PGetAppleTalkInfo Obtain information about **The .MPP Driver**

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

OSErr	**PGetAppleTalkInfo(***thePBptr, async***);**	
MPPPBPtr	*thePBptr*;	address of a device control entry (DCE) structure
Boolean	*async*;	0=await completion; 1=immediate return
	returns	Error Code; 0=no error

PGetAppleTalkInfo returns information about **The .MPP Driver**. If the node on which your program is running happens also to be running AppleTalk Internet Router software in the background, more than one set of .MPP global variables may be in RAM. .

thePBptr is a pointer to a device control entry (DCE) 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 **PGetAppleTalkInfo**
→	version	short	2	28	Version of function
←	varsPtr	long	4	30	Pointer to .MPP globals
←	dcePtr	long	4	34	Pointer to DCE for .MPP
←	portID	short	2	38	Port number
←	configuration	long	4	40	Configuration flags
←	selfSend	short	2	44	Nonzero if self-send is enabled
←	netLo	short	2	46	Low value of the network range
←	netHi	short	2	48	High value of the network range
←	ourAddr	long	4	50	Local 24-bit AppleTalk address
←	routerAddr	long	4	54	24-bit address of router
←	numOfPHs	short	2	58	Max number of protocol handlers
←	numOfSkts	short	2	60	Max number of static sockets
←	numNBPEs	short	2	62	Max concurrent NBP requests
←	ntQueue	long	4	64	Pointer to registered names table
↔	laLength	short	2	68	Length in bytes of data link address
	(extended networks only)				
→	linkAddr	long	4	70	Pointer to data link address buffer
	(extended networks only)				
→	zoneName	long	4	74	Pointer to zone name buffer

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
paramErr	(-50)	Version number is too high

Notes: To make sure you are obtaining information about **The .MPP Driver** that handles application software, always use the **PGetAppleTalkInfo** function rather than the **Device Manager**'s **PBControl** function. If you are using assembly language or want to use the **PBControl** function, you must be sure to use a device driver reference number of -10 for **The .MPP Driver**

Field descriptions

ioResult The result of the function. When you execute the function asynchronously, the function sets this parameter to 1 and returns a function result of noErr as soon as the function begins execution. When the function completes execution, it sets the ioResult parameter to the actual result code.

csCode Routine selector, automatically set by the high-level language interface. Always equal to **PGetAppleTalkInfo** for this function.

version The version number of the **PGetAppleTalkInfo** function you are calling. For version number 53 of **The .MPP Driver**, this number is always 1.

varsPtr A pointer to the .MPP global variables. This parameter is reserved for the use of Apple Computer, Inc.; you cannot rely on the validity of the variables pointed to by this parameter.

dcePtr This is a pointer to the device control entry (**DCE**) data structure for **The .MPP Driver**. The DCE is described in **Device Manager**

portID The port number for **The .MPP Driver**. The port number is always 0 unless you are requesting information for an .MPP driver being used by a router.

configuration A 32-bit long word of configuration flags. The following flags are currently defined:

Bit	**Flag**	**Description**
31	SrvAdrBit	TRUE (1) if the routine that opened **The .MPP Driver** requested a server node number. Server node numbers are described in **Picking a Node Address**. This flag indicates only that the server node number was requested, not that it was returned. Some AppleTalk data links, such as EtherTalk, do not honor a request for a server node number.
30	RouterBit	TRUE (1) if an AppleTalk Internet Router was loaded at system startup (that is, there's a router operating on the same node as your application). A router can be loaded and not active.
15	ExtendedBit	TRUE (1) if the node is on an extended network. Testing this bit is the only way to determine whether you are on an extended network.
7	BadZoneHintBit	TRUE (1) if the zone name of the node you are on was not the same as the zone name stored in parameter RAM (some-times referred to as the *zone name hint*) when **The .MPP Driver** was opened. If the zone name hint is invalid, then the **AppleTalk Manager** uses the default zone for the network. The default zone is defined by the network administrator.
6	OneZoneBit	TRUE (1) if only one zone is assigned to your extended network or if you are not on an extended network. Use the ExtendedBit flag to determine whether you are on an extended network.

selfSend The ability of a node to send packets to itself. This feature is enabled when this parameter is nonzero. Use the **PSetSelfSend** function to enable or disable this feature.

netLo The low value of the range of network numbers on the local cable. Only extended networks can have a range of network numbers. For a nonextended network, this parameter returns the network number.

netHi The high value of the range of network numbers on the local cable. Only extended networks can have a range of network numbers. For a nonextended network, this parameter returns the network number.

ourAddr The 24-bit AppleTalk network address of the node you are on. The least significant byte of the long word is the node ID. The middle 16 bits are the network number. The most significant byte of the long word is reserved for use by Apple Computer, Inc.

routerAddr The 24-bit AppleTalk network address of the last router from which your node heard traffic. The least significant byte of the long word is the node ID. The middle 16 bits are the network number. The most significant byte of the long word is reserved for use by Apple Computer, Inc. You should always use this address when you want to communicate with a router

numOfPHs The maximum number of protocol handlers that this .MPP driver allows.

numOfSkts The maximum number of statically assigned sockets that this .MPP driver allows. Statically assigned sockets are described in *Inside AppleTalk,* second edition

numNBPEs The maximum number of concurrent requests to NBP that this .MPP driver allows.

ntQueue A pointer to the first entry in the names table for the local node. You can use NBP routines to look up and register names in the names table.

laLength The number of bytes of the data link address that the function should place in the buffer pointed to by the LinkAddr parameter. You use this parameter when you call the **PGetAppleTalkInfo** function on a node on an extended network. If you request more bytes than the total number of bytes in the address, then the function returns in the laLength parameter the actual number of bytes it placed in the buffer. If the address is longer than the size of the buffer, then the **PGetAppleTalkInfo** function fills the buffer and returns in the laLength parameter the actual length of the address, not the number of bytes returned. The function does *not* return an error when the buffer is too large or too small for the address. A value of 6 bytes for laLength is sufficient for most purposes.

linkAddr A pointer to a buffer for the data link address returned for extended networks only. You use the laLength parameter to specify the number of bytes of the address that you want placed in this buffer. You must allocate a buffer large enough to hold the number of bytes you specify. Specify NIL for this parameter if you do not want the function to provide a data link address.

zoneName A pointer to a buffer into which the **PGetAppleTalkInfo** function places the local node's zone name. You must allocate a buffer of at least 33 bytes to hold this data, or you must specify NIL for the zoneName parameter if you do not want to obtain the zone name. This field is returned only if the node is on an extended network.

A New NBP Wildcard Character

The Name-Binding Protocol (NBP) allows the use of certain wildcard characters in AppleTalk names when you call the **PLookupName** function. NBP now supports the following wildcard characters:

NBP wildcard characters

= All possible values. The equal sign (=) can be used alone instead of a name in the object or type field.

* This zone. The asterisk (*) can be used in place of the name of the zone to which this node belongs.

≈ Any or no characters in this position. The double tilde (≈) can be used to obtain matches for object or type fields. For example, pa≈l matches pal, paul, paper ball, and so forth. You can use only one double tilde in any string. Press Option-x to type the double tilde character on a Macintosh keyboard. If you use the double tilde alone, it has the same meaning as the equal sign (=). Note that any node not running AppleTalk Phase 2 drivers will not recognize this character