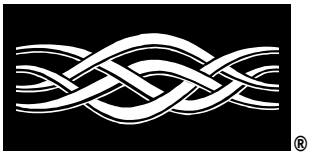
Microsoft

Microsoft® Windows NT® Server

Server Operating System



White Paper

Comparing Microsoft Windows NT® Server 4.0, Terminal Server Edition, and UNIX Application Deployment Solutions

Abstract

This white paper discusses the deployment of Microsoft® Windows NT® Server 4.0, Terminal Server Edition, in UNIX-based environments including solutions for the integration and migration of existing UNIX applications into a Terminal Server environment.

INTRODUCTION

Microsoft® Windows NT® Server 4.0, Terminal Server Edition (Terminal Server) delivers the Windows experience to diverse desktop hardware through terminal emulation. Terminal Server supports Windows-based terminals, existing Windows-based 32-bit computers, older 16-bit Windows-based desktops, and Apple Macintosh, MS-DOS®, and UNIX-based desktops. (Non-Windows devices are supported via a third party add-on from Citrix.) Terminal Server:

- Provides a "thin client" solution to deliver 32-bit Windows to a wide range of desktop hardware.
- Combines the low cost of the traditional mainframe and terminals with the familiarity, ease of use, and breadth of applications support offered by the Windows operating system platform.

Terminal Server provides user access to 16- or 32-bit Windows-based applications from any of the following types of desktops:

- A new class of low-cost hardware, commonly referred to as Windows-based Terminals, marketed by third-party vendors. A Windows-based Terminal contains an embedded terminal-emulation client.
- Any existing 32-bit Windows desktop operating system, such as Windows 98 or Windows NT Workstation (running the terminal emulation client as a window within the local desktop environment).
- Older 16-bit Windows-based desktops running the Windows 3.11 operating system (running the 16-bit terminal emulation client as a window within the local desktop environment).
- X-based terminals, Apple Macintosh, MS-DOS, Networked Computers, or UNIX-based desktops via a third-party add-on product.

Product Overview

The Terminal Server product consists of four components: the Windows NT Server multi-user core, the Remote Display Protocol, the Windows-based client software, and enhanced system administration tools.

- **Windows NT Server multi-user core** Provides the ability to host multiple, simultaneous client sessions on Windows NT Server 4.0 or higher. Terminal Server can directly host compatible multi-user client desktops running on a variety of Windows-based and non Windows-based hardware. Standard Windows-based applications, if properly written, do not need modification to run on Terminal Server, and all standard Windows NT-based management infrastructure and technologies can be used to manage the client desktops.
- **Remote Display Protocol** Allows a client to communicate with the Terminal Server over the network. This protocol is based on the International Telecommunications Union T.120 protocol, used first in NetMeeting™

conferencing software. It is a multi-channel protocol tuned for high-bandwidth enterprise environments. The protocol also supports three levels of encryption.

- **Terminal Server Client** Displays the familiar 32-bit Windows user interface on a range of desktop hardware – new Windows-based terminal devices (embedded); personal computers running Windows 95, Windows 98, or Windows NT Workstation 3.51 or higher; personal computers running Windows for Workgroups (Windows 3.11).
- **Administration Tools** To the standard Windows NT Server administration tools, adds the Terminal Server License Manager, Terminal Server Client Creator, Terminal Server Client Connection Configuration and Terminal Server Administration tools for managing client sessions. Two new objects, Session and User, are also added to the Performance Monitor to allow tuning of the server in a multi-user environment.

Windows NT Terminal Server Edition Multi-User Architecture

Using Windows-based terminals and the Remote Desktop Protocol (RDP), Windows applications run on the Terminal Server and are distributed to the thin clients. The Terminal Server executes the applications on behalf of the clients and only passes I/O (screen and keyboard) information to the thin clients (see Figure 1).

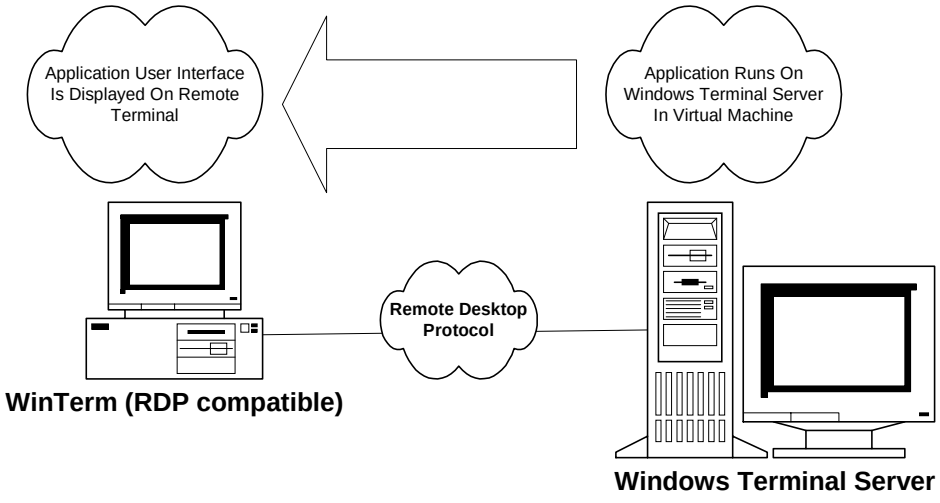


Figure 1. Terminal Server interaction with client.

Terminal Server communicates with RDP-compatible Windows Terminals such as those produced by Network Computing Devices (http://www.ncd.com), Wyse Technology (http://www.wyse.com), Tektronix (http://www.tektronix.com), NeoWare

Systems (http://www.neoware.com), and Boundless Technologies (http://www.boundless.com), as well as RDP clients on traditional desktop PCs. When MetaFrame (from Citrix Systems, Inc., http://www.citrix.com) is added, Terminal Server can communicate to even more thin client devices. The ICA (Independent Computing Architecture) protocol stack included with MetaFrame can also run on traditional PC clients, or even through ICA-enabled web browsers. This flexibility allows application access to any client -- regardless of their platform. Figure 2 illustrates how the Remote Desktop and Citrix MetaFrame protocols are used to communicate with various thin clients.

