Calling Your ATQ Entry Operations flow

When you have used the **LAPAddATQ** function to add an entry to the AppleTalk Transition Queue, the **AppleTalk Manager** calls your entry when any of the following events occurs:

- a routine opens **The .MPP Driver**.
- a routine closes **The .MPP Driver**.
- a routine calls the **PATalkClosePrep** function.
- One of the routines in the AppleTalk Transition Queue denies permission for the routine that called the **PATalkClosePrep** function to close AppleTalk.
- an application calls the **ATEvent** or **ATPreFlightEvent** routine to send its own AppleTalk transition event to the entries in the AppleTalk Transition Queue.

When the **AppleTalk Manager** calls your AppleTalk Transition Queue routine, the stack looks like this:

SP→ Return address (4 bytes)
Routine selector (2 bytes)
Pointer to AppleTalk Transition Queue entry (4 bytes)
Routine-dependent parameter (4 bytes)
Previous contents

The first item on the stack (after the return address) is a routine selector. There is one routine selector for each type of transition. The open, prepare-to-close, permission-to-close, and cancel-close transitions each have a single-digit routine selector; all other routine selectors for AppleTalk transition events are 4-character codes. Codes starting with an uppercase letter (A through Z) are reserved for use by developers. All other codes are reserved for use by Apple Computer, Inc.

Routine selector	Transition
0x000000000	MPP driver opened
0x000000002	MPP driver about to close
0x000000003	**PATalkClosePrep** function has been called
0x000000004	Closing of .MPP driver has been canceled
0x041000000 – 0x05AFFFFFF	Reserved for use by developers
all others	Reserved for use by Apple Computer

You can use the following constants for the standard AppleTalk transitions:

ATTransOpen	open transition
ATTransClose	prepare-to-close transition
ATTransClosePrep	permission-to-close transition
ATTransCancelClose	cancel-close transition

The second item passed to your routine on the stack is a pointer to your routine's entry in the AppleTalk Transition Queue. You can use this pointer to

THINK Reference © 1991-1992 Symantec Corporation

get access to any fields at the end of the queue entry that you allocated for your own use. The last item passed to your routine on the stack is a parameter whose meaning depends on the type of transition.

The interface between the AppleTalk Transition Queue and your routine follows these conventions: your routine must preserve all registers except D0, D1, D2, A0, and A1; all parameters are passed on the stack as long words. Because your routine might be called at interrupt time, your routine must not make any direct or indirect calls to the Memory Manager and can't depend on handles to unlocked blocks being valid (This restriction on **Memory Manager** use does not apply to the open transition or the prepare-to-close transition). If you want to use any of your applications global variables, you must save the contents of the A5 register before using the variables, and you must restore the A5 register before your routine terminates.

> **Note:** It is important that you return a 0 in the D0 register whenever you receive a transition event routine selector that you do not recognize or do not choose to handle. Returning a nonzero value in the D0 register might cause the system to cancel an attempt to close AppleTalk, for example, or might be misinterpreted in some other way.

Open Transition

When an application software program initially calls the **MPPOpen** function, the **AppleTalk Manager** begins by attempting to open **The .MPP Driver**. If **The .MPP Driver** is already open, the **AppleTalk Manager** does not call the AppleTalk Transition Queue. If the **AppleTalk Manager** successfully opens **The .MPP Driver**, it then calls every routine listed in the AppleTalk Transition Queue with an open transition.

The third item on the stack for an open transition is a pointer to the start of the **Device Manager** extended parameter block used by the routine that opened **The .MPP Driver**. This pointer is provided for your information only; you must not change any of the fields in this parameter block.

Your AppleTalk Transition Queue routine can perform any tasks you wish in response to the notification that **The .MPP Driver** has been opened, such as using the Name-Binding Protocol (NBP) to register a name on the internet. Return 0 in the D0 register to indicate that your routine executed with no error.

Prepare-to-Close Transition

When any routine calls the **MPPClose** function in an attempt to close **The .MPP Driver**, the **AppleTalk Manager** calls every routine listed in the AppleTalk Transition Queue before **The .MPP Driver** closes. If the .MPP driver is already closed when a routine calls the **MPPClose** function, the **AppleTalk Manager** does not call the routines in the AppleTalk Transition Queue.

When the **AppleTalk Manager** calls your routine for a prepare-to-close transition, the third item on the stack is a NIL pointer.

Your routine can perform any tasks you wish to prepare for the imminent closing of AppleTalk, such as ending a session with a remote terminal and informing the user that the connection is being closed. You must return control to the **AppleTalk Manager** as quickly as possible. Return 0 in the D0

THINK Reference © 1991-1992 Symantec Corporation

register to indicate that your routine executed with no error.

Note: When the **AppleTalk Manager** calls your routine with a prepare-to-close transition (that is, a routine selector of ATTransClose), you cannot prevent **The .MPP Driver** from closing.

Permission-to-Close Transition

When a routine calls the **PATalkClosePrep** function to inform the **AppleTalk Manager** that it wants to close **The .MPP Driver**, The **AppleTalk Manager** calls every routine listed in the AppleTalk Transition Queue to request permission to close **The .MPP Driver**.

When the **AppleTalk Manager** calls your routine to request permission to close **The .MPP Driver**, the third parameter on the stack is a pointer to a 4-byte buffer. If you intend to deny the request to close AppleTalk, you should place in the buffer a pointer to a pascal string containing the name of your application. The **PATalkClosePrep** function returns this pointer. The routine that called the **PATalkClosePrep** function can then display a dialog box telling the user the name of the application that is currently using AppleTalk.

Your routine can return either a function result of 0 in the D0 register, indicating that it accepts the request to close, or a 1 in the D0 register, indicating that it denies the request to close. Note that the Operating System might elect to close AppleTalk anyway; for example, if the user grants permission to close in response to a dialog box.

Because the **AppleTalk Manager** calls your routine again (with the routine selector set to ATTransClose) before **The .MPP Driver** actually closes, it is not necessary for your routine to do anything other than grant or deny permission in response to being called for a permission-to-close transition. However, you might want to prohibit the users from opening new sessions or establishing new connections while you are waiting for **The .MPP Driver** to close.

Cancel-Close Transition

When any routine in the AppleTalk Transition Queue denies permission for **The .MPP Driver** to close, the **AppleTalk Manager** calls each routine listed in the AppleTalk Transition Queue that has already received the permission-to-close transition to inform them that the request to close **The .MPP Driver** has been canceled.

When the **AppleTalk Manager** calls your AppleTalk Transition Queue routine for a cancel-close transition, the third item on the stack is a NIL pointer.

If your routine performed any tasks to prepare for the closing of AppleTalk, it should reverse their effects when it is called with the routine selector set to ATTransCancelClose. Return 0 in the D0 register to indicate that your routine executed with no error.

Developer-Defined Transitions

Any AppleTalk transition event code that begins with an uppercase letter (that is, any value in the range 0x041 00 00 00 through 0x05A FF FF FF)

THINK Reference © 1991-1992 Symantec Corporation

indicates a developer-defined event. You can use such events to send messages to your own entries in the AppleTalk Transition Queue, or you can define events and make them public for others to use. Because you cannot tell how the originator of such an event might interpret a nonzero function result, you must always return 0 in the D0 register for any AppleTalk transition event code that you do not recognize.

When you return a nonzero result code for certain developer-defined transitions, the **AppleTalk Manager** may call your AppleTalk Transition Queue routine a second time with a cancel transition analogous to the cancel-close transition.

THINK Reference © 1991-1992 Symantec Corporation