Connection Control Block structure

#include <ADSP.h>

```
typedef struct TRCCB {
  TPCCB          *ccbLink;          4    0   Address of next CCB
  short          refNum;            2    6   Volume reference number
  short          state;             2    8   Driver reference number
  char           userFlags;         1   10   User flags for connection
  char           localSocket;       1   11   Local socket number
  AddrBlock      remoteAddress;     4   12   Remote end internet address
  short          attnCode;          2   16   Attention code received
  short          attnSize;          2   18   Size of attention data
  Ptr            attnPtr;           4   22   Pointer to attention data
  unsigned char  reserved;        220   26   Reserved for use by .ADSP
} TRCCB;                          242

typedef TRCCB *TPCCB;
```

Notes: The internet address of the remote connection end is defined in the **CCB** by the AddrBlock data type:

Field descriptions

ccbLink — A pointer to the next **CCB**. This field is for use by **ADSP** only.

refNum — The reference number of the **CCB**. This number is assigned by **ADSP** when you establish the connection end.

state — The state of the connection end, as follows:

sListening	1	The socket is a **connection listening socket**-that is, a socket that accepts **ADSP** requests to open connections and passes them on to a socket client. This state is ordinarily used only by connection servers.
sPassive	2	The socket client is inactive but capable of accepting an **ADSP** request to open a connection. Unlike a connection listening socket, which passes the open-connection request on to a routine that can establish the connection on any socket, a socket client in the sPassive state can accept an open-connection request only to establish itself as a connection end.
sOpening	3	The socket client has sent an open-connection request and is waiting for acknowledgment.
sOpen	4	The connection is open.
sClosing	5	The socket client has requested that

THINK Reference © 1991-1992 Symantec Corporation

			ADSP close the connection, and **ADSP** is sending data or waiting for acknowledgment of data it has sent before closing the connection.
	sClosed	6	The connections closed.

userFlags — Flags that indicate an unsolicited connection event has occurred. An **unsolicited connection event** is an event initiated by **ADSP** or the remote connection end that is not in response to any .DSP routine that you executed. Each time an unsolicited connection event occurs, **ADSP** sets a flag in the userFlags field of the **CCB** and calls the routine you specified in the userRoutine parameter to the dspInit routine (if any). The user routine must read the userFlags field and then clear the flag to 0. **ADSP** cannot notify your routine of future events unless you clear the flag after each event.**ADSP** recognizes four types of unsolicited connection events, one corresponding to each of the flags in this field. The events and flags are defined as follows, where bit 7 is the most significant bit:

Event	Flag bit	Meaning
eClosed	7	**ADSP** has been informed by the remote connection end that the remote connection end has closed the connection.
eTearDown	6	**ADSP** has determined that the remote connection end is not responding and so has closed the connection.
eAttention	5	**ADSP** has received an attention message from the remote connection end.
eFwdReset	4	**ADSP** has received a forward reset command from the remote connection end, has discarded all **ADSP** data not yet delivered-including the data in the local client end's receive queue-and has resynchronized the connection.
none	3-0	Reserved.

localSocket — The socket number through which DDP transmits and receives the **ADSP** packets.

remoteAddress — The internet address of the socket used by the remote connection end.

attnCode — The attention code received by **ADSP** when the remote connection end sends an attention message.

attnSize — The size of the attention message received by **ADSP** when the remote connection end sends an attention message.

attnPtr — A pointer to a buffer containing the attention message received by **ADSP** from the remote connection end.

reserved — A data buffer reserved for use by **ADSP**.

THINK Reference © 1991-1992 Symantec Corporation