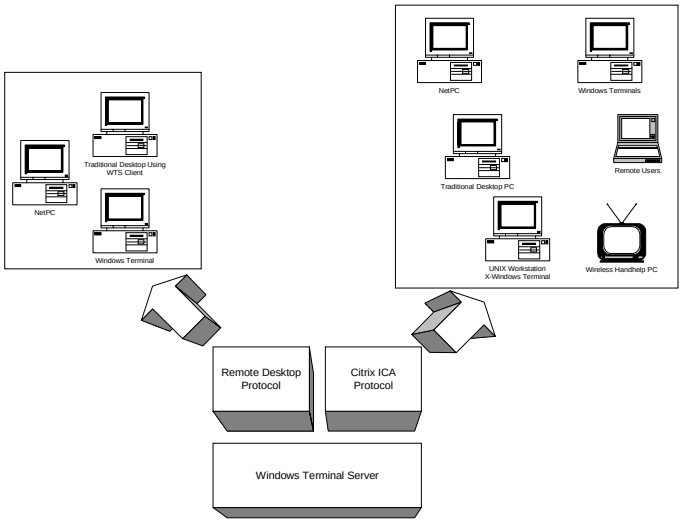


Figure 2. The role of the Remote Desktop and Citrix ICA protocols with thin clients.

In typical UNIX environments, application deployment involves a centralized UNIX host system with client access provided through UNIX workstations or X-Windows terminals. The hardware for such clients is quite expensive. Terminal Server offers a cost-effective and highly scalable alternative that can be closely integrated with existing UNIX systems.

Terminal Server and UNIX installations have quite different system requirements. Terminal Server relies on powerful servers to provide all of the processing with very thin and cheap clients. Typical UNIX applications require moderate to high-end servers (depending on the environment) and clients that also possess at least a moderate amount of computing power. Table 1 illustrates the suggested client and server system requirements of a 10-user deployment of Terminal Server and SCO OpenServer 5, both using Graphical User Interface applications. As you can see, while the server requirements for Terminal Server are higher, the cheaper clients will result in an overall lower cost.

System Requirements	Windows NT Terminal Server Edition	SCO OpenServer 5
Server Hardware	Intel Pentium Pro 200 MHz Processor 128 MB RAM Network interface card At least 1 GB hard drive	Intel Pentium 200Mhz Processor 64 MB RAM Network interface card At least 1 GB hard drive
Client Hardware	Windows-based Terminal with RDP protocol support (priced starting from $375) **– OR –** PC with an 80386 or higher microprocessor bytes 4 MB RAM 4 MB available hard disk space VGA or higher-resolution video adapter Network interface card (NIC) using the Microsoft TCP/IP protocol Microsoft mouse or 100 percent compatible	PC with 80486 or higher processor 8 MB RAM Disk space required dependent on X Server Super VGA or higher-resolution monitor Network interface card Mouse
Client Software	Windows for Workgroups or higher Remote Desktop Protocol Client (included with Windows Terminal Sever)	X Server application

Table 1. Client and server system requirements of a 10-user deployment of Terminal Server and of SCO OpenServer 5

How Terminal Server Compares

The bulk of SCO OpenServer deployments are in two primary areas: Replicate Site and Small Office installations. Typically, these environments are a single UNIX server and 5 to 10 clients.

Replicated Site UNIX Installations

Replicate Site installations make up about 10-15 percent of the SCO UNIX market. These installations are typically in small and franchise business locations with 5 to 10 users, where the primary requirements are high reliability and remote access. The same applications and system configuration is usually replicated in multiple

locations. In this environment, clients are dumb terminals, running point of sale and inventory applications.

Terminal Server and thin-client architecture are a suitable alternative in such an environment. Table 2 shows the advantages of Terminal Server in replicated site installations.

Terminal Server Advantage	Explanation
Remote Access	Using RAS (Remote Access Services), an administrator can use RDP client software on a local PC to gain complete access to Terminal Server administrative applications.
Reliability	Windows NT Terminal Server Edition is an architectural peer to UNIX.
Wide Choice of Terminals	Terminal Server can support a variety of thin clients. Using the MetaFrame add-on from Citrix Systems, Terminal Server can also support many handheld and wireless terminals.

Table 2. The advantages of deploying Terminal Server in replicated sites.

Small Office UNIX Installations

Small Office installations comprise more than 50 percent of the SCO UNIX installed base. These installations typically have 2 to 5 users and the primary requirements are reliability and access to applications needed for day-to-day tasks. Clients in this environment will range from dumb terminals to stand-alone PCs with host access software installed. The benefits of Terminal Server in this type of environment are summarized in Table 3.

Terminal Server Advantage	Explanation
Access To Applications	Most users want access to popular Windows-based applications such as Microsoft Office.
Reduced Total Cost of Ownership (TCO)	Terminal Server lowers the TCO through a reduction in hardware and software costs and reduced support costs through centralized management and simplified desktop devices.
Ease of Use	Using Terminal Server and thin clients, companies don't have to burden users with a complex operating system. Instead, they can use the familiar Windows interface

Improved Application Availability	Many more business applications are available for the Windows platform at a reasonable price when compared to the UNIX environment.

Table 3. The advantages of deploying Terminal Server in small offices.

Scalability

Terminal Server can be deployed on a handful of desktops or throughout an entire corporate organization. The general guidelines for planning your Terminal Server resource requirements are summarized in Table 4.

Projected Processor Requirements for Terminal Server	
Concurrent Users	**Processor**
25	Single 200 MHz Pentium Pro
60	Dual 200 MHz Pentium Pro
75	4-way 200 MHz Pentium Pro
100	6-way 200 MHz Pentium Pro
Projected Memory Requirements for Terminal Server	
Concurrent Users	**Memory**
25	256 MB
50	512 MB
75	768 MB
100	1 GB
Projected Bandwidth Requirements for Terminal Server (ICA and RDP Protocols)	
Concurrent Users	**Bandwidth**
2-4	60Kbps-80Kbps (ISDN)
75-100	1.5 Mbps-2.0 Mbps (T1)
100-500	10 Mbps (Ethernet LAN)
500-2000	40 Mbps (T3/Fast Ethernet)

Table 4. Projected processor, memory, and bandwidth requirements for Terminal Server.

