C Language Structures

This chapter provides C language definitions of Cisco IOS for S/390 data structures for use with the basic C library. It includes these sections:

- Introduction to C Language Data Structures

  Provides a brief description of the C library data structures and shows their correspondence to C language structures.

- C Language Structures

  Includes the C language definition for all C language structs (structures) provided with Cisco IOS for S/390.

Introduction to C Language Data Structures

Data structures that are provided by the application program as arguments of the API transport service functions are defined in this chapter and Chapter 4, Socket Library Include Files. This chapter defines these data structures as used by application programs written in C language using the basic C library.

Correspondence Between dsects and C Language Structures

There is a one-to-one correspondence between the API assembler language dsects and C language structures. This table shows the correspondence between dsect names and structure names:

API Data Structure	DSECT Name	Structure Name
Application Program Control Block	APCB	apcb
APCB Exit List	APCBXL	apcbxl
Transport Endpoint Error Message	TEM	tem
Transport Service Information Block	TIB	tib
Transport Protocol Address	TPA	tpa
Transport Service Parameter List	TPL	tpl
Transport Protocol Options	TPO	tpo
Transport Endpoint State Word	TSW	tsw
Transport Endpoint User Block	TUB	tub
Transport Endpoint Exit List	TXL	txl
Transport Endpoint Exit Parameters	TXP	txp

Data structures are listed in detail in the following sections.

C Language Structures

The definitions and declarations in this section are written to ANSI C specifications, and can be used to compile programs with the IBM C/370 and SAS/C compilers. They are contained in the include file api.h.

Include files are installed on the local system as members of a partitioned data set. This data set should be included in the SYSLIB DD concatenation when the application program is compiled. The data set name is determined during installation unless changed by the local system programmer. The API member in this data set corresponds to the api.h include file.

Each page that pertains to a particular include file has the name of the file in the upper outside corner.

apcb

apcb (Application Program Control Block):

C Language Definition

```
/*
* defines for length of variables used in apcb
*/
#define APCBTALN             4    /* length of control block id*/
#define APCBDGLN             2    /* length of diagnostic code  */
#define APCBAMLN             4    /* length of subsys id */
#define APCBAPLN             8    /* length of application id*/
#define APCBPSLN             8    /* length of password */
/*
* definition of the application program control block used at
 * AOPEN time
*/
struct apcb
{
    char            apcbtag [ APCBTALN ];   /* control block id */
    unsigned long   apcbsl;          /* control block length */
    unsigned char   apcbam;          /* access method & vers */
    unsigned char   apcbflag;        /* flag byte */
    unsigned char   apcboptc;        /* option code */
    unsigned char   apcbrsv1;        /* reserved */
    unsigned char   apcbenvr;        /* language envir. code*/
    unsigned char   apcberrc;        /* error code */
    unsigned char   apcbdgnc [ APCBDGLN ];  /* diagnostic code  */
    char            *apcbamcb;        /* acc. mthd control blk */
    char       *apcbamcv;      /* acc. mthd cvt addr.      */
    char          *apcbamtv;           /* AM TUAS xfer vector */
    char           apcbamid [ APCBAMLN ]; /* acc. mthd subsys id*/
    char          *apcbrsv2;        /* reserved         */
    struct txl    *apcbexls;        /* appl lvl exit list addr */
    unsigned long  apcbactx;        /* appl lvl context var */
    unsigned long  apcbectx;        /* envir lvl context var */
    char           apcbappl [ APCBAPLN ];  /* application id   */
    char           apcbpswd [ APCBPSLN ];  /* appl password  */
};
/*
* apcbtag must contain this string
*/
#define APCBIDENT            "APCB"
/*
* access method and version ( apcbam )
*/
#define APCBAMSK          0xF0        /* access method id */
#define APCBATLI          0x10        /* transport layer interface */
#define APCBAMAX          APCBATLI    /* maximum access method  */
#define APCBAVER          0x0F        /* access method version */

/*
* flag byte ( apcbflag )
*/

#define APCBFSTP        0x80      /* applid is stpnam frm tiot */
#define APCBF31B        0x40      /* AMODE=31 */
#define APCBFANY        0x20      /* RMODE=ANY */
#define APCBFOPN        0x10      /* apcb is open */
#define APCBFERR        0x08      /* permanent error flag  */
#define APCBFTRM        0x04      /* task termina. in progress */
#define APCBFBSY      0x01         /* open/close in progress */
```

```
/*
* option code ( apcboptc )
*/

#define APCBOTRC       0x80       /* optcd=notrace | trace */
#define APCBOGTF       0x40       /* optcd=nogtf|gtf
/*
* language environment code ( apcbenvr )
*/

#define APCBASM        0            /* assembler language  */
#define APCBIBMC       1            /* ibm c  */
#define APCBSASC       2            /* sas c  */
#define APCBPLI        3            /* pli  */
#define APCBCOBL       4            /* cobol   */
#define APCBFORT       5            /* fortran  */
#define APCBEMAX       APCBFORT    /* maximum environment code  */

/*
* error code ( apcberrc )
*/

#define APCBECFG       1        /* subsystem not configured */
#define APCBEACT       2        /* subsystem not active */
#define APCBERDY       3        /* subsystem not initialized */
#define APCBESTP       4        /* subsystem is stopping */
#define APCBEDRA       5        /* subsystem is draining */
#define APCBEVCK       6        /* apcb validity check error */
#define APCBELER       7        /* internal logic error  */
#define APCBEPRB       8        /* not issued from PRB */
#define APCBEOPN       9        /* apcb already opened */
#define APCBECLS       10           /* apcb already closed */
#define APCBEBSY       11          /* apcb busy w/aopen/aclose */
#define APCBEPER       12          /* apcb has permanent error */
#define APCBECVT       13          /* access mthd cvt not avail */
#define APCBEMEM       14          /* insuff. memory avail */
#define APCBEENV       15          /* cannot initialize envir */
#define APCBEBEG       16          /* cannot estab api session */
#define APCBEVER       17          /* invalid access mthd vers */
#define APCBEOPT       18          /* invalid/unsupported opt */
#define APCBEDUP       19          /* dup session for this am */
#define APCBEAMD       20          /* AMODE inconsist w/AOPEN */
#define APCBETRV       21          /* AMTV validity check error */
#define APCBEEND       22          /* Cannot release API sess  */
#define APCBERMX       APCBEEND    /* max apcb error code  */

#define APCBLEN        sizeof(apcb) /* length of apcb */
```

