ATDDPRec structure

#include <AppleTalk.h>

typedef struct **ATDDPRec** {

		Size	Offset	Description
ABCallType	abOpcode;	1	0	Type of call
short	abResult ;	2	2	Result code
long	abUserReference;	4	4	For your use
short	ddpType;	2	8	DDP protocol type
short	ddpSocket	2	10	Source or listening socket number
AddrBlock	ddpAddress;	4	12	Destination or source socket address
short	ddpReqCount;	2	16	Length of datagram data or buffer size in bytes
short	ddpActCount;	2	18	Number of frame data bytes actually received
Ptr	ddpDataPtr;	4	20	Pointer to frame data or pointer to buffer
short	ddpNodeID;	2	24	original destination node ID
} ATDDPRec;		26		

typedef ATDDPRec ***ATDDPRecPtr**;
typedef ATDDPRec ****ATDDPRecHandle**;

Notes: When a DDP datagram is sent, ddpReqCount indicates the size of the datagram data in bytes and ddpDataPtr points to a buffer containing the datagram data. DDPSocket specifies the socket from which the datagram should be sent. DDPAddress is the internet address of the socket to which the datagram should be sent:

The network number you specify in ddpAddress.aNet tells MMP whether to create a long header (for an internet) or a short header (for a local network only). A short DDP header will be sent if ddpAddress.aNet is 0 or equal to the network number of the local network.

When a DDP datagram is received, ddpDataPtr points to a buffer in which the incoming data can be stored and ddpReqCount indicates the size of the buffer in bytes. The number of bytes actually sent or received is returned in the ddpActCount field. DDPAddress is the internet address of the socket from which the datagram was sent.

DDPType is the DDP protocol type of the datagram, and ddpSocket specifies the socket that will receive the datagram.

Warning: DDP protocol types 1 through 15 and DDP socket numbers 1 through 63 are reserved by Apple for internal use. Socket numbers 64 through 127 are available for experimental use. Use of these experimental sockets is not recommended for commercial products since there's no mechanism for eliminating conflicting usage by different developers.

Using DDP

Before it can use a socket, the program must call **DDPOpenSocket**, which adds a socket and its socket listener to the socket table. When a program is finished using a socket, call **DDPCloseSocket**, which removes the socket's

entry from the socket table. To send a datagram via DDP, call **DDPWrite**. To receive datagrams, you have two choices:

- Call **DDPOpenSocket** with NIL for sktListener; this installs the default socket listener provided by the **AppleTalk Manager**. Then call **DDPRead** to receive datagrams.
- Write you own socket listener and call **DDPOpenSocket** to install it. DDP will call your socket listener for every incoming datagram for that socket; in this case, you should not call **DDPRead**.