



# Tech Info Library

## Open Transport 1.1.1: Applications Compatibility FAQ (10/96)

Revised: 10/23/96  
Security: Everyone

Open Transport 1.1.1: Applications Compatibility FAQ (10/96)

=====  
Article Created: 16 October 1996  
Article Reviewed/Updated: 23 October 1996

TOPIC -----

This article is the Open Transport 1.1.1, Applications Compatibility FAQ (frequently asked questions).

DISCUSSION -----

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

Answer: Apple and third party developers have announced hundreds of Open Transport compatible applications. Open Transport includes software libraries that provide "backward compatibility" services in four areas:

- to support existing applications that utilize the documented AppleTalk APIs;
- to support existing applications that utilize the documented MacTCP APIs;
- to support MacTCP Link Access Extensions (mdevs) on case by case basis; and,
- to support existing NuBus based network interface cards and drivers.

Note: WinSock compatibility services are provided through third party software, available from NetManage. For more information, please refer to Open Transport and Cross-Platform Issues.

Open Transport compatibility services are available to applications accelerated for PowerPC as well as for 680x0 applications, as it provides the necessary "mixed-mode glue" as part of the Software Developer's Kit (SDK).

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 - known as 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 PowerPC MacOS systems, nor can they take advantage of Open Transport's transport-independence capabilities.

Open Transport Ready applications are those that have been modified to adopt the new Open Transport APIs (XTI). They are PowerPC native, in addition to running on 680x0 MacOS 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 PowerPC MacOS.

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 PowerPC 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 PowerPC MacOS; essentially only native DDP is actually in use in these cases. (see Figure 1)

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 PowerPC MacOS, 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. (See Figure 1)

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: Are there other compatibility issues with PCI MacOS systems and existing network software products?

Answer: Certain network software products - such as MacIPX from Novell, PathWORKS (LAT and DECnet) from Digital Equipment Corp. or Thursby Software Systems, and Insignia Solutions SoftWindows - interact directly with the MacOS ethernet hardware, expansion slots, and driver software. With the introduction of PCI and Open Transport to MacOS, these elements have changed substantially; PCI has replaced NuBus, the system Registry has replaced the Slot Manager, Open Firmware has replaced the role of NuBus ROMs, and DLPI has replaced the ".enet" driver API. If not addressed, these changes would have resulted in significant compatibility problems between PCI MacOS systems and these network software products.

To support these types of products, MacOS System Software 7.5.2 (and more recent) includes a compatibility library that allows these products to identify and communicate with the built-in ethernet controller on PCI MacOS systems as if it were a "classic" NuBus ethernet device. This compatibility software emulates not only the necessary low-level ethernet driver calls, but also the Slot Manager and other related APIs necessary to preserve compatibility with these applications.

This compatibility software is limited, however, in that it supports access only to the built-in ethernet adapter of PowerPC MacOS systems with PCI. Thus, existing versions of products such as MacIPX and SoftWindows will not be able to take advantage of PCI network interface options such as token ring, fast ethernet, or FDDI. New versions of these applications will be required to gain full access to all PCI NIC options.

Note that this compatibility module was not available in the original distribution of System 7.5.2 for the Power Macintosh 9500 (with Open Transport

v1.0). This is one of the reasons that customers with 9500s were urged to update to System Enabler 1.1 or greater.

With the availability of System 7.5.3 this compatibility library is included as a part of system software. Customers using MacIPX should note that an updated version of the compatibility library is included as a Custom Install option with the System Update 2.0 installer. For more information regarding this updated library, please refer to the System Update 2.0 release notes.

Question: How is backward compatibility for NuBus network interface cards implemented?

Answer: For 680x0 and PowerPC MacOS systems with NuBus, Open Transport v1.1 will allow use of existing NuBus NICs and drivers. This compatibility is provided by software support mapping DLPI driver calls generated by new Open Transport protocols to corresponding calls to "classic" MacOS LAP Manager, .enet and .tokn APIs.

Question: Are there other known limitations to applications backward compatibility?

Answer: Yes, there are some. Applications that rely on undocumented APIs or examine private data structures in classic AppleTalk or MacTCP will 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 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 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 all customers who have any earlier versions of MacX.

Although not related to Open Transport, there is a known bug in MacX 1.2 that can cause a system crash when running on System 7.5.2.

Question: There have been reports of problems with Apple Remote Access 2.0.1 and Open Transport. What is the status?

Answer: Apple Remote Access 2.0.1 is fully compatible with Open Transport v1.1.

Apple Remote Access 2.0.1 running on a system with the earlier Open Transport 1.0.x releases always operated in the "Remote Only" mode. In that mode, only resources at the remote site would be visible in the Chooser while connected via

dial-up; local resources would reappear when the dial-up link was disconnected.

Question: There have been reports of problems with the use of Open Transport 1.0.x and network protocol analysis tools such as ag Group Etherpeek and Neon Software NetMinder. What is the status of this?

Answer: Open Transport v1.1 includes support for promiscuous mode and raw packet access. Apple is currently working with vendors to assure that ethernet and other NIC driver developers are implementing promiscuous mode support in a consistent manner.

Open Transport v1.0.x did not include support for the DLPI messages necessary to enable promiscuous mode on ethernet hardware. Because applications such as Etherpeek and NetMinder rely on this capability, they could not perform data collection on systems running these earlier versions of Open Transport.

Question: There have been reports of problems with the Apple LaserWriter Bridge and LocalTalk Bridge and Open Transport. What is the status?

Answer: Apple LaserWriter Bridge (LWB) and LocalTalk Bridge (LTB) software were designed to check for a specific version of AppleTalk software. Because Open Transport registers itself as a more recent version, these programs currently will not launch. An update for Open Transport-compatibility is currently underway.

Availability and distribution of these updates will be announced in the near future.

Question: There have been reports of problems with MacTCP Ping and Open Transport. What is the status?

Answer: MacTCP Ping is an unsupported utility developed by Apple Computer specifically for MacTCP. It is not compatible with Open Transport.

Compatible Ping utilities, such as MacPing Pro from Dartmouth, are currently available.

Question: There have been reports of problems with Assistant ToolBox and Open Transport. What is the status?

Answer: Prior versions of Assistant Toolbox (v1.2 and earlier) , which has shipped as a component of the PowerBook Productivity Bundle, were discovered to have a conflict with Open Transport v1.1.

An updated version of Assistant Toolbox is included as a part of System Update 2.0 (System 7.5.3).

Question: There have been reports of problems with At Ease and Open Transport.

What is the status?

Answer: At Ease v3.0.1 and earlier have a conflict with Open Transport. At Ease v3.0.2 and later are compatible.

Article Change History:

23 Oct 1996 - Changed distribution status.

Copyright 1996, Apple Computer, Inc.

Tech Info Library Article Number:20544