apcbxl

apcbxl (Application Program Control Block Exit List):

C Language Definition

```
/*
* definition of the application program control block exit list
*/

struct apcbxl
{
    unsigned long    apcbxlen;    /* total len of exit list */
    struct txl      *apcbxlst;    /* exit rtn entry pts list  */
};
```

tem

tem (Transport Endpoint Error Message):

C Language Definition

```
#define TEMSGTAGLEN     4          /* length of control block  */
#define TEMSGIDLEN      8          /* length of message ID     */
#define TEMSGBDYLEN     26         /* length of message body   */
#define TEMSGLEN        34         /* length of message        */

/*
* terror generated message
*/

union temmsg
{
    char      temmsgtxt [ TEMSGLEN ];       /* msg txt len (1st)*/
    struct
    {
         char     temsgid [ TEMSGIDLEN ];    /* message id */
         char     temsgbdy [ TEMSGBDYLEN ];  /* message body */
    } temsgparts;
};

/*
* terror generated message parameter list return value
*
* a pointer to this structure is returned when a terror()
* function completes successfully.
*/

struct tem
{
     char          temtag[ TEMSGTAGLEN ];    /* cntrl block tag */
     unsigned long temsl;         /* subpool and length*/
     char          temsglen [2];  /* msg length + 4 (1st line) */
     char          temmcsf1;      /* mcs flag byte #1 */
     char          temmcsf2;      /* mcs flag byte #2 */
     union temmsg  temmsg;        /* first line */
     unsigned short  temdesc;       /* descriptor codes  */
     unsigned short  temrout;       /* routing codes  */
     char          temsgtyp [2];  /* mlwto line typ (1st line) */
     char          temarea;       /* mlwto area id  */
     char          temnline;      /* mlwto number of lines  */
     short         temmllen;      /* mlwto line length + 4 */
     short         temmltyp;      /* mlwto line type */
     char          temmltxt;      /* mlwto line text */
                                  /* element size is variable */
};
#define TEMLEN           sizeof ( struct tem )
```

tib

tib (Transport Service Information Block):

