PSendResponse Send a response to a socket

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

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

PSendResponse sends a response to a socket.

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 sendResponse
→	atpSocket	char	1	28	socket number
→	atpFlags	char	1	29	control information
→	addrBlock	AddrBlock	4	30	response destination
→	bdsPointer	Ptr	4	32	pointer to response BDS
→	numBuffs	char	1	44	number of response packets being sent
→	bdsSize	Ptr	1	45	BDS size in elements
→	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
noRelErr	(-1101)	No release received
noDataArea	(-1104)	Too many outstanding ATP calls

Notes: If the response was part of an exactly-once transaction, userData will contain the user bytes from the TRel packet. atpSocket contains the socket number from which the response should be sent. The end-of-message flag in atpFlags should be set if the response contains the final packet in a transaction composed of a group of packets and the number of responses is less than requested. addrBlock indicates the address of the socket to which the response should be sent. bdsPointer points to a response BDS containing room for the maximum number of responses to be sent; bdsSize contains this maximum number. numOfBuffs contains the number or response packets to be sent in this call; you may wish to make **PAddResponse** calls to complete the response. transID indicates the transaction ID of the associated request.

During exactly-once transactions, **PSendResponse** doesn't complete until either a TRel packet is received from the socket that made the request, or the retry count is exceeded.

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**.