Technical Note NW505
AppleTalk Data Stream Protocol Q&As

CONTENTS

This Technical Note contains a collection of archived Q&As relating to a specific topic - questions sent the Developer Support Center (DSC) along with answers from the DSC engineers. Current Q&A's can be found on the Macintosh Technical Q&A's web site.

[Oct 01 1990]

AppleTalk Data Stream Protocol references

Date Written: 11/13/90

Last reviewed: 8/1/92

Where are the AppleTalk Data Stream Protocol (ADSP) calls documented?

APDA has a product called "ADSP: AppleTalk Data Stream Protocol Development Kit" that gives you ADSP and the ADSP call descriptions. ADSP and ADSP interfaces can be found on the Developer CD and can be used for your development purposes. However, you'll need to license ADSP through Apple Software Licensing (AppleLink: SW.LICENSE) before you ship your product. Calls are also documented in Chapter 32 of *Inside Macintosh* Volume VI.

If you want to use ADSP, then you should read ALL of the documentation. There are various tips and explanations that may be vital to your implementation of ADSP in your product. It is useful for transferring data between machines and provides built-in error checking to ensure that you do receive the data at the other end. For these reasons, you should read *Inside AppleTalk,* Chapter 12, Apple Data Stream Protocol, to make sure you utilize it to its fullest.

Back to top

Sending broadcast packets to transmit data to multiple clients

Date Written: 11/19/90

Last reviewed: 6/14/93

How do I broadcast information over AppleTalk using the Communications Toolbox (CTB) and the AppleTalk Data Stream Protocol Development (ADSP) Tool or multiple access? I cannot find any significant information regarding broadcast or multiple access transmissions.

There is no supported method to send broadcast packets to transmit data to multiple clients. The reason is that this method does not ensure the data was received by every client. It is possible to lose a packet on a network before it reaches a client even though it may reach the others. If this is the case then the data is lost forever for this client unless you resend the packet. This is highly inefficient and unreliable.

If you want to have multiple clients, then you'll need to write your own server software that can handle multiple clients logged on simultaneously. The server will then need to keep a record of every connection and its corresponding transaction ID even though you will be sending the same information to each client; this is to ensure that each client receives the data. Following the ADSP protocol should be enough for you to implement this server/client connection and ensure that the data is reliably transmitted to each node.

Back to top

ADSP closes connections on eClosed or eTearDown events

Date Written: 7/30/91

Last reviewed: 6/14/93

When I receive an `eClosed` or `eTearDown` ADSP connection event, do I have to close my end of the connection, or does ADSP close it automatically?

When you get a `eClosed` unsolicited event, it means the other (remote) side of the ADSP connection sent a Close Connection Advice control packet and ADSP has closed the connection. It is purely advisory and does not require you to take any action. The Close Connection Advice control packet allows ADSP to close connections faster than just waiting for the connection to time-out due to the open timer expiring.

When you get a `eTearDown` unsolicited event, it means the network connection with other side of the ADSP connection was lost (the open timer expired) and ADSP has closed the connection. Again, it is purely advisory and does not require you to take any action.

The ADSP chapter of *Inside AppleTalk* (Chapter 12) might with help other questions you may have about ADSP in the future since it explains how ADSP works beneath the Macintosh implementation documented in *Inside Macintosh* Volume VI.

Back to top

ADSP dspAttention and attention retransmit timer interval

Date Written: 9/16/91

Last reviewed: 6/14/93

I'm having problems with the AppleTalk Data Stream Protocol (ADSP) call dspAttention. Sometimes it takes several minutes for `dspAttention` to complete.I've tried using ADSP 1.5.1 with System 6.0.7 and ADSP 56, which ships as part of System 7.

ADSP 1.5.1 (and ADSP 56) no longer uses the `attnInterval` field in the `DSPParamBlock` for the attention retransmit timer interval. Instead, the attention retransmit timer interval is based on the round-trip time of ADSP data packets. When a connection end is established, the attention retransmit acknowledged, the attention retransmit timer interval is adjusted for the connection speed. This mechanism works well when a connection end is established, an ADSP connection is opened, the connection is used, the connection is closed, and then the connection is eliminated.