C Language Definition

```
#define TIBSYSIDLEN       4    /* max length of subsystem name */
#define TIBSVCIDLEN       8    /* max length of service name  */
#define TIBHOSTNAMELEN    64   /*max length of local host name */
/*
* structure returned by tinfo()function
*/
struct tib
{
     unsigned char   tibtsdom;     /* transport service domain  */
     unsigned char   tibtstyp;     /* transport service type  */
     unsigned char   tibtschr;     /* transport service char. */
     unsigned char   tibtsopt;     /* transport service options */
     char        tibsysid [ TIBSYSIDLEN ];   /* TP subsys name */
     char        tibsvcid [ TIBSVCIDLEN ];  /* TP service name */
     int             tibproto;    /* transport protocol number  */
     int             tibqlstn;    /* max size of listen queue */
     int             tibqsend;    /* max size of send queue */
     int             tibqrecv;    /* max size of receive queue */
     int             tibltsnd;    /* max size of send TIDU */
     int             tibltrcv;    /* max size of receive TIDU */
     int             tiblsend;    /* max size of send buffer*/
     int             tiblrecv;    /* max size of recv buffer */
     int             tibladdr;  /* max size of protocol address */
     int             tibloptn;  /* max size of protocol options */
     int                    tibltsdu;       /* max size of TSDU */
     int             tiblxpdt;  /* max size of ETSDU  */
     int             tiblconn;  /* max size of connect data */
     int             tibldisc;  /* max size of disconnect data */
     int            tiblinfo;   /* max size of information unit */
};
/*
* defines for transport service domain
*/
#define TIBDINET         2     /* Internet domain */
#define TIBDACP          4     /* ACP inet domain */


/*
* transport service type
*/

#define TIBTCOTS       1     /* connection-mode service */
#define TIBTCLTS       2     /* connectionless-mode service */


/*
* transport service characteristics
*/

#define TIBCTSDU      0x10   /* message boundaries preserved*/
#define TIBCXPDT      0x08   /* expedited data supported */
#define TIBCOPTN      0x04   /* user-settable options supported */
#define TIBCCOND      0x02   /* connect with user data */
#define TIBCDISD      0x01   /* disconnect with user data */

/*
* transport service options
*/
```

```
#define TIBOASSO     0x80   /* datagram associations supported */
#define TIBOSCND     0x40   /* secondary information available */
#define TIBOSTAT     0x20   /* statistical info available */
#define TIBCRLSE     0x10   /* orderly release supported */

/*
* length of the TIB
*/

#define TIBLEN            sizeof ( struct tib )
```

tpa

tpa (Transport Protocol Address):

C Language Definition

```
/*
* transport protocol address in the internet domain
*/

struct tpainet
{
    unsigned short    tpainetd;      /* internet domain */
    unsigned short    tpainett;      /* TCP port number */
    unsigned long     tpainetn;      /* IP host address */
    };

/*
* length of transport protocol address
*/

#define TPALEN    sizeof ( struct tpainet )
```

tpl

tpl (Transport Service Parameter List):

