

2015  
**HP 3000 and NT Interoperability**  
**Lars Appel, Alvina Nishimoto**  
**19111 Pruneridge Ave.**  
**Cupertino, CA 95014**  
**(+49 2102) 90-6743, (408) 447-5649**

The best solution to an enterprise's technology concerns is often to use more than one platform and to leverage the strengths of each platform where it makes sense to do so. MPE/iX is not standing still amidst the changing computing arena. MPE/iX is continuing to provide interoperability and data exchange with other operating systems, including NT and UNIX systems.

With the current explosion of NT in business environments, there has been a lot of interest in how your HP 3000 will interoperate with NT in your environment. This paper will outline various methods that are available to interoperate your HP 3000 with NT. Some of these topics include

- File and print sharing -- Samba
- Current Trends in Application Development
  - \* "Facelift" Phase  
GUI front ends
  - \* "Migration" Phase  
Database Interoperability - ODBC Interfaces, JAVA and JDBC
  - \* "Re-architecting" Phase  
Database Interoperability - Database Gateways, Middleware

#### **File and Print Sharing -- Samba**

One of the most popular file and print sharing methods with NT is the UNIX-based Samba. Samba has been ported to the HP 3000 and provides an easy mechanism for file and print sharing the HP 3000 and NT platforms

Samba for UNIX has been created by Andrew Tridgell in Australia. It is basically a freeware implementation of Microsoft's SMB protocol, which stands for Server Message Block and defines file and printer sharing for PC networks. The protocol is used by Microsoft Lan Manager and compatible products like Advanced Server for UNIX or Resource Sharing for MPE/iX.

PC's have built-in SMB support with Windows for Workgroups, Windows 95 and Windows NT. For Windows 3.x you can obtain free add-on software at the Microsoft ftp site that provides at least the client functionality.

The Samba package contains programs like SMBD and NMBD to provide server functionality and also contains utilities like SMBCLIENT or NMBLOOKUP to provide client functionality.

To set up your 3000 as a Samba server you have to create a configuration file. This file defines which directory trees of the MPE file system should be accessible by clients. Figure 1 shows an example. There is a global section for general parameters and defaults and one section for each service (also known as share) which defines attributes like associated directory path, read or write access, which users are allowed to access this share etc.

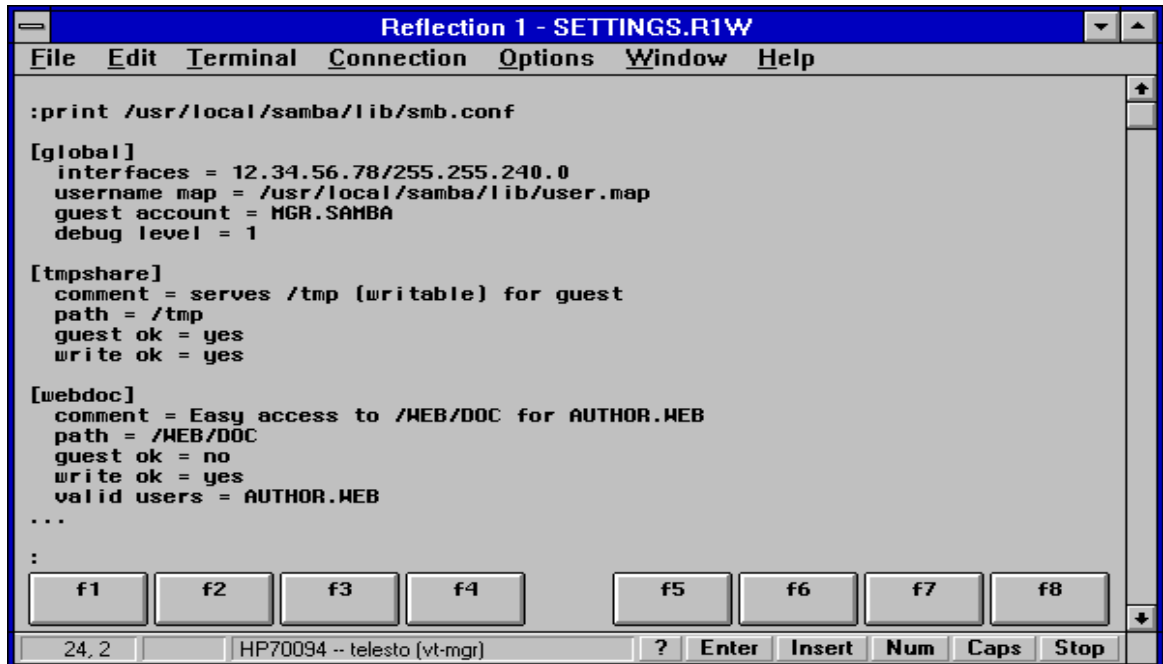
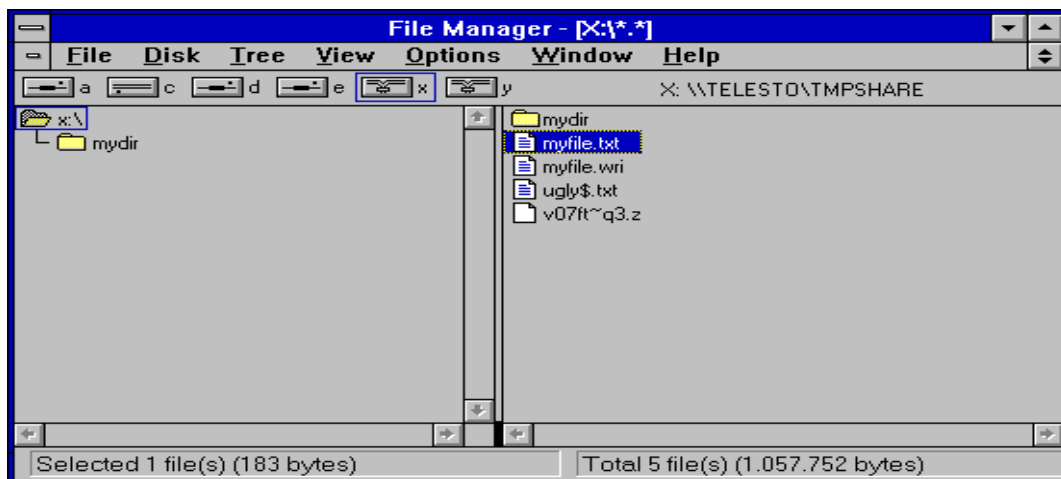