For more information on the scalability and performance of Terminal Server and thin-client architectures, read the Microsoft/Citrix Scalability and Performance White Paper at http://www.microsoft.com/ntserver/basics/terminalserver/default.asp

Terminal Server Networking from a UNIX Perspective

The X-Window System environment is based on the typical distributed processing model: *client/server*. A fundamental concept of X-Window System is the separation of processing the algorithm (X Client) from the handling of user interactions on the display, keyboard and mouse (X Server). These are two different programs, which may run on the same physical processor or on two different systems. If they are separated on distinct machines, they require a communication link, preferably a

high-speed link, such as a LAN, but one could use asynchronous lines as well. The X Client application, which typically resides on the client, provides the following functions:

- Manages the output on the monitor
- Manages the keyboard and mouse input
- Manages the input/output queues
- Handles the communication between the X Server and the X Client
- Stores off-screen data
- Downloads fonts.

Figure 3 illustrates the flow of information through an X-Windows based system.

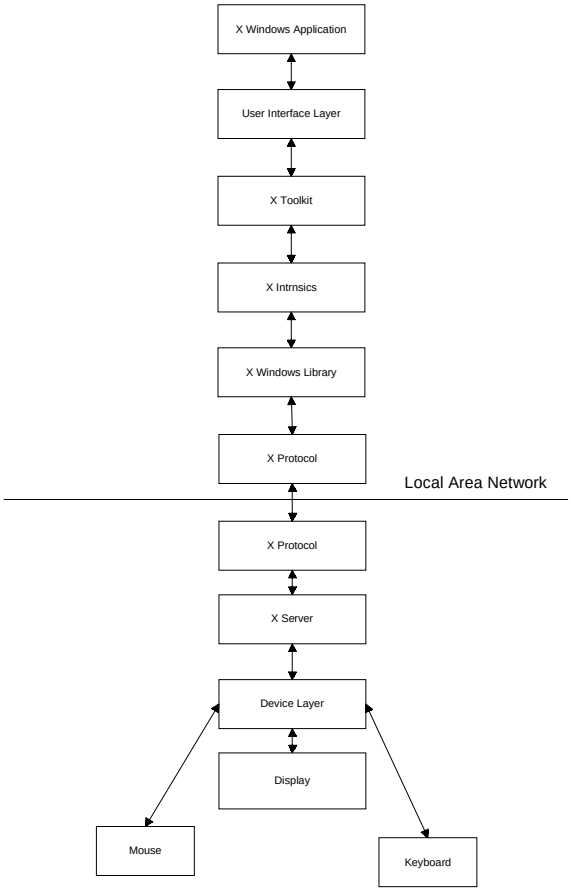


Figure 3. The X-Window System architecture.

The presentation of an X-Windows application is dependent of the X Server. The X-Window environment is more than just a network protocol; it's also an application development paradigm. Applications are developed using the X-Windows toolkit, just as Windows applications are developed using the Windows SDK. The two primary weaknesses in the X-Windows protocol are:

- It is largely dependent on the LAN throughput.
- The client requires a considerable amount of processing power to process the X-Windows protocol.

RDP and ICA Protocols

The RDP and ICA protocols separate an application's user interface from its logic, providing true location independence for Windows-based applications because the Windows application is run at one location and while the program's user interface is displayed somewhere else. Only the user interface, keystrokes, and mouse movement are transferred between the server and the client device over a network or communications protocol, so client resource consumption is minimized, as is the consumption of network bandwidth. This distributed Windows architecture allows Windows 16-bit, Windows 32-bit, and client/server applications to perform at very high speed over low bandwidth connections. It also allows 16- and 32-bit applications to run on older PCs as well as the new-generation of lightweight client devices.

RDP and ICA are robust and extensible protocols that include definitions for the following capabilities:

- Full-screen text presentation
- Graphical Windows application screen presentation
- Keyboard and mouse input
- Session control
- Framing for asynchronous connections
- Error detection and recovery
- Encryption
- Compression hooks
- File system redirection
- Print redirection
- Multiple generic virtual channels
- Cut and paste across sessions
- Local device support – drive mapping, local printer mapping, local COM port mapping
- General purpose WinFrame server browsing

Because RDP is an extension of the core T.120 series protocol, several other capabilities are retained, such as the architectural features necessary to support multi-point (multi-party sessions). Multi-point data delivery allows data from an application to be delivered "real-time" to multiple parties, without having to send the same data to each session individually. RDP currently runs only over TCP/IP (including RAS, PPP, PPTP, or any other standard TCP/IP compliant protocol).

On its own, Terminal Server does not provide client support for UNIX. Citrix's thin-

client/server system software for Terminal Server, MetaFrame, provides a high-performance, cost-effective, and secure way to deploy, manage and access business-critical applications throughout an enterprise -- regardless of client device or network connection. ICA runs over industry-standard network protocols, such as TCP/IP, NetBEUI, IPX/SPX, and PPP and industry-standard transport protocols, such as async, ISDN, Frame Relay and ATM (Asynchronous Transfer Mode). Table 5 provides an overview of the RDP and ICA protocols.

Feature	Description	RDP	ICA
Clients	32-bit client for Windows-based PCs (Windows 95, Windows 98, Windows NT Workstation/Server 3.51, Windows NT Workstation/Server 4.0)	√	√
	16-bit client for Windows for Workgroups 3.11	√	√
	16-bit client for older versions of Windows		√
	16-bit client for MS-DOS		√
	Unix client, Mac client, Java client, Windows CE client		√
	Browser client (Internet Explorer and Netscape)		√
Transport Protocol	TCP/IP	√	√
	SPX, IPX, NetBEUI and Direct Asynch		√
WAN connection	Connect client over Wide Area Network	√	√
LAN connection	Connect client over Local Area Network	√	√
Audio	System Sounds	√	√
	Support for .wav (Windows Audio) with 16-bit stereo. SoundBlaster Compatible		√
Local Printing	Printing to a local printer attached to a PC client	√**	√
	Printing to a local printer attached to a WBT or Non-Windows client		√
Local Drive Mapping	Manual re-mapping of local client drive to server drives	√**	√
	Automatic re-mapping of local drives by administrator		√
Local COM Port Redirection	Manual client COM port redirection for PC clients	√	√
	Automatic Com port redirection by administrator		√
Cut and Paste	Cut and paste of text between sessions		√

Web publishing	Publishing of windows-based applications to the web		√
Load balancing	Pooling of servers behind a single server address and for increased availability		√*
Application Access	User/Administrator configuration of applications for access	√	√
Shadowing	Administrator viewing or “remote control” of client session		√
Direct dial-up connections	Allows a WBT client to dial in directly to the server without using a dial-up service such as RAS		√
Bitmap caching	Allows client to cache display bitmaps for improved performance	√	√
Encryption	Multiple level encryption for security of client communications	√	√*
Automatic Client Update	Administrative means for updating client connection software from the server		√
Pre-Configured Client	Predefined client with published applications, phone numbers, IP addresses, server names and connections options		√

** Requires optional Load Balancing Services or SecureICA Services in addition to Citrix MetaFrame.*
*** Uses Windows native networking*

Table 5. Comparing RDP and ICA features

Comparing X11 and RDP/ICA Protocols

While the X11 and RDP/ICA protocols can both be used to deliver graphically based applications to desktops, their attributes are compared in Table 6.