C Language Definition

```
#define TPLSVCIDLEN    8

/*
* internal ECB, external ECB address, exit routine address union
*/

union tplecbexit
{
    union
    {
        unsigned long tpliecb;   /* internal ECB */
        unsigned long *tplxecb;  /* external ECB address */
        } tplecb;
        void         ( *tplexit )();  /* exit routine address */
} ;

/*
* TPL option code structure
*/

union tploptcd
{
    unsigned long    tploptcdl;               /* option codes */
    struct
    {
        unsigned char    tplopcd1;            /* option code #1 */
        unsigned char    tplopcd2;            /* option code #2 */
        unsigned char    tplopcd3;            /* option code #2 */
        unsigned char    tplopcd4;            /* option code #2 */
    } tploptcds;
} ;
/*
* TPL return code structure
*/

union tplrtncd
{
    unsigned long        tplrtncdl;           /* return codes */
    struct
    {
        unsigned char    tplactcd;       /* recovery action code*/
        unsigned char    tplerrcd;       /* specific error code */
        unsigned short   tpldgncd;       /* diag & sense codes */
        } tplrtncds;
} ;
/*
* TPL fixed-length parameters structure
*/

struct tplparm
{
    union
    {
        int         tplqlstn;          /* listen queue length */
        int         tplseqno;          /* sequence number */
        unsigned char  *tpltcb;        /* tcb address */
    } tplparm1;
    union
    {
```

```
        unsigned long    tplnewep;       /* new endpoint */
        unsigned char   *tplascb;        /* ascb address */
        int              tplcount;       /* transfer byte count */
    } tplparm2;
    union
    {
        unsigned char   *tpluser;     /* tub or acee address */
        unsigned long    tpldiscd;    /* disconnect reason code */
        unsigned long    tpldgerr;    /* datagram error code */
        unsigned long    tplstate;    /* old endpoint state */
        unsigned long    tplxcnt;     /* xdata residual count */

    } tplparm3;
} ;
/*
* TPL variable-length protocol address parameter
*/

struct tpladdr
{
    unsigned char    *tpladbuf;       /* parameter address */
    int               tpladlen;       /* parameter length  */
} ;

/*
* TPL variable-length user data parameter
*/

struct tpldata
{
    unsigned char     *tpldabuf;       /* parameter address */
    int               tpldalen;        /* parameter length */
} ;

/*
* TPL variable-length protocol options parameter
*/

struct tploptn
{
    unsigned char     *tplopbuf;       /* parameter address */
    int               tploplen;        /* parameter length */
} ;


/*
* TPL variable-length parameters structure
*/

struct tplbufp
{
    struct tpladdr     tpladdr;         /* protocol address */
    struct tpldata     tpldata;         /* user data */
    struct tploptn     tploptn;         /* protocol options */
} ;

/*
* TPL structure specific to TOPEN
*/

struct tplopen
{
    unsigned char     tpldom;          /* communication domain */
    unsigned char     tploflag;        /* open flags */
    union
```

```
    {
        unsigned short     tpltype;     /* transport serv type */
        unsigned short     tplproto;    /* transport proto num */

    }   tplservc;
    struct apcb    *tplapcbp;       /* address of APCB */
    char            tplsvcid [ TPLSVCIDLEN ];/* TP service name */
    struct txlh    *tplexlst;       /* address of exit list */
    unsigned long   tplucntx;       /* word of user context */
} ;

/*
* Transport Service Parameter List (TPL) structure
*/

struct tpl
{
    unsigned char     tplident;     /* control block identifier */
    unsigned char     tplfnccd;        /* function code */
    unsigned char     tplactiv;        /* active semaphore */
    unsigned char     tplflags;        /* flags used by AP */
    union
    {
        unsigned long           tplepid;     /* endpoint id */
        unsigned long           tpltcep;     /* TCEP address */
    } tplep;
    union tplecbexit   tplecbexit;  /* ECB or exit routine addr */
    union tploptcd     tploptcd;    /* option codes  */
    union tplrtncd     tplrtncd;    /* return codes  */
    struct tplparm     tplparm;     /* fixed-length parameters */
    union
    {
        struct tplbufp     tplbufp    /* variable-length parms */
        struct tplopen     tplopen;   /* tpl open parameters */
    } tplopbfp;
};

/*
* control block ID values (tplident)
*/

#define TPLIDSTD 0xEA               /* standard long format */
#define TPLIDSHT 0xEB               /* short format */

/*
* defines for tpl function codes ( tplfnccd )
*/

#define TFORG1          0        /* origin for std functions */
#define TFACCEPT        1        /* accept connection request */
#define TFADDR          2        /* get address information */
#define TFBIND          3        /* bind protocol address */
#define TFCLEAR         4        /* confirm disconnect*/
#define TFCLOSE         5        /* close endpoint */
#define TFCONFRM        6        /* confirm connection req */
#define TFCONNCT        7        /* initiate connection req */
#define TFDISCON        8        /* initiate disconnect */
#define TFINFO          9        /* get transport service info*/
#define TFLISTEN        10       /* listen for connection req */
#define TFOPEN          11       /* open an endpoint */
#define TFOPTION        12       /* negotiate options  */
#define TFRECV          13       /* receive data */
#define TFRECVER        14       /* receive datagram error  */
#define TFRECVFR        15       /* receive datagram   */
#define TFREJECT        16       /* reject connection request */
```

```
#define TFRELACK        17       /* confirm orderly release */
#define TFRELESE        18       /* initiate orderly release */
#define TFRETRCT        19       /* retract a listen request */
#define TFSEND          20       /* send data*/
#define TFSENDTO        21       /* send datagram */
#define TFUNBIND        22       /* unbind protocol address */
