PAddResponse Send an additional response packet

#include <AppleTalk.h> **AppleTalk Manager**

```
OSErr     PAddResponse(thePBptr, async);
ATPPBPtr  thePBptr;   pointer to an ATPParamBlock structure
Boolean   async;      0=await completion; 1=immediate return
          returns     Error Code; 0=no error
```

PAddResponse sends an additional response packet to a socket that has already been send the initial part of a response via **PSendResponse**.

thePBptr iis a pointer to an ATPParamBlock structure.

Out-In	Name	Type	Size	Offset	Description
→	userData	long	4	18	user bytes from TRel
→	csCode	short	2	26	always addResponse
→	atpSocket	char	1	28	socket number
→	atpFlags	char	1	29	control information
→	addrBlock	AddrBlock	4	30	response destination
→	reqLength	short	2	34	response size
→	reqPointer	Ptr	4	36	pointer to response
→	rspNum	char	1	44	sequence number
→	transID	short	2	48	transaction ID

async is a Boolean value. Use FALSE for normal (synchronous) operation or TRUE to enqueue the request and resume control immediately. See Async I/O.

Returns: an operating system Error Code. It will be one of:

noErr	(0)	No error
badATPSkt	(-1099)	Bad responding socket
badBuffNum	(-1100)	Sequence number out to range
noSendResp	(-1103)	**PAddResponse** issued before **PSendResponse**
noDataArea	(-1104)	Too many outstanding ATP calls

Notes: userData contains the four user bytes. atpSocket contains the socket number from which the response should be sent. The end-of-message flag in atpFlags should be set if this response packet is the final packet in the transaction composed of a group of packets and the number of responses is less than requested. addrBlock indicates the socket to which the response should be sent. reqLength and reqPointer contain the size (in bytes) and location of the response to send; rspNum indicates the sequence number of the response (in the range 0 to 7). transID must contain the transaction ID.

Warning: If the transaction is part of an exactly-once transaction, the buffer used in the **PAddResponse** call must not be altered or released until the corresponding **PSendResponse** call has completed.

To send a request to another socket and get a response, call **PSendRequest**. The call terminates when either an entire response is received or a specified retry timeout interval elapses. To open a socket for the purpose of responding to requests, call **POpenATPSkt**. Then call **PGetRequest** to receive a request; when a request is received, the call is

completed. After receiving and servicing a request, call **PSendResponse** to return response information. If you cannot or do not want to send the entire response all at once, make a **PSendResponse** call to send some of the response, and then call **PAddResponse** later to send the remainder of the response. To close a socket opened for the purpose of sending responses, call **PCloseATPSkt**.