Attributes	UNIX X11 Protocol	ICA/RDP Protocol
Graphics Performance	Higher performance with graphics applications, such as CAD systems	Lower performance with graphically intensive applications, high performance with traditional business applications such as Word and Microsoft Excel
CPU Load	Lower CPU requirements on the host system, higher CPU requirements on the clients	Higher CPU requirements on the host system, minimal CPU requirements on the clients
Connection Support	Supported under IP only	Supported under direct asynchronous, IPX, SPX,

		TCP/IP, SLIP/PPP, and NetBIOS
Bandwidth Requirements	Large amount of network traffic can be generated by graphic functions	Minimal network traffic to support typical business applications
Compression?	Not supported	Integrated within the protocol
Optimized For	Graphic intensive applications	Low bandwidth connections

Table 6. Contrasting UNIX and Terminal Server protocols.

Ease of use

Ease of use is a major concern for companies who are trying to get as much as they can from their computing platforms. Greater ease-of-use translates to greater productivity, less training expense, and a greater potential resource pool, all of which are significant benefits to the company – and also help to reduce the overall TCO.

Many users have had to become UNIX savvy because they had no other choice in a UNIX environment. Part of using the application or related tools required running shell-scripts or having some knowledge of the filesystem. With Terminal Server and thin clients, users can have the familiar Windows interface with common menu and dialog options, so new users quickly become efficient and effective.

In addition, because Terminal Server has an industry-standard set of APIs, technical users can easily integrate both custom and off-the-shelf applications into their environment. This standardized 'open' environment makes integration and development both more straightforward and less expensive.

Reliability

When deploying any solution into a store, branch office, or small business, reliability is a prime consideration. Typically, the on-site staff will know only how to run the applications associated with their job. Administrative or maintenance issues with the on-site server are handled by a regional support office or a local computer services company. In these environments, it is critical that the server operates reliably. Furthermore, reliability must be provided in a cost-effective manner.

NCR Corporation (http://www.ncr.com) has had extensive experience in deploying solutions based on Windows NT and UNIX, and has considered the reliability of both solutions in detail. Since actual field reliability data is not yet available for Terminal Server, NCR has analyzed the reliability of Terminal Server from an operating system feature level – and concludes that Terminal Server is a peer to UNIX solutions. Essential components of system reliability for Terminal Server and

UNIX can be categorized as Fault Detection, Fault Isolation, and Fault Recovery.

Fault Detection

The first line of defense in providing a highly reliable server is to detect low resource conditions so that corrective action can be taken before they become a problem. The operating system features needed to provide this capability are Error Logging and Resource Instrumentation.

From the first release of Windows NT, the Windows NT Event Log has been the central place for logging errors, warnings, and information messages for applications as well as the operating system. There is no standard corollary to the Windows NT Event Log among UNIX OS's. Some vendors have developed proprietary API's for driver and kernel error logging into the `/var/adm/streams` log. There is also no standard directory for applications to log error events – any application that performs error logging will create its own logfile. The UNIX administrator would typically write custom shell scripts to scan and filter these logs to detect abnormal events.

In addition, from the earliest releases of Windows NT, a rich set of performance counters has been provided. The administrator may configure the Windows NT Performance Monitor application to send alerts when key counters cross threshold values or may run Performance Monitor remotely to obtain "samples" of important performance counters. The UNIX corollary would be the use of system utilities such as `sar`, which provides the basic set of OS resource counters. The typical UNIX administrator would again be writing scripts to run the utilities and post-processing the output via additional scripts to filter out abnormal conditions.

Fault Isolation

Of prime concern in a multi-user operating system is how gracefully the OS handles a fatal fault in a given user's session. The differences between Windows NT Terminal Server Edition and UNIX in this area are implementation related – both have mature virtual memory subsystems that prevent a process from arbitrarily reading or writing the memory space of another process. Both also provide cleanup facilities to reclaim the memory of abnormally terminated processes and to handle outstanding I/O's. While a user may run an application that crashes, or hangs, this fault is isolated to user and does not bring down the entire system.

Fault Recovery

The Windows NT Filesystem (NTFS) is a journaling, or transactional file system. This means that any I/O that alters the file system data or meta-data (directory structure, etc.) is completed atomically so that either all of the changes are completed, or none of the changes is completed. This design means the transaction log can be used to restore the file system to a known good state after a system crash. In addition, NTFS keeps copies of vital file system information in multiple sectors for extra redundancy.

Most UNIX system vendors will provide a journaling file system such as the Veritas VxFS, which provide the same basic features as NTFS, including a transaction log and redundant metadata storage. It should be noted that while journaling file systems provide reliable recovery from system crashes, they do not protect data against all possible failures such as disk failures, controller failures, or software faults in the file system. Terminal Server provides native software RAID capability that can create a virtual disk drive having redundancy across multiple disk drives and multiple disk controllers.

Remote Administrative Access

Support for remote management and diagnostics allows enterprise system administrators to more quickly and cost-effectively maintain their systems, without the need to be physically present at the server location. The benefits of remote management and diagnostics—faster response to issues, simplified server management, the ability to use off-site technical specialists for problem solving or system repair—include less downtime and this more satisfied customers.

System administrators need the ability to define software policies that specify the applications, data, and desktop environment a user can access, and automatically update and synchronize applications, resources, and data on a per-computer or per-user basis. To meet these needs, Terminal Server includes remote management and diagnostics capabilities. For more information on the remote administration of Terminal Server sites, consult the Comparing Windows NT and UNIX Remote Management, available from the Terminal Server home page at http://www.microsoft.com/NTServer/Basics/TerminalServer/default.asp

INTEGRATING TERMINAL SERVER INTO UNIX-BASED ENVIRONMENTS

Deploying Windows-based applications in existing UNIX environments has traditionally been a no-win choice between cost and application performance. Either desktops are supplied with traditional PCs, requiring yet another PC client server, or Windows-based applications are provided through emulation. Typically emulation has a high performance penalty, and does not provide complete support for all native Windows applications. Terminal Server offers existing UNIX environments a new cost- effective choice to deploying Windows applications to their corporate desktops.

