ATATPRec structure

#include <AppleTalk.h>

typedef struct **ATATPRec** {

		Size	Offset	Description
ABCallType	abOpcode;	1	0	Type of call
short	abResult ;	2	2	Result code
long	abUserReference;	4	4	For your use
short	atpSocket ;	2	8	Listening or responding socket number
AddrBlock	atpAddress ;	4	10	Destination or source socket address
short	atpReqCount ;	2	14	Request size or buffer size
Ptr	atpDataPtr ;	4	16	Pointer to buffer
BDSPtr	atpRspBDSPtr;	4	20	Pointer to response **BDS**
BitMapType	atpBitMap ;	8	24	Transaction bit map
short	atpTransID ;	2	226	Transaction ID
short	atpActCount ;	2	28	Number of bytes actually received
long	atpUserData ;	4	30	User bytes
Boolean	atpXO ;	2	34	Exactly-once flag
Boolean	atpEOM ;	2	35	End-of-message flag
short	atpTimeOut ;	2	36	Retry timeout interval in seconds
short	atpRetries ;	2	38	Maximum number of retries
short	atpNumBufs ;	2	40	Number of elements in response **BDS** or number of response packets sent
short	atpNumRsp ;	2	42	Number of response packets received or sequence number
short	atpBDSSize ;	2	44	Number of elements in response **BDS**
long	atpRspUData;	4	46	User bytes sent or received in transaction response
Ptr	atpRspBuf ;	4	50	Pointer to response message buffer
short	atpRspSize ;	2	54	Size of response message buffer
} **ATATPRec**;		56		

typedef ATATPRec ***ATATPRecPtr**;
typedef ATATPRec ****ATATPRecHandle**;

Notes: The socket receiving the request or sending the response is identified by atpSocket. ATPAddress is the address of either the destination or the source socket of a transaction, depending on whether the call is sending or receiving data, respectively. ATPDataPtr and atpReqCount specify the location and size (in bytes) of a buffer that either contains a request or will receive a request. The number of bytes actually received in a request is returned in atpActCount ATPTransID specifies the transaction ID. The transaction bit map is contained in atpBitMap, in the form:

typedef char BitMapType ;

Each bit in the bit map corresponds to one of the eight possible packets in a response. For example, when a request is made for which five response packets are expected, the bit map sent is binary 00011111 or decimal 31. If the second packet in the response is lost, the requesting socket will retransmit the request with a bitmap of 00000010 or decimal 2.ATPUserData contains the user bytes of an ATP header.ATPXO Is TRUE if the transaction is to be made with exactly-once service. ATPEOM is TRUE if

the response packet is the last packet of a transaction. If the number of responses is less than the number that were requested, then ATPEOM must also be TRUE. ATPNumRsp contains either the number of responses received or the sequence number of a response.

The timeout interval in seconds and the maximum number of times that a request should be made are indicated by atpTimeOut and atpRetries, respectively.

Setting atpRetries to 255 will cause the request to be retransmitted indefinitely, until a full response is received or the call is canceled.

ATP provides a data structure, known as a response buffer data structure (response **BDS**), for allocating buffer space to receive the datagram(s) of the response. A response **BDS** is an array of one to eight elements. Each **BDS** element defines the size and location of a buffer for receiving one response datagram; they're numbered 0 to 7 to correspond to the sequence numbers of the response datagrams.

ATP needs a separate buffer for each response datagram expected, since packets may not arrive in the proper sequence. It does not, however, require you to set up and use the **BDS** data structure to describe the response buffers; if you do not ATP will do it for you. Two sets of calls are provided for both requests and responses; one set requires you to allocate a response **BDS** and the other does not

Assembly-language note: The two calls that do not require you to define a **BDS** data structure (**ATPRequest** and **ATPResponse** are available in high-level language only.

The number of **BDS** elements allocated (in other words, the maximum number of datagrams that the responder can send) is indicated in the TReq by an eight-bit bit map. The bits of this bit map are numbered 0 to 7 (the least significant bit being number 0); each bit corresponds to the response datagram with the respective sequence number.

ATPRspBDSPtr and atpBDSSize indicate the location and number of elements in the response **BDS**.

ATPNumBufs indicates the number of elements in the response **BDS** that contain information. In most cases, you can allocate space for your variables of BDSType statically. However, you can allocate only the minimum space required by your ATP calls by doing the following:

```
BDSPtr myBDSPtr;
short numOfBDS;

numOfBDS = 3; /* Number of elements needed */
myBDSPtr = (BDSPtr) NewPtr(sizeof(BDSElement) * numOfBDS);
```

Note: The userBytes field of the **BDSElement** and the atpUserData field of the **ATATPRec** represent the same information in the datagram. Depending on the ATP call made, one or both of these fields will be used.