



Tech Info Library

Open Transport 1.0.8: Applications Compatibility Q & A (3/96)

Revised: 3/26/96
Security: Everyone

Open Transport 1.0.8: Applications Compatibility Q & A (3/96)

=====
Article Created: 27 October 1995
Article Reviewed/Updated: 26 March 1996

TOPIC -----

This article is a series of questions and answers on applications compatibility in Apple Open Transport.

Open Transport 1.1 is now available, and Apple recommends upgrading to it. Also refer to Open Transport 1.1 Reference Questions and Answers Tech Info Library articles for the most recent information.

DISCUSSION -----

Question: Is Open Transport compatible with existing applications and network extensions?

Answer: Apple and third party developers have to date announced over 130 Open Transport compatible applications. A compatibility list is available from:
ftp://seeding.apple.com/opentransport/OT_Compat_List

Open Transport provides "backward compatibility" services in four areas:

- to support existing applications using the documented AppleTalk APIs;
- to support existing applications using the documented MacTCP APIs;
- to support existing Chooser devices; and,
- to existing NuBus based network interface cards to work with Open Transport.

Question: What is implied when an application is "Open Transport Compatible"? Does that mean that it takes advantage of new Open Transport features?

Answer: Apple has defined three levels of interoperability with Open Transport. The first, Open Transport Compatible, is used to describe a network application originally developed for "classic" AppleTalk or MacTCP, that now takes advantage

of Open Transport Compatibility Services. These applications automatically gain the benefits associated with the new Open Transport configuration utilities. However, they will not realize a significant performance increase on Power Macintosh computers, nor can they take advantage of Open Transport's transport-independence capabilities.

The second, Open Transport Ready, are those applications that have been modified to adopt the new Open Transport APIs (XTI). They are PowerPC native, in addition to running on 680x0-based Macintosh systems. Open Transport Ready applications not only benefit from the new configuration utilities, but have the opportunity for a significant performance boost when running on Power Macintosh.

The third and final category of interoperability is referred to as Open Transport Enhanced. In addition to adopting the new Open Transport APIs and being Power PC native, these applications have been modified to exploit the transport-independent capabilities of Open Transport, that is, they can be dynamically configured to support AppleTalk, TCP/IP, or serial communications.

Question: How is backward compatibility for AppleTalk implemented?

Answer: AppleTalk applications backwards compatibility is accomplished by intercepting all AppleTalk networking calls at the ".ddp" driver API. Above this protocol layer, applications written to the classic AppleTalk APIs continue to rely on the classic (680x0 based) implementation of AppleTalk. Calls to the ".ddp" driver are translated to the corresponding Open Transport XTI calls and are then passed to the new native implementation of DDP for processing. The process is reversed for incoming packets.

Using this approach, backwards compatibility is very robust - the classic implementations of ADSP, ASP, ATP, NBP, ZIP, and PAP are actually present (vs. simply mimicked). This also decreases the total memory footprint of backwards compatibility as compared to an implementation based on individual adaptation layers for each of the AppleTalk protocols. The primary trade-off of this approach is that applications relying on backwards compatibility do not gain any meaningful performance increases on Power Macintosh; essentially only native DDP is actually in use in these cases.

Open Transport/AppleTalk also includes broad support for existing applications software and network devices that rely on the Chooser or the Network Control Panel software for selection and configuration, known as "cdevs" and "adevs" respectively.

Question: How is backward compatibility for MacTCP implemented?

Answer: TCP/IP (MacTCP) applications backwards compatibility is accomplished by intercepting all MacTCP networking calls at the ".ipp" driver level. Calls to the ".ipp" driver are translated to corresponding Open Transport XTI calls and then passed to the native TCP/IP stack for processing. The process is reversed for incoming packets.

This approach allows most MacTCP applications to benefit from the native

implementation of the TCP/IP protocols on Power Macintosh, at least to some degree. While the backwards compatibility layer itself must run as 680x0 code, most of the handling of the packet happens in the new native Open Transport/TCP implementation. The drawback of this implementation is that "warts and all" backward compatibility is somewhat less robust; applications depending on idiosyncrasies of MacTCP or referencing internal MacTCP data structures are likely to need an update.

TCP/IP backward compatibility also includes targeted support for select software products that rely on the MacTCP (or Admin TCP) Control Panel software for configuration. Support for these software modules, known as MacTCP Link Access Modules, or simply "mdevs", is more limited than that provided for AppleTalk "adevs", due to certain technical considerations.

Question: How is backward compatibility for other network products implemented?

Answer: Certain networking extensions -- such as MacIPX from Novell or PATHWORKS (LAT and DECnet) from Digital Equipment Corp. or Thursby Software Systems -- as well as applications such as Insignia Solutions SoftWindows do not access the AppleTalk or MacTCP APIs; instead they deal directly with the Macintosh OS Ethernet driver software. With the introduction of PCI bus to the Macintosh this driver architecture and software has changed.

Additional backward compatibility software included with the System Enabler 1.1 for System 7.5.2 (that is, System 7.5.2 version 2) allows these networking extensions and applications to communicate with the new DLPI based Ethernet driver as if it were a "classic" Ethernet device.

This compatibility software supports only the built-in Ethernet adapter of Power Macintosh systems with PCI bus (7200, 7500, 8500, and 9500). Thus current versions of such network extensions and applications (those that write directly to the Ethernet driver) cannot access PCI based network interface cards.

This backward compatibility module was not available in the original distribution of System 7.5.2 for the Power Macintosh 9500 (with Open Transport v1.0). It is now included as part of System 7.5.2 version 2, or more recent, and is recommended for use with Open Transport 1.0.6 or later (see Availability and Distribution).

Question: Are there known limitations to applications backward compatibility?

Answer: Yes. Applications that rely on undocumented APIs or examine private data structures in the current AppleTalk or MacTCP may not be fully compatible with Open Transport.

Examples include the MacSNMP AppleTalk and TCP/IP Agents (however, MacSNMP and the Macintosh System Agent are compatible), the Apple Internet Router 3.x, the Apple Remote Access MultiPort Server 2.x, LaserWriter Bridge, and some utilities like MacTCP Watcher and MacTCP Spy. Updated versions of these software products will be required for full compatibility.

Question: There have been reports of problems with the use of Open Transport and MacX 1.2. Is there an Open Transport compatible X Window System server available?

Answer: Apple MacX 1.5 is compatible with Open Transport, and is a recommended upgrade for customers who have earlier versions of MacX. There is a known bug in MacX 1.2 that can cause it to crash when running on a Macintosh system with System 7.5.2 and Open Transport.

Article Change History:

26 Mar 1996 - Added statement on Open Transport 1.1 release.
20 Nov 1995 - Added Q & A on Open Transport Compatible means.

Copyright 1995-96, Apple Computer, Inc.

Tech Info Library Article Number:18836