Access to Terminal Server can be provided through standalone Windows-based Terminals, Windows-based Terminal client applications running on existing PCs, or through the integration of Citrix's MetaFrame with Terminal Server. It's also possible to access Terminal Server from existing UNIX workstations and X-Windows Terminal servers.

Network Terminal Based Deployment

The Windows-based Terminal is the desktop device of choice in Terminal Server environments. The Windows-based Terminal consists of a standard graphics monitor, keyboard, and mouse and is capable of communicating with a Terminal Server via a serial connection, dial-up modem, or directly on the LAN, WAN, or wireless network. The Windows-based Terminal puts full Windows NT functionality on a workstation terminal and operates using either the RDP or ICA protocol. ICA- and RDP-compatible Windows-based Terminals are produced by a number of companies, including Network Computing Devices (http://www.ncd.com), Wyse Technology (http://www.wyse.com), Tektronix (http://www.tektronix.com), NeoWare Systems (http://www.neoware.com), and Boundless Technologies (http://www.boundless.com).

Citrix ICA-Based Clients

A thin client doesn't have to be truly thin in a Terminal Server environment. Terminal Server includes a client application that runs in a standard Windows environment (using the RDP protocol) which allows a standard PC to access a Terminal Server. With MetaFrame, Citrix includes UNIX and Java clients that run on UNIX workstations and X-Terminals (using the ICA protocol). If you're deploying Terminal Server applications in an environment with no existing hardware and no other PC requirements, you would likely install Windows-based Terminal devices. However, if you have existing PCs (386, 486, Pentium, and so forth), UNIX workstations, or X-Terminals, they all can be used as Windows-based Terminals, extending your initial investment. The client software is also appropriate for users that still need to access a standalone PC or UNIX workstations, but also need access to applications on Terminal Server.

Terminal Server vs. Windows Emulation

The need to run Windows-based applications under UNIX is not a new one. The UNIX world has offered up a number of solutions, including:

- **Wabi** A UNIX application that allows Microsoft Windows-based applications to run in several UNIX operating environments using the X-Window system. Wabi software is *middleware* – it acts as an interface between Windows and UNIX, translating the language of Microsoft Windows-based applications to the language of UNIX and the X-Window System. Wabi works by intercepting a Microsoft Windows application's request and then making an equivalent request in the UNIX environment to deliver the desired result. In slightly more concrete terms, an application uses the Windows API calls to open an icon, for example, and Wabi translates the request to equivalent X-Windows calls. A print request, or any other request involving a device, is translated and redirected to an appropriate UNIX command or device. Much of an application's "behind the scenes" activity involves Intel x86 instructions, which Wabi passes directly to the x86 processor. Wabi only provides emulation for 16-bit applications for Windows, and is not compatible with Windows 95/Windows 98, or with applications for Windows 95 and Windows 98 networking.

- **WINE** Both a program loader and an emulation library. The program loader loads and executes Windows-based application binaries, while the emulation library takes calls to Windows functions and translates these into calls to UNIX or X Windows, so that equivalent functionality is achieved. Windows binaries run directly; there is no need for machine-level emulation of program instructions. WINE compatibility includes Linux, NetBSD, FreeBSD and SCO UnixWare, and there is now support for SCO OpenServer. As with Wabi, WINE provides emulation for 16-bit applications for Windows only, and is not compatible with applications for Windows 95 and Windows 98 or networking.

- **SoftWindows 95** The only commercially available Windows emulation for UNIX, SoftWindows is also the only emulator providing Windows 95 and Windows 98-based application, multimedia, and networking support. SoftWindows 95 for UNIX is available for HP, SPARC (SunOS 4 and Solaris 2.5) and IBM workstations.

While the solutions offer a first step in the deployment of Windows applications under UNIX, they can't compete with Terminal Server in three important areas:

- **Management**: Centralized management drastically reduces the total cost of ownership.
- **Applications**: Corporations today are demanding universal access to applications, independent of platform and connectivity method.

- **Performance**: The performance of business applications must be acceptable regardless of connectivity method (LAN, WAN, Asynchronous, Internet). Businesses today depend on all employees, including those in the field, having acceptable access to critical applications.

Terminal Server and UNIX Enterprise File Sharing

When integrating Terminal Server and UNIX in the enterprise, it can be useful to create file shares that are accessible throughout the enterprise. Two third-party applications which integrate with Terminal Server make this possible, NetApp Multiprotocol Filer from Network Appliance, Inc., and Enterprise Data Server from Auspex Systems, Inc.

In enterprise-class environments of UNIX and Windows NT-based systems, Network Appliance (http://www.netapp.com) multi-protocol file-server appliances ('filers') enable high performance, reliable, and seamless data-sharing for Terminal Servers and its UNIX clients. This is especially useful for UNIX workstation customers that want to run Windows-based Terminals on their desktops using Terminal Server, while accessing files that reside in their home-directories via NFS (Network File System), as shown in Figure 4.

The file-sharing protocol on UNIX servers and workstations is NFS, so for Terminal Server to access those files, there are two options:

- Store the files on a Network Appliance filer and access them through the Windows NT native file-sharing protocol, CIFS (Common Internet File System – formerly SMB – Server Message Block).
- Install NFS client software on Windows NT Server running Terminal Server clients to access the files on the NFS server.

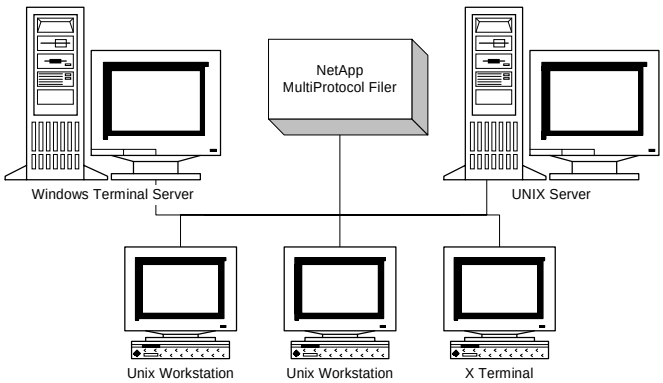


Figure 4. Deploying NetApp in a Terminal Server/UNIX environment.

