

EParamHeader

#define macro

#include <ENET.h>

#define EParamHeader		Size	Offset	Description
<u>QElemPtr</u>	qLink;	4	0	next queue entry
<u>short</u>	qType;	2	4	queue type
<u>short</u>	ioTrap;	2	6	routine trap
<u>Ptr</u>	ioCmdAddr;	4	8	routine address
<u>ProcPtr</u>	ioCompletion;	4	12	completion routine or <u>NIL</u>
<u>OSErr</u>	ioResult;	2	16	result code
<u>StringPtr</u>	ioNamePtr;	4	18	ptr to filename
<u>short</u>	ioVRefNum;	2	22	volume refnum or drive number
<u>short</u>	ioRefNum;	2	24	driver reference number
<u>short</u>	csCode;	2	26	call command code
		28	(size of aggregate)	

Notes: **EParamHeader** is a #define'd macro that defines the fields that are shared by the parameter block types used by the ENET Driver. These fields are placed in the beginning of every parameter block that has **EParamHeader** as its first field.

The qLink, qType, ioTrap, ioCmdAddr, ioNamePtr, and ioVRefNum fields are filled in by the **Device Manager**; your application should not have to set or read these fields. The ioResult field 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 ioCompletion field is a pointer to a completion routine that you can provide; the **Device Manager** calls your completion routine when it completes execution of the function. If you are not providing a completion routine, specify NIL for this field. You must obtain the driver reference number from the **OpenDriver** function and use it for the ioRefNum field.

The csCode field specifies the command to be executed; the high-level language interface fills in this value for you. **The .ENET Driver** accepts the following constants as routine selectors:

.ENET driver routine selectors

<u>ENetSetGeneral</u>	Set "general" mode
<u>ENetGetInfo</u>	Get info
<u>ENetRdCancel</u>	Cancel read
<u>ENetRead</u>	Read
<u>ENetWrite</u>	Write
<u>ENetDetachPH</u>	Detach protocol handler
<u>ENetAttachPH</u>	Attach protocol handler
<u>ENetAddMulti</u>	Add a multicast address
<u>ENetDelMulti</u>	Delete a multicast address