#define TFUSER          23       /* associate user id */
#define TFMAX1          TFUSER

#define TFORG2          128      /* origin for ctl functions */
#define TFCHECK         129      /* check tpl for completion */
#define TFERROR         130      /* format error message */
#define TFSTATE         131      /* get endpoint state */
#define TFMAX2          TFSTATE

/*
* flag byte of tpl defines ( tplflags )
*/

#define TPLFCMPL       0x80      /* tpl completed */
#define TPLFCERR       0x40      /* tpl completed with error */
#define TPLFXECB       0x20      /* external ecb being used  */
#define TPLFEXIT       0x10      /* exit being used  */
#define TPLF31B        0x08      /* tpl issued in 31 bit mode */
#define TPLFACPT       0x04      /* accepting on this endpoint */
/*
* option code #1 defines ( tplopcd1 )
*/

#define TOASYNC        0x80      /* OPTCD=SYNC | ASYNC  */
#define TOSHORT        0x40      /* OPTCD=LONG | SHORT  */
#define TOTRUNC        0x20      /* OPTCD=NOTRUNC | TRUNC */
#define TONEGOT        0x10      /* OPTCD=NONEGOT | NEGOT */
#define TONOBLOK       0x04      /* OPTCD=BLOCK | NOBLOCK  */

/*
* option code #2 defines ( tplopcd2 )
*/

#define TOMORE         0x80      /* OPTCD=NOMORE | MORE */
#define TOEXPDTE       0x40      /* OPTCD=NORMAL | EXPEDITE */
#define TONOTEOM       0x20      /* OPTCD=EOM | NOEOM  */
#define TOABORT        0x10      /* OPTCD=CLEAR | ABORT  */
#define TOINDIR        0x08      /* OPTCD=DIRECT | INDIR  */
#define TODLOCAL       0x04      /*OPTCD=DNOTLOCAL | DLOCAL */
#define TOLOCATE       0x02      /*OPTCD=NOLOCATE | LOCATE  */


/*
* option code #3 defines ( tplopcd3 )
*/

#define TOACEE         0x80      /* OPTCD=TUB | ACEE */
#define TOCIPHER       0x40      /* OPTCD=PLAIN | CIPHER */
#define TOOLD          0x20      /* OPTCD=NEW | OLD */
#define TOASSIGN       0x10      /* OPTCD=USE | ASSIGN */
#define TOREMOTE       0x08      /* OPTCD=LOCAL | REMOTE */
#define TOPASS         0x04      /* OPTCD=DELETE | PASS */

/*
* option code #4 defines ( tplopcd4 )
*/

#define TOINFO         0xC0      /* tinfo option codes */
#define TOPRIMRY       0x00      /* OPTCD=PRIMARY*/
```

```
#define TOSCNDRY       0x80      /* OPTCD=SECNDRY */
#define TOSTATS        0x40      /* OPTCD=STATS */

#define TOOPTION       0x30      /* toption option codes */
#define TODECLAR       0x00      /* OPTCD=DECLARE  */
#define TOVERIFY       0x10      /* OPTCD=VERIFY */
#define TOQUERY        0x20      /* OPTCD=QUERY  */
#define TODFAULT       0x30      /* OPTCD=DFAULT  */

#define TOAPI          0x08      /* OPTCD=TP | API*/

/*
* recovery action codes ( tplactcd )
*/

#define TAOKAY         0     /* successful completion*/
#define TAEXCPTN       4     /* exceptional condition */
#define TAINTEG        8     /* connection/data integrity error */
#define TAENVIRO       12    /* environmental condition */
#define TAFORMAT       16    /* format or specification error */
#define TAPROCED       20    /* sequence or procedure error */
#define TATPLERR       24    /* logic errors with no tpl rtncd */
#define TAUSER         28    /* user-defined action codes */


/*
* conditional completion code ( tplerrcd )
*/
#define TCOKAY       0x00        /* 00: no conditionals */
#define TCVERIFY     0x80        /* 00: options did not verify */
#define TCNEGOT      0x40        /* 00: options negotiated */
#define TCTRUNC      0x20        /* 00: buffer truncated */
#define TCSTOP       0x08        /* 00: subsystem is stopping */

/*
* specific error code ( tplerrcd )
*/
/* TENOEP was made obsolete and replaced by TEDRAIN          */
/* #define TENOEP     1      04: no new endpoints allowed  */
#define TENONEGO      6     /* 04: no negotiation allowed */
#define TENOBLOK      9     /* 04: no blocking allowed */
#define TENOLSTN      10    /* 04: no listen pending */

#define TEPROTO       1     /* 08: protocol error */
#define TEOVRFLO      2     /* 08: buffer overflow */
#define TEDISCON      3     /* 08: disconnect received */
#define TERELESE      4     /* 08: orderly release received */
#define TEOVLAY       5     /* 08: control block overlaid */
#define TEFLOW        9     /* 08: temporary flow control */
#define TERETRCT      10    /* 08: listen retracted */
#define TEPURGED      11    /* 08: request purged for TCLOSE  */

#define TESYSERR      1     /* 12: system error */
#define TESUBSYS      2     /* 12: subsystem error */
#define TENOTCNF      3     /* 12: subsys not config in o/s  */
#define TENOTACT      4     /* 12: subsys not started */
#define TENOTRDY      5     /* 12: subsys not initialized */
#define TEDRAIN       6     /* 12: subsys drained by operator  */
#define TESTOP        7     /* 12: subsystem stopped by open. */
#define TETERM        8     /* 12: subsys abnormally termin */
#define TEUNSUPO      9     /* 12: unsupported option/facility */
#define TEUNSUPF      10    /* 12: unsupported function/serv */
#define TEUNAVBL      11    /* 12: unavailable service/facil */
#define TEUNAUTH      12    /* 12: user unauthorized */
#define TERSOURC      13    /* 12: insufficient resources */
```

```
#define TEINUSE       14    /* 12: TPA in use  */

#define TEBDOPCD      1      /* 16: invalid option code  */
#define TEBDEPID      2      /* 16: invalid endpoint */
#define TEBDXECB      3      /* 16: invalid exit/ecb address */
#define TEBDDOM       4      /* 16: invalid communication dom */
#define TEBDPROT      5      /* 16: invalid transport protocol*/
#define TEBDTYPE      6      /* 16: invalid transport serv type */
#define TEBDXLST      7      /* 16: invalid exit list */
#define TEBDUSER      8      /* 16: invalid user parm*/
#define TEBDACEE      9      /* 16: invalid accessor element */
#define TEBDSQNO      10     /* 16: invalid sequence number */
#define TEBDQLEN      11     /* 16: invalid queue length */
#define TEBDTCB       12     /* 16: invalid tcb address */
#define TEBDASCB      13     /* 16: invalid ascb address */