CIFS is an Internet file system, based on the native file system used on Microsoft Networks. It is being targeted as a replacement for both FTP and NFS file system

services and allows to directly mount CIFS remote system as a directory or drive. Table 7 compares the features of the CIFS and NFS file systems.

Feature	CIFS	NFS
Can be mounted as your local drive	Yes	Yes
Supports Encrypted Passwords	Yes	No
Supports Unicode file names	Yes	No
Allows secure anonymous access	Yes	No
No Extra Software required for file transfers	Yes	Yes
No extra drivers required for Windows 3.11	Yes	No
No extra drivers required for Windows95	Yes	No
No extra drivers required for Windows NT	Yes	Yes
No extra drivers required for UNIX	Yes	Yes
Fault Tolerance – connection can be auto-restored	Yes	Yes
Used for both Internet and LAN networks	Yes	No

Table 7. Features of CIFS and NFS.

Network Appliance filers offer faster performance, native CIFS compatibility, and file locking across UNIX and Windows NT. Since the filers maintain both UNIX and Windows NT style security attributes for all files, file-security is not compromised. Additionally, on adding Windows NT-based workstations to the environment, the same files are accessible via CIFS, and do not require installation of any NFS client software.

The Enterprise Data Server from Auspex Systems, Inc (http://www.auspex.com), with both NFS and CIFS serving capability, provides a centralized data store accessible to both Terminal Server and UNIX servers. Enterprise Data Server uses a dedicated back end file server, in a three-tier architecture as shown in Figure 5.

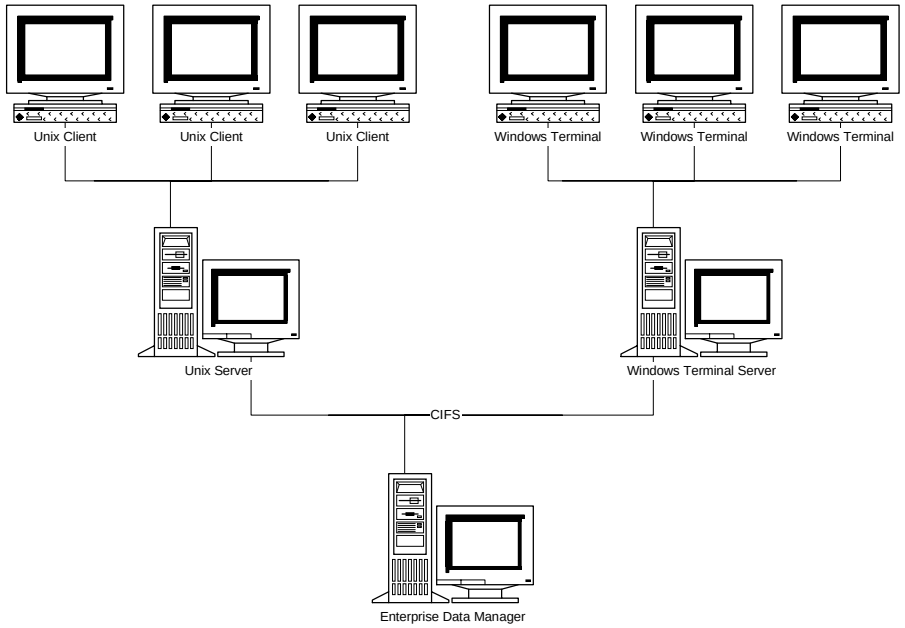


Figure 5. Deploying Enterprise Data Server in a Terminal Server/UNIX environment.

The Enterprise Data Server holds both the executables and the file data, and will create shares to the Terminal Servers through CIFS, their native protocol. The advantages of this approach are:

- One server to backup, greatly easing the systems management burden
- One master copy of applications executables, so updates are simplified
- Each resource is used in an optimum way; applications servers run applications and data servers manage and deliver data
- Enterprise Data Server can scale to terabyte and up capacities; you don't need to add more servers as data grows
- Enterprise Data Server is fully transparent to the Terminal Server and clients.

Migrating Existing UNIX Applications

Migrating existing UNIX applications into a Terminal Server environment can viewed on a continuum – from a Windows-centric approach to a UNIX-centric approach, as shown in Figure 6. The most Windows-centric migration approach would be to convert existing UNIX applications into pure Win32® applications. The primary benefit of this approach is that in the end you have applications that can fully take advantage of the Windows architecture and interface. The downside is that for complex applications, a rewrite may require considerable effort. The most UNIX-centric approach to application migration is, not surprisingly, to do nothing. Just continue to use your UNIX applications as they are and access Windows applications using Citrix MetaFrame and the Citrix Java or UNIX client. The primary benefit of this approach is that it requires the least amount of migration effort) since you're keeping the status quo with your existing UNIX applications and not

changing anything on your current desktops. The downside of this UNIX-centric approach is that you won't reap all of the TCO benefits provided by thin-client architectures.

There are a number of third-party tools and utilities to ease the migration – from applications that integrate your existing UNIX applications into a Terminal Server environment to tools that help convert current existing UNIX applications to run under Terminal Server. Options along this UNIX-Terminal Server migration continuum are shown in Figure 6.

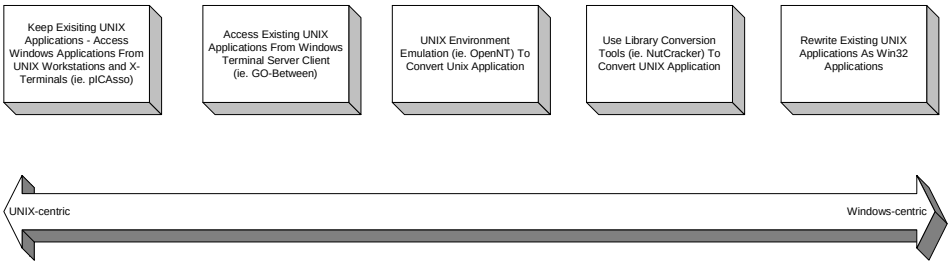


Figure 6. The continuum of UNIX-to-Terminal Server application migration.

