MINPKTLEN. LITERALLY '46'
3			MIPECHOPORT. LITERALLY '7'
3			ONBDTCLECHOPORT. LITERALLY '7'
4	0000H	124	TCLCDBS STRUCTURE BASED(W)
	0000H	1	STATE. BYTE
	0001H	1	OWNERDEVICE. BYTE
	0002H	2	OWNERPROCESSID WORD
	0004H	2	LOCCID WORD
	0006H	2	LOCPORT. WORD
	0008H	2	REMNET WORD
	000AH	6	REMHOST. WORD ARRAY(3)
	0010H	2	REMPORT. WORD
	0012H	2	PERSIST. WORD
	0014H	2	ABORTTOHI. WORD
	0016H	2	REMCID WORD
	0018H	4	RETRANTODW DWORD
	001CH	2	RESERVED WORD
	001EH	2	TIMEDSEQNO WORD
	0020H	4	SEGTRANSTIMEDW DWORD
	0024H	4	CUMRETRANDW. DWORD
	0028H	2	PERSISTCNT WORD
	002AH	4	CBTQHDR. POINTER
	002EH	4	PCBQHDR. POINTER
	0032H	4	DEFSTATUSP POINTER
	0036H	2	MYACKNO. WORD
	0038H	2	SEEN WORD
	003AH	1	MYCREDIT BYTE
	003BH	1	CURBLKINDEX. BYTE
	003CH	2	CBDATAINDEX. WORD
	003EH	2	RCVBYTESCONSUMED WORD
	0040H	2	CURBLKLENLEFT. WORD
	0042H	2	HISACKNO WORD
	0044H	2	NEXTTRANSMIT WORD
	0046H	1	CLOSEDREASON BYTE
	0047H	1	HISCREDIT. BYTE
	0048H	2	HIGHESTSENT. WORD
	004AH	1	CBTQBUFCNT BYTE
	004BH	1	PCBQBUFCNT BYTE
	004CH	2	PKTSREJ. WORD
	004EH	2	PKTSRETRAN WORD
	0050H	2	NOCONFID WORD
	0052H	2	LASTNOCONFID WORD
	0054H	1	RETRANSMITSTATE. BYTE
	0055H	1	SENDFLAG BYTE
	0056H	2	PENDINGRCVDATA WORD

	0058H	2	RCVBUFREJCNT . . .	WORD
	005AH	2	AYTCOUNT . . .	WORD
	005CH	4	DATAALARMCB . . .	WORD ARRAY(2)
	0060H	1	DATAACBIRBTYPE . . .	BYTE
	0061H	1	DATAACBFLAG . . .	BYTE
	0062H	10	DATAACBREM . . .	BYTE ARRAY(10)
	006CH	4	CTLALARMCB . . .	WORD ARRAY(2)
	0070H	1	CTLACBIRSTYPE . . .	BYTE
	0071H	1	CTLACBFLAG . . .	BYTE
	0072H	10	CTLACBREM . . .	BYTE ARRAY(10)
3			TCLHEADERLEN . . .	LITERALLY '20'
3			TCLMIPPORT . . .	LITERALLY '4'
3			TCLPROTOCOLCODE . . .	LITERALLY '5001H'
3			TCLPROTOCOLCODEREV . . .	LITERALLY '0150H'
7	0000H	42	TCLRBO . . .	STRUCTURE BASED(W)
	0000H	4	CMXPTR . . .	POINTER
	0004H	4	MIPBUFBASE . . .	POINTER
	0008H	2	MIPLength . . .	WORD
	000AH	1	MIPIDSID . . .	BYTE
	000BH	1	MIPOWNERDEVID . . .	BYTE
	000CH	2	OWNERPROCESSID . . .	WORD
	000EH	1	REQ . . .	BYTE
	000FH	1	RESP . . .	BYTE
	0010H	2	RTNMIPSKT . . .	WORD
	0012H	4	LINK . . .	POINTER
	0016H	2	CID . . .	WORD
	0018H	2	LOCPORT . . .	WORD
	001AH	2	REMNET . . .	WORD
	001CH	6	REMHST . . .	WORD ARRAY(3)
	0022H	2	REMPORT . . .	WORD
	0024H	2	PERSIST . . .	WORD
	0026H	2	ABORTTIMEOUT . . .	WORD
	0028H	2	SEQ . . .	WORD
8	0000H	32	TCLRBS . . .	STRUCTURE BASED(W)
	0000H	1	CONTENTS . . .	BYTE
	0001H	1	CREDIT . . .	BYTE
	0002H	2	LASTSEQ . . .	WORD
	0004H	4	MIPBUFBASE . . .	POINTER
	0008H	2	MIPLength . . .	WORD
	000AH	1	MIPIDSID . . .	BYTE
	000BH	1	MIPOWNERDEVID . . .	BYTE
	000CH	2	INTERNALPROCESSID . . .	WORD
	000EH	1	REQ . . .	BYTE
	000FH	1	RESP . . .	BYTE
	0010H	2	RTNMIPSKT . . .	WORD
	0012H	4	LINK . . .	POINTER
	0016H	2	CID . . .	WORD
	0018H	2	FIRSTSEQ . . .	WORD
	001AH	2	CLIENTUSE . . .	WORD
	001CH	2	BUFLen . . .	WORD
	001EH	1	NUMBLKS . . .	BYTE
	001FH	1	VB . . .	BYTE
6	0000H	41	TCLSEG . . .	STRUCTURE BASED(W)
	0000H	4	KAOSMSGHDR . . .	POINTER
	0004H	2	BUFLen . . .	WORD
	0006H	6	DLDEST . . .	WORD ARRAY(3)