#define TEBDADDR      14     /* 16: invalid protocol address */
#define TEBDOPTN      15     /* 16: invalid options */
#define TEBDDATA      16     /* 16: invalid data buffer */
#define TEBDTSID      18     /* 16: invalid transport serv id */

#define TESTATE       1     /* 20: invalid state for function */
#define TEINEXIT      2     /* 20: invalid function w/in exit */
#define TEINACTV      3     /* 20: check issued to inact tpl */
#define TEINCMPL      4     /* 20: endpoint has incomplete fnc */
#define TEINDICA      5     /* 20: pending connect indication */
#define TEBUFOVR      6     /* 20: send / recv buffer overrun  */
#define TEREQOVR      7     /* 20: send / recv request overrun  */
#define TENOCONN      8     /* 20: no connection */
#define TENODISC      9     /* 20: no disconnect indication */
#define TEOUTSEQ      10    /* 20: request is out of sequence */
#define TENOERR       11    /* 20: no error indication */
#define TEAMODE       13    /* 20: AMODE conflicts with APCB */
#define TEOWNER       14    /* 20: not opened by this task  */
#define TELISTEN      15    /* 20: listen q full */
#define TEACCEPT      16    /* 20: accepting on this endpoint */

#define TEB4EXIT      1     /* 24: TPL check before exit */
#define TEACTIVE       2    /* 24: TPL is still active  */

#define TEMAXCODE     20    /* MAX error code defined */

/*
* disconnect reasons codes
*/

#define TDTRANTO      1         /* Transmission timeout */
#define TDHOSTUN      2         /* Host unreachable */
#define TDPORTUN      3         /* Port unreachable */
#define TDRABORT      4         /* Remote aborted connection */
#define TDLNIOWN      5         /* Local net I/F down  */
#define TDPROTUN      6         /* Protocol unreachable  */
#define TDACPRR       7         /* ACP connection error   */
#define TDAPIRR       8         /* API connection error  */

/*
* minimum TPL length
*/

#define TPLMIN (sizeof (unsigned char) + sizeof (unsigned char) + \
           sizeof (unsigned char) + sizeof (unsigned char) + \
           sizeof (unsigned long) + sizeof (struct tplecbexit) + \
           sizeof (struct tploptcd)+ sizeof (struct tplrtncd) )
/*
* length of short form TPL ( function-specific )
*/
```

```
#define TLRELACK      TPLMIN
#define TLRELESE      TPLMIN
#define TLUNBIND      TPLMIN
#define TLRETRCT      TPLMIN

#define TLACCEPT      ( TPLMIN + sizeof ( struct tplparm ) )
#define TLCLEAR       ( TPLMIN + sizeof ( struct tplparm ) )
#define TLCLOSE       ( TPLMIN + sizeof ( struct tplparm ) )
#define TLDISCON      ( TPLMIN + sizeof ( struct tplparm ) )
#define TLREJECT      ( TPLMIN + sizeof ( struct tplparm ) )
#define TLUSER        ( TPLMIN + sizeof ( struct tplparm ) )
#define TLADDR        ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) )
#define TLBIND        ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) )
#define TLCONFRM      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) )
#define TLCONNCT      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) )
#define TLLISTEN      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) )
#define TLRECVER      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) )
#define TLINFO        ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) + \
                                 sizeof ( struct tpldata ) )
#define TLRECV        ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) + \
                                 sizeof ( struct tpldata ) )
#define TLRECVFR      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) + \
                                 sizeof ( struct tpldata ) )
#define TLSEND        ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) + \
                                 sizeof ( struct tpldata ) )
#define TLSENDTO      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) + \
                                 sizeof ( struct tpldata ) )
#define TLOPTION      ( TPLMIN + sizeof ( struct tplparm ) + \
                                 sizeof ( struct tpladdr ) + \
                                 sizeof ( struct tpldata ) + \
                                 sizeof ( struct tploptn ) )

/*
* length of standard ( long ) form TPL
*/

#define TPLLEN        ( sizeof ( struct tpl ) + \
                      sizeof ( struct tplbufp ) - \
                      sizeof ( struct tplopen ) )

#define TLOPEN        sizeof ( struct tpl )
#define TPLMAX        sizeof ( struct tpl )

/*
* general return codes ( returned in R15 )
*/

#define TROKAY        0     /* successful cmpl, accepted */
#define TRFAILED      4     /* unsuccessful cmpl, not accepted */
#define TRFATLFC      8     /* invalid function code */
#define TRFATLPL      12    /* fatal tpl error  */
#define TRFATLAM      16    /* fatal access method error */
#define TRFATLAP      20    /* apcb is closed  */
```

```
#define TRUSER        24    /* first user return code */

/*
* topen communication domains
*/
#define TDINET        2     /* Internet domain */
#define TDACP         4     /* ACP internet domain */
#define TDMAX         TDSNA     /* max value for domain  */


/*
* topen open flags
*/

#define TPLOFPRO      0x80    /* protocol number specified  */
#define TPLOFORD      0x40    /* COTS orderly release required  */
#define TPLOFASO      0x20    /* CLTS association required   */

/*
* topen transport service types
*/

#define TTCOTS        1        /* connection-mode service */
#define TTCLTS        2        /* connectionless-mode service  */
#define TTMAX         TTCLTS   /* max value for service type */

/*
* tpopen transport protocol number
*/

#define TPINTTCP      6        /* darpa internet tcp  */
#define TPINTUDP      17       /* darpa internet udp  */
#define TISOTP4       0        /*iso transport class 4 */
#define TLOPEN    sizeof ( struct tpl )
                  /* length of short tpl: topen */
#define TPLMAX    sizeof ( struct tpl )
                  /* maximum tpl length: topen */
```