Accessing Existing UNIX Applications from Terminal Server

While it is possible to use Terminal Server and Citrix MetaFrame to deploy applications for Windows to UNIX desktops and X-terminals, you may want access existing UNIX systems using Windows-based Terminals. Deploying a third party application, GO-Between from GraphOn Corporation (http://www.graphon.com), access to X-Window applications can be provided from any Terminal Server thin client. As shown in Figure 7, GO-Between centralizes X-Windows application processing on a UNIX host. Using a proprietary protocol called RapidX, a lightweight protocol that carries only application screen updates and application user interaction, GO-Between executes X-Windows applications on the UNIX host and sends just the necessary screen updates down the wire. Deploying a small RapidX client (less than 300 KB) on your Terminal Server, you have complete access to your UNIX X-Windows applications.

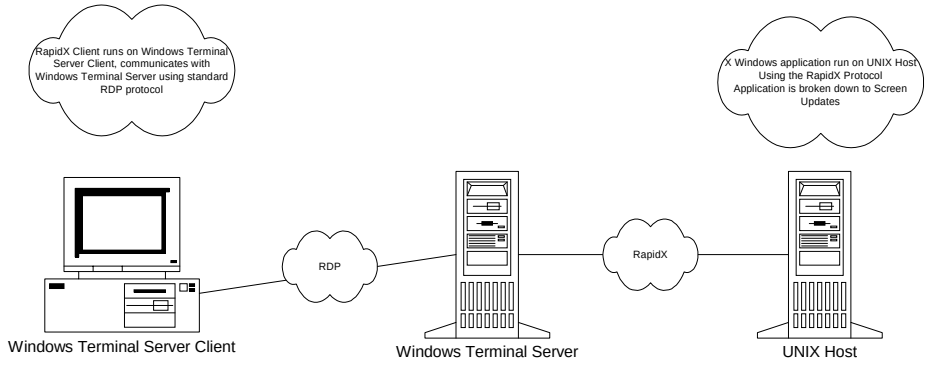


Figure 7. Deploying GO-Between in a Terminal Server/UNIX environmnet.

With this approach to UNIX X-Windows application access, the benefits include:

- All the processing of the X-Windows application interface is centralized on the host, running in its native environment.
- The RapidX lightweight protocol eliminates all X-Windows processing overhead and allows the client deployed on Terminal Server to be small, improving scalability and network performance.
- While GO-Between can work in conjunction with the ICA protocol, MetaFrame is not required, because the RapidX client is run on the Terminal Server. GO-Between is completely compatible with the RDP protocol.
- No software is required on the Windows Terminal Client.

GlobalHost is required on the UNIX server. It will be shipped as part of the standard Sun Solaris operating system and as part of the IBM AIX operating system later this year. GlobalHost will take advantage of all the extensions that are implemented in the standard Sun and IBM X servers. For example, Sun Microsystems implements Adobe PostScript in its X Server and GlobalHost can take advantage of that – Windows Terminal Clients will have complete access to that UNIX extension. GO-Global is 100 percent compatible with X11 and Terminal Server. There is no software modification required on either the UNIX host or the Terminal Server.

Using UNIX Environment Emulation

OpenNT 2.1 from Softway Systems, Inc. (http://www.opennt.com) permits existing applications developed for the UNIX environment to be ported to Terminal Server and Windows NT just by recompilation. Terminal Server is directly based on Windows NT, so both support multiple environment subsystems including the POSIX subsystem. The Microsoft POSIX subsystem is an exact implementation of the ISO/IEEE POSIX.1 standard, which includes the ANSI C library by reference.

The overall architecture of OpenNT consists of the environment subsystem, its mapping to NTFS, its relationship to functional subsystems (security, for example), and its compiler environment. The OpenNT subsystem has three parts:

- A subsystem that functionally re-maps the Windows NT kernel and manages such items as process relationships and signal delivery.
- A terminal session manager that manages the desktop console window for each OpenNT-based session leader.
- A dynamic link library that handles certain system service requests directly, as well as the communication between the subsystem and OpenNT processes.

The OpenNT subsystem is a peer environment to the Win32 subsystem. OpenNT processes communicate with the OpenNT subsystem using the same mechanisms that the Win32 subsystem processes use to communicate with the Win32 subsystem. Applications that run as clients of one environment subsystem cannot make calls to interfaces supported by another environment subsystem, so

separate libraries are provided to ensure no dependencies to the Win32 world are referenced within the libraries.

OpenNT provides a complete environment for building UNIX applications made up of header files, libraries, and shell script wrappers around the command-line version of the Microsoft Visual C++ ® development system compiler and linker. Using the OpenNT development environment, you can port your applications using familiar UNIX tools and command-line syntax.

Using Library Conversion Tools

NuTCRACKER from Data Focus, Inc. (http://www.datafocus.com) provides an enterprise UNIX operating environment directly on the Win32 subsystem of Windows NT. It includes UNIX libraries, commands, utilities (including shells and other standard UNIX applications), and an X server. This permits existing applications developed for the UNIX environment to be ported over to Terminal Server/Windows NT by simply recompiling them. UNIX applications ported by NuTCRACKER to Terminal Server run as native Win32-based applications, and can take advantage of all Windows features, including COM, Windows Help, the complete Windows API, and the registry. NuTCRACKER can also be used to port UNIX daemon process into Windows NT-compliant services. NuTCRACKER can port UNIX based C, C++, or Fortran applications to Terminal Server.

The primary NuTCRACKER component is the NuTCRACKER Operating Environment Plug-in, which is a DLL that provides UNIX support to your ported application. The NuTCRACKER operating environment is a set of Dynamic Link Libraries that must be licensed for each client and server machine your application is ported to. NuTCRACKER provides full support for porting XWindows (Motif and X11R6) applications, including an XServer, support for OpenGL graphics, and XRT PDS 3d-Widget support. NuTCRACKER also provides Wintif – a set of Motif replacement libraries, which can provide a complete Windows interface to your XWindows applications.

NuTCRACKER provides complete UNIX style application support for:

- Process control
- Process identification
- Memory management
- Signal management
- Security
- Users and groups
- File security (when using Windows NTFS)
- File management, I/O, and control
- Path names
- Devices
- X11R6 and Motif

- Interprocess communication and networking

SUMMARY

Windows NT Terminal Server Edition offers organizations the opportunity to deploy native 16- and 32-bit applications for Windows to UNIX-based desktops, without the need to sacrifice performance or application compatibility. With Terminal Server clients, users can do everything they could with dedicated PCs, but at a lower cost. Using a number of commercially available add-ons, Terminal Server can be fully integrated into existing UNIX environments providing:

- Access to Windows-based applications from UNIX workstations and X-Windows Terminals.
- Access to existing UNIX X-Windows applications from thin-client Windows-based Terminals.
- Access to UNIX host resources such as the file system and printers.