	000CH	6	DLSOURCE	WORD ARRAY(3)
	0012H	2	DLTYPE	WORD
	0014H	2	TCLVERSION	WORD
	0016H	2	DESTPORT	WORD
	0018H	2	SOURCEPORT	WORD
	001AH	2	DESTCID	WORD
	001CH	2	SOURCECID	WORD
	001EH	2	SEGSEQNO	WORD
	0020H	2	SEGACKNO	WORD
	0022H	2	SEGDATALEN	WORD
	0024H	2	CTL	WORD
	0026H	2	CHECKSUM	WORD
	0028H	1	SEGDATA	BYTE ARRAY(1)
5	0000H	69	TCLSTA	STRUCTURE BASED(W)
	0000H	1	TCLSTATE	BYTE
	0001H	2	DEFABORT	WORD
	0003H	4	DEFRETRANDW	DWORD
	0007H	2	DEFPERSIST	WORD
	0009H	1	CURMAXCDBS	BYTE
	000AH	1	NUMCDBS	BYTE
	000BH	2	LOCNET	WORD
	000DH	6	LOCHOST	WORD ARRAY(3)
	0013H	2	TOTPKTSREJ	WORD
	0015H	2	TOTRETRANEVENTS	WORD
	0017H	2	TOTRCVBUFREJ	WORD
	0019H	4	RTCDW	DWORD
	001DH	2	LOCPORT	WORD
	001FH	2	REMNET	WORD
	0021H	6	REMHOST	WORD ARRAY(3)
	0027H	2	REMPORT	WORD
	0029H	2	LOCCID	WORD
	002BH	2	REMCID	WORD
	002DH	2	CONNABORT	WORD
	002FH	4	CONNRETRANTODW	DWORD
	0033H	2	CONNPERIST	WORD
	0035H	1	CONNSTATE	BYTE
	0036H	2	PENDINGRCVDATA	WORD
	0038H	2	RCVBUFREJCNT	WORD
	003AH	1	CBTQBUFCNT	BYTE
	003BH	1	PCBQBUFCNT	BYTE
	003CH	1	LOCCREDIT	BYTE
	003DH	1	REMCREDIT	BYTE
	003EH	2	HIGHESTSENT	WORD
	0040H	2	MYACKNO	WORD
	0042H	2	NOCONFID	WORD
	0044H	1	LASTENTRY	BYTE
3			TCLVERSIONLIT	LITERALLY '101H'
3			TIMEOUTINCREASESTATE	LITERALLY '1'
3			TIMEOUTSTEADYSTATE	LITERALLY '0'
3			TRUE	LITERALLY 'OFFH'
	0000H		TS	PROCEDURE STACK=0000H
2	0000H	2	W	WORD

MODULE INFORMATION:

CODE AREA SIZE = 0000H 0D
CONSTANT AREA SIZE = 0000H 0D
VARIABLE AREA SIZE = 0002H 2D
MAXIMUM STACK SIZE = 0000H 0D
299 LINES READ
1 PROGRAM WARNING
0 PROGRAM ERRORS

END OF PL/M-86 COMPILATION