tpo

tpo (Transport Protocol Options):

C Language Definition

```
/*************************************************************/
/* This structure defines the format of an option when        */
/* issuing a TOPTION request                                 */
/*************************************************************/
#define TPOMAXLEN     320                /* for ifconfg */

struct tpo
{
    short           tpoptlen            /* option length  */
    short           tpoption;           /* option name */
    unsigned char   tpovalue [ TPOMAXLEN ];    /* option value  */
};
/*
* API defined option names
*/

#define TPOAQSND      0              /* max # of sends */
#define TPOAQRCV      1              /* max # of recvs */
#define TPOALSND      2              /* length of send buffer */
#define TPOALRCV      3              /* length of receive buf */

/*
 * ACP defined option names
 */

#define TPOPRWND      1         /* TCP receive window */
#define TPOPKTIM      2         /* TCP Keepalive time  */
#define TPOPKEEP      3         /* TCP Keepalive options */
#define TPOPDNAG      4         /* Defeat Nagle algorithm  */
#define TPOPRTIM      5         /* Full Receive Timeout  */
#define TPOIPOPT      6         /* IP Option Text */
#define TPOSIOAR      7         /* Add Route  */
#define TPOSIODR      8         /* Delete Route */
#define TPOSIFCF      9         /* Interface Config. */
#define TPOSIFLG      10        /* Interface Flags */
#define TPOSIFMT      11        /* Interface MTU */
#define TPOSIFME      12        /* Interface Metric  */
#define TPOSIFNM      13        /* Interface Network Mask */
#define TPOSIFBA      14        /* Ifc. Broadcast Address  */
#define TPOSIFAD      15        /* Interface Address   */
#define TPOSIFEN      16        /* Ifc. Ethernet  Address */
#define TPOSIFNO      17        /* Number of Interfaces */
#define TPOSIFDA      18        /* Ifc. Destination Addr. */
#define TPOIPTTL      19        /* IP Time To Live  */
#define TPOIPTOS      20        /* IP Type Of Service  */
/*
*
*/
#define TPOLEN   sizeof(struct tpo)
```

tsw

tsw (Transport Endpoint State Word):

C Language Definition

```
struct tsw
{
    unsigned char      tswflags;
    unsigned char      tswpfunc;
    unsigned short     tswstate;
};
/*
* tswflags defines
*/

#define TSWFCHNG       0x80       /* state is changing  */
#define TSWFACPT       0x40       /* accepting to this endpoint */

/*
* tswpfunc defines (pending functions)
*/

#define TSWPFCLS       0x80       /* TCLOSE  */
#define TSWPFDIS       0x40       /* TDISCONN, TCLEAR, TRETRACT */
#define TSWPFREL       0x20       /* TRELEASE  */
#define TSWPFACK       0x10       /* TRELACK  */
#define TSWPFCON       0x08       /* connection establishment */
#define TSWPFLCL       0x04       /* local endpoint management  */
#define TSWPFOPN       0x01       /* TOPEN */
#define TSWPFRCV       0x00       /* TRECV */
#define TSWPFSND       0x00       /* TSEND  */
#define TSWPFDGM       0x00      /* TSENDTO, TRECVFRM, TRECVERR */

/*
* defines for tswstate
*/

#define TSCLOSED       0          /* closed non-existent */
#define TSOPENED       1          /* opened but not bound */
#define TSDSABLD       2          /* bound and disabled */
#define TSENABLD       3          /* bound and enabled */
#define TSINCONN       4          /* connect indication pending */
#define TSOUCONN       5          /* connection in progress */
#define TSCONNCT       6          /* connected or associated */
#define TSINRLSE       7          /* release indication pending */
#define TSOURLSE       8          /* release in progress*/
#define TSMAX          TSOURLSE   /* max value for tswstate */

/*
* length of the TSW
*/

#define TSWLEN    sizeof ( struct tsw )
```

tub

tub (Transport Endpoint User Block):

