ERead Read a data packet and place it in a data buffer

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

```
OSErr        ERead(thePBptr, async);
EParamBlkPtr thePBptr;   address of an EParamBlock structure
Boolean      async;      0=await completion; 1=immediate return
             returns     Error Code; 0=no error
```

ERead uses the default protocol handler to read a data packet and place it in a data buffer.

thePBptr is a pointer to an EParamBlock structure. The relevant fields are as follows:

Out-In	Name	Type	Size	Offset	Description
←	ioResult	short	2	16	Result code
→	csCode	short	2	26	Always ENetRead
→	eProtType	short	2	28	Ethernet protocol type
→	ePointer	long	4	30	pointer to data buffer
→	eBuffSize	short	2	34	size of data buffer
←	eDataSize	short	2	36	number of bytes read

async is a Boolean value. Use FALSE for normal (synchronous) operation or TRUE to function asynchronously. See Async I/O.

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

noErr	(0)	No error
LAPProtErr	(-94)	No protocol is attached or protocol handler pointer was not 0
reqAborted	(-1105)	ERdCancel or EDetachPH function called
buf2SmallErr	(-3101)	Packet too large for buffer; partial data returned

---

Notes: The **ERead** function uses the default protocol handler to read a data packet and place it in a data buffer. You can use the **ERead** function to read packets of a particular protocol type only after you have used the **EDetachPH** function to specify a NIL pointer to the protocol handler for that protocol type.

The ioResult parameter returns the result of the function. If you call the function asynchronously, the function sets this field to 1 as soon as it begins execution, and it changes the field to the actual result code when it completes execution. The csCode parameter is the routine selector, automatically set by the high-level language interface. It is always ENetRead for this function.

The eProtType parameter specifies the protocol type of the packet you want to read. The ePointer parameter is a pointer to the data buffer into which you want to read data, and the eBuffSize parameter indicates the size of the data buffer. If you are expecting EtherTalk data packets, the buffer should be at least 621 bytes in size; if you are expecting general Ethernet data packets, the buffer should be at least 1514 bytes in size.

The **ERead** function places the entire packet, including the packet header, into your buffer. The function returns in the eDataSize parameter the

number of bytes actually read. If the packet is larger than the data buffer, the **ERead** function places as much of the packet as will fit into the buffer and returns the buf2SmallErr result code.

Call the **ERead** function asynchronously to await the next data packet. When **The .ENET Driver** receives the data packet, it completes execution of the **ERead** function and calls your completion routine. The driver discards the data packet.if **The .ENET Driver** receives a data packet with a protocol type for which you specified the default protocol handler while no **ERead** command is pending.

You can have several asynchronous calls to the **ERead** function pending execution simultaneously, as long as you use a different parameter block for each call.