A problem in this mechanism can surface when a connection end is established and is reused over and over for several ADSP connections. When a connection is closed and then a new connection is opened using the same connection end, the attention retransmit timer interval is not reset to the default starting value (the problem). Instead, some garbage value in memory is used. If the first ADSP attention packet or the acknowledgment packet returned for that attention packet is lost on the network, ADSP will use the "garbage" value for the attention retransmit timer interval and it could take several minutes for ADSP to retry sending the attention packet.

This has been fixed for versions of ADSP greater than version 56. With versions of AppleTalk 56 and earlier, the only workaround is to use a new connection end for every new ADSP connection requiring ADSP's attention mechanism.

Back to top

Using ADSP with AppleTalk for Apple's VMS

Date Written: 5/3/89

Last reviewed: 6/14/93

I am trying to create a simple program to try ADSP (AppleTalk Data Stream Protocol) with AppleTalk for Apple's VMS (version 2). I started the bridge process but got an access violation on the first call.

You need to make three calls before you can call any other functions with AppleTalk for VMS. These calls are:

```
    ATK$INIT_ATALK
    ATK$INIT_PORT
```

Then you can use the other routines.

Back to top

Appletalk Internet Router and Internet Protocol support

Date Written: 5/15/92

Last reviewed: 8/1/92

Does the Appletalk Internet Router support the routing of Internet Protocol packets, or Datagram Delivery Protocol-to-Internet Protocol translation?

The AppleTalk Internet Router does not provide support for routing IP packets or the translation of IP packets encapsulated inside AppleTalk packets into IP packets. It only supports the routing of AppleTalk packets between AppleTalk link layers such as LocalTalk, EtherTalk, and TokenTalk. To route IP packets between, for instance, a LocalTalk network and an IP only backbone, you'll need to purchase a hardware router such as a Cayman Gatorbox or a Shiva Fastpath. Using one of these routers you can, using MacTCP, tunnel IP packets in AppleTalk packets by selecting the appropriate AppleTalk adev from within the MacTCP Control Panel. This will allow transfer of IP packets over the wire and the hardware routers will then strip the AppleTalk protocol, allowing IP packets to reach their destination over Ethernet. In addition, these routers usually support tunneling AppleTalk within IP packets (IPTalk).

If both the host computer running the TCP/IP protocol stack and the Macintosh you wish to connect from are on the same Ethernet wire, you can select Ethernet from the MacTCP Control Panel to bypass the encapsulation of IP packets within AppleTalk DDP packets.

Back to top

ADSP session limits

Date Written: 7/21/92

Last reviewed: 6/14/93

What's the absolute maximum number of ADSP sessions that a single Macintosh can have? What's the largest client community that can be served at any given time?

As documented in *Inside AppleTalk* 2nd Edition, page 4-5, dynamic sockets can range from 128-254. Therefore, when using ADSP you have 126 different dynamic sockets available for use as connection ends.

Note that you may have only one connection open between a pair of sockets. However, it is possible to have multiple connections open on the same socket, provided that the other end of each connection resides on a different socket. For example, socket 128 on machine B may have connections with socket 128 on machine C and socket 128 on machine D, but it may not have more than one connection with the same socket of a remote machine.

When a connection is established between two sockets, each connection end is assigned a unique connection ID value (`ConnID`). The `ConnID` is an unsigned integer. Therefore, it is theoretically possible to have 65,535 different connections on a single machine.

However, since each connection requires that you allocate a CCB data structure (242 bytes), you will most likely run out of memory long before this limit is reached. In addition, the more connections you have, the more traffic you will generate on the network since each connection is maintained internally by ADSP with probe control packets and connection timers (see Chapter 12 of *Inside AppleTalk* 2nd Edition for the details). On top of that, if each connection is transferring a lot of data, the burden of passing ADSP packets around will put an even greater strain on your machine and the network in general.

Back to top

Downloadables

	Acrobat version of this Note (K)	Download

Back to top

Technical Notes by Date | Number | Technology | Title
Developer Documentation | Technical Q&As | Development Kits | Sample Code