C Language Definition

```
#define TUBUIDLEN    9
#define TUBGRPLEN    9
#define TUBPWDLEN    9
struct tubuids
{
    unsigned char  tubuidl;                /* len of user id */
    char           tubuidc [ TUBUIDLEN-1 ];  /* user id string */
} ;

struct tubgrps
{
    unsigned char tubgrpl;         /* len of group id */
    char          tubgrpc [ TUBGRPLEN-1 ];  /* group id string*/
} ;

struct tubpwds
{
    unsigned char  tubpwdl;     /* len of password */
    char           tubpwdc [ TUBPWDLEN-1 ];    /* user password */
} ;
/*
* definition of the TUB
*/
#if 0
#define TUBSTRUCT       1         /* allow less convenient tub */
#endif
#ifndef TUBSTRUCT

struct tub
{
    union
    {
        unsigned char      tubuid [ TUBUIDLEN ];
        struct tubuids     tubuids;
    } tubuidu;
    union
    {
        unsigned char      tubgrp [ TUBGRPLEN ];
        struct tubgrps     tubgrps;
    } tubgrpu;
    union
    {
        unsigned char      tubpwd [ TUBPWDLEN ];
        struct tubpwds     tubpwds;
    } tubpwdu;
};
#else
/*
* length of the TUB
*/
#define TUBLEN             sizeof ( struct tub )
```

txl

txl (Exit List Structure): C Language Definition

```
/*
* Common header for txls used with AOPEN and TOPEN
*/

struct txlh
{
    int     txllenxl;               /* length of exit list */
    void    ( *txlconn )();         /* connect indication */
    void    ( *txlconf )();         /* confirm indication */
    void    ( *txldata )();         /* data indication */
    void    ( *txlxdata )();        /* expedited data indication */
    void    ( *txldgerr )();        /* datagram error indication */
    void    ( *txldisc )();         /* disconnect indication */
    void    ( *txlrelse )();        /* orderly release indication*/
    void    ( *txlresvd1 )();       /* reserved for future use */
    void    ( *txlresvd2 )();       /* reserved for future use */
    void    ( *txlresvd3 )();       /* reserved for future use */
    void    ( *txltpend )();        /* transport provider end */
    void    ( *txlresvd4 )();       /* reserved for future use */
    void    ( *txlresvd5 )();       /* reserved for future use */
    void    ( *txlresvd6 )();       /* reserved for future use */
};

/*
* length of TXLH
*/

#define TXLHLEN    sizeof ( struct txlh )
/*
* definition of the exit list structure used by the TEXLST macro
*/

struct txl
{
    struct txlh         txlh;        /* protocol exit list */
    struct
    {
        int      ( *txlsynad )();         /* physical errors */
        int      ( *txllerad )();         /* logic errors */
    } txlerror;
    void    ( *txlapend )();         /* API subsys end  */
    void    ( *txlrsvd7 )();         /* reserved for future */
    void    ( *txlrsvd8 )();         /* reserved for future */
};

/*
* length of the TXL
*/

#define TXLLEN    sizeof ( struct txl )
```

txp

txp (Transport Exit Parameter List):

C Language Definition

```
struct txp
{
    unsigned short        txptype;        /* exit type */
    unsigned short        txprsvd;        /* resrvd future use */
    union
    {
        unsigned long     txpapcb;        /* APCB pointer  */
        unsigned long     txpepid;        /* end point id */
        unsigned long     txptcep;        /* TCEP address */
    } txpep;
    void        ( *txpexit )();    /* exit routine entry point */
    union
    {
        unsigned long      txpreasn;   /* TPEND reason code */
        unsigned long      txpevent;   /* protocl event code */
        struct tpl        *txptpl;     /* TPL pointer  */
    }   txpparm;
    unsigned long         txpacntx;     /* application context */
    unsigned long         txpucntx;     /* user context */
    unsigned long         txpecntx;     /* environment context */
};
/*
* exit types
*/
#define TXPTPROT      1           /* protocol event exit */
#define TXPTCMPL      2           /* endpoint completion exit */
#define TXPTPEND      3           /* TP end exit */
#define TXPTSYNC      4           /* synchronous error exit */
#define TXPAPEND      5           /* API subsystem end */
/*
* protocol event code defines
*/
#define TXPECONN      0           /* connect indication */
#define TXPECONF      4           /* confirm indication */
#define TXPEDATA      8           /* normal data received */
#define TXPEXPDT      12          /* expedited data received  */
#define TXPEDISC      20          /* disconnect indication */
#define TXPERLSE      24          /* orderly release indication */
/*
* TPEND reason codes
*/
#define TXPRDRAN      0           /* operator drained subsystem */
#define TXPRSTOP      4           /* operator stopped subsystem */
#define TXPRTERM      8           /* subsystem terminated  */
/*
* length of TXP
*/
#define TXPLEN sizeof ( struct txp )
```