Windows NT Terminal Server Edition provides centralized management and application deployment. Terminal Server provides access to the latest Windows applications from Windows Terminals and other RDP-compatible devices. When used in conjunction with MetaFrame, it's accessible from Windows-based Terminals, Network PCs, Java Network Clients, UNIX workstations and X-terminals, and wireless handheld PCs. Terminal Server is fully scalable and can be used to provide access to mission-critical applications across the enterprise.

For More Information

For the latest information on Windows NT Terminal Server Edition, check out our World Wide Web site at http://www.microsoft.com/ntserver/guide/hydra.asp.

GLOSSARY OF TERMS

Bandwidth

The range of electrical frequencies that a device can handle. The amount of bandwidth a channel is capable of carrying is equivalent to how much capacity is possible.

CIFS

A standard remote file system access protocol enabling groups of users to work together and share documents across the Internet or within intranets. CIFS is an open, cross-platform technology based on the native file-sharing protocols built into Microsoft Windows and supported on dozens of other platforms, including UNIX.

Component Object Model (COM)

A software architecture that allows applications to be built from binary software components. COM is the underlying architecture that forms the foundation for higher-level software services, like those provided by OLE. OLE services span various aspects of commonly needed system functionality, including compound documents, custom controls, interapplication scripting, data transfer, and other software interactions. Hewlett Packard, Digital Equipment Corporation, Siemens-Nixdorf, and Silicon Graphics ship or have announced plans to ship COM on their UNIX systems.

Ethernet

A local area network protocol that connects computing devices, printers, and terminals. Ethernet operates over twisted-pair, coaxial and fiber optic cable at speeds at 10 Mbps.

Frame Relay

A form of packet switching using smaller packets and less error checking than traditional forms of packet switching (such as X.25). An International standard for efficiently handling high-speed, bursty data over wide area networks. Frame Relay has traditionally been used strictly for data applications, but new technology advancements are now allowing limited voice and video applications to run over Frame Relay networks.

ICA (Independent Computing Architecture)

Developed by Citrix, a general-purpose presentation services protocol for Microsoft Windows. Conceptually, ICA is similar to the UNIX X-Windows protocol. ICA allows an application's logic to execute on a WinFrame multi-user Windows application server, located on the LAN. Only the user interface, keystrokes, and mouse movement are transferred between the server and the client device over any network or communications protocol, resulting in minimal client resource consumption. ICA is designed to run over industry-standard network protocols, such as TCP/IP, NetBEUI, IPX/SPX, and PPP and industry-standard transport protocols,

such as async, ISDN, Frame Relay and ATM. The ICA protocol presents only the user interface from an executing machine on the display of another machine.

Network File System (NFS)
A public standard that allows clients to access remote files stored on computers of different types. NFS provides "transparent" user access and users can manipulate shared files as if they were stored locally.

POSIX
Defines a standard operating system interface and environment to support application portability at the source code level. If you include a POSIX call in your program, then your program will work on any system that is POSIX compliant. Since POSIX has its origins in UNIX, most UNIX systems are already fully or nearly, POSIX compliant. Windows NT is fully POSIX compliant.

Remote Desktop Protocol (RDP)
An application sharing protocol (previously known as T.SHARE) currently used in Microsoft's NetMeeting and supported by Terminal Server.

TCP/IP (Transmission Control Protocol/Internet Protocol)
A set of protocols that link dissimilar computers across networks.

TCO (Total Cost of Ownership)
The cost of procuring, deploying and maintaining a management information system.

Windows NT Server 4.0, Terminal Server Edition
Adds Microsoft Windows-based Terminal support to the Microsoft Windows NT Server operating system and a “super-thin client” to the Windows operating system.

X-Windows
The standard protocol for graphically based UNIX applications. The X-Window environment is based on the typical distributed processing model: *client/server*. A fundamental concept of X-Window System is the separation of processing the algorithm (X Client) from the handling of user interactions on the display, keyboard and mouse (X Server).

APPENDIX A: CONSIDERING THE TOTAL COST OF OWNERSHIP (TCO) OF THIN CLIENT SOLUTIONS

To appreciate the savings of thin-client architecture across the enterprise you must consider total cost of ownership. The Total Cost of Ownership (TCO) is defined as the cost of obtaining, deploying and maintaining your management information systems. Several recent studies have determined that the typical TCO per client in traditional PC LAN-based environments is in the range of $8,000 to $15,000 per year. While these figures are debatable, there is consensus that the cost of maintaining a traditional PC LAN environment is quite high.

Thin Client technologies help to lower significantly the TCO through a reduction in hardware and software costs and reduced support costs through centralized management and simplified desktop devices. Unfortunately, TCO is a very elusive number that varies dramatically based on factors specific to individual companies. There is debate over what ingredients should be used in the TCO calculation, but the following list represents the major components that you should consider:

- Hardware and software costs
- System, storage and network management
- General operations and maintenance
- Help desk
- Employee (self and peer) support
- Other hidden costs

Thin Client technologies provide a reduction in the TCO through metrics that are fairly easy to measure. Thin Client desktop devices are now below the $500 price point and provide a robust interface on a standardized platform that is very easily maintained. Thin Clients often require fat servers and/or faster networks since many of the traditional tasks have been off loaded to centralized devices for processing and storage. Again however, the increased server and network costs are easy to calculate, and most, if not all, of these expenses can be offset through the cost savings associated with the low-cost desktop devices.

The real TCO savings in Thin Client environments are found in the reduction in total support costs. Thin Client environments are designed to centralize and leverage management tasks and to reduce the need to visit the desktop. The desktop devices themselves are highly standardized, easily replaced when necessary, and require very little configuration. All applications can be installed and deployed from a single server – further reducing the headaches of installing and configuring applications for each desktop. Of course, the actual TCO savings will depend on the environment and the type of Thin Clients being deployed (Windows-based

Terminals or NetPCs).

Terminal Server also supports “shadowing” which allows Help Desk staff to become part of a user’s session so even complex questions can be quickly and efficiently sorted out.

Thin Clients also provide consistent response times from critical applications whether connected directly to the LAN, at a remote WAN location, connected through the Internet or dialed in over standard analog lines. This benefit translates into savings that are important if not easily quantifiable.