PSendRequest Send a request to another socket and wait for response

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

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

PSendRequest sends a request to another socket and waits for a reponse.

thePBptr iis a pointer to an SendReqparms structure.

Out-In	Name	Type	Size	Offset	Description
→	userData	long	4	18	user bytes
←	reqTID	short	2	22	transaction ID used in request
→	csCode	short	2	26	always sendRequest
←	atpSocket	char	1	28	bit map
↔	atpFlags	char	1	29	control information
→	addrBlock	AddrBlock	4	30	destination socket address
→	reqLength	short	2	34	request size in bytes
→	reqPointer	Ptr	4	36	pointer to request data
→	bdsPointer	Ptr	4	40	pointer to response BDS
→	numOfBuffs	char	1	44	number of responses expected
→	timeOutVal	char	1	45	timeout interval
→	numOfResps	char	1	46	number of responses received
→	retryCount	char	1	47	number of retries

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
reqFailed	(-1096)	Retry count exceeded
tooManyReqs	(-1097)	Too many concurrent requests
noDataArea	(-1104)	Too many outstanding ATP calls
reqAborted	(-1105)	Request cancelled by user

Notes: userData contains the four user bytes. addrBlock indicates the socket to which the request should be sent. reqLength and reqPointer contain the size and location of the request too send. bdsPointer points to a response BDS where the responses are to be returned; numOfBuffs indicates the number of responses requested. The number of responses received is returned in numOfResps. If a nonzero value is returned in numOfResps, you can examine atpSocket to determine which packets of the transaction were actually received. and to detect pieces for higher-level recovery.

timeOutVal indicates the number of seconds that **PSendRequest** should wait for a response before resending the request. retryCount indicates the maximum number or retries **PSendRequest** should attempt. The end-of-message flag of atpFlags will be set if the EOM Bit is set in the last packet received in a valid response sequence. The exactly-once flag should be set if you want the request to be part of an exactly-once transaction.

To cancel a **PSendRequest** call, you need the transaction ID; it's returned in reqTID. You can examine reqTID before completion of the call, but its contents are only valid after the tidValid bit of atpFlags has been set.

PSendRequest completes when either an entire response is received or the retry count is exceeded.

The value provided in retryCount will be modified during **PSendRequest** if any retries are made. This field is used to monitor ther number of retries; for each retry, it's decremented by 1.

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**.