Figure 1 - Samba Server, Sample Config File

A listener process is then started (either under INETD or as a separate job) that waits for incoming client connection requests and creates child processes (servers) as needed. The server side validates the username and password which are sent by the client and grants access to the requested share if appropriate. A share may also be configured to allow guest access i.e. without a valid username/password pair specified by the client. This is similar to anonymous ftp in concept.

Figure 2 shows how you can connect to your server shares using the Windows File Manager. The Disk menu includes an item Network Connections which brings up the small windows shown in the front. You can either connect a network drive by explicitly typing in a share name (with the \\server\share syntax) or by using the browsing pane of the same window. Here you can view which workgroups are available on the network or subnet, click on a workgroup to see computers that belong to it, click on a computer to see available shares and click on a share to select it for connecting. The browsing feature required the NMBD server process running on the 3000. Without NMBD being active or when the 3000 is not on the PC subnet you can still type in explicit share names.



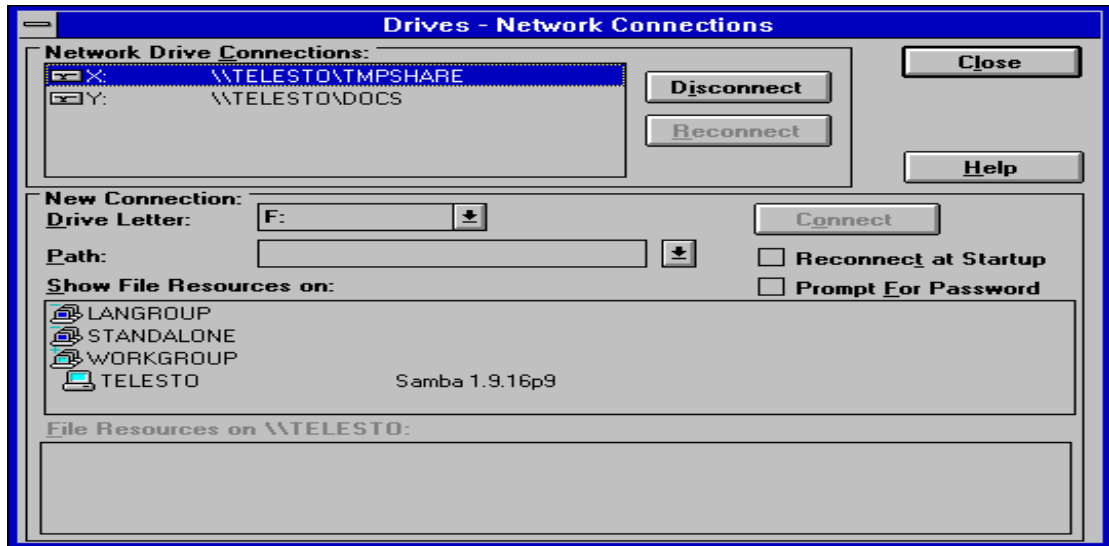


Figure 2 - Windows File Manager

These network drives are then accessible to regular PC applications. Figure 3 shows creating a text in the Windows Write editor and saving it to the X: drive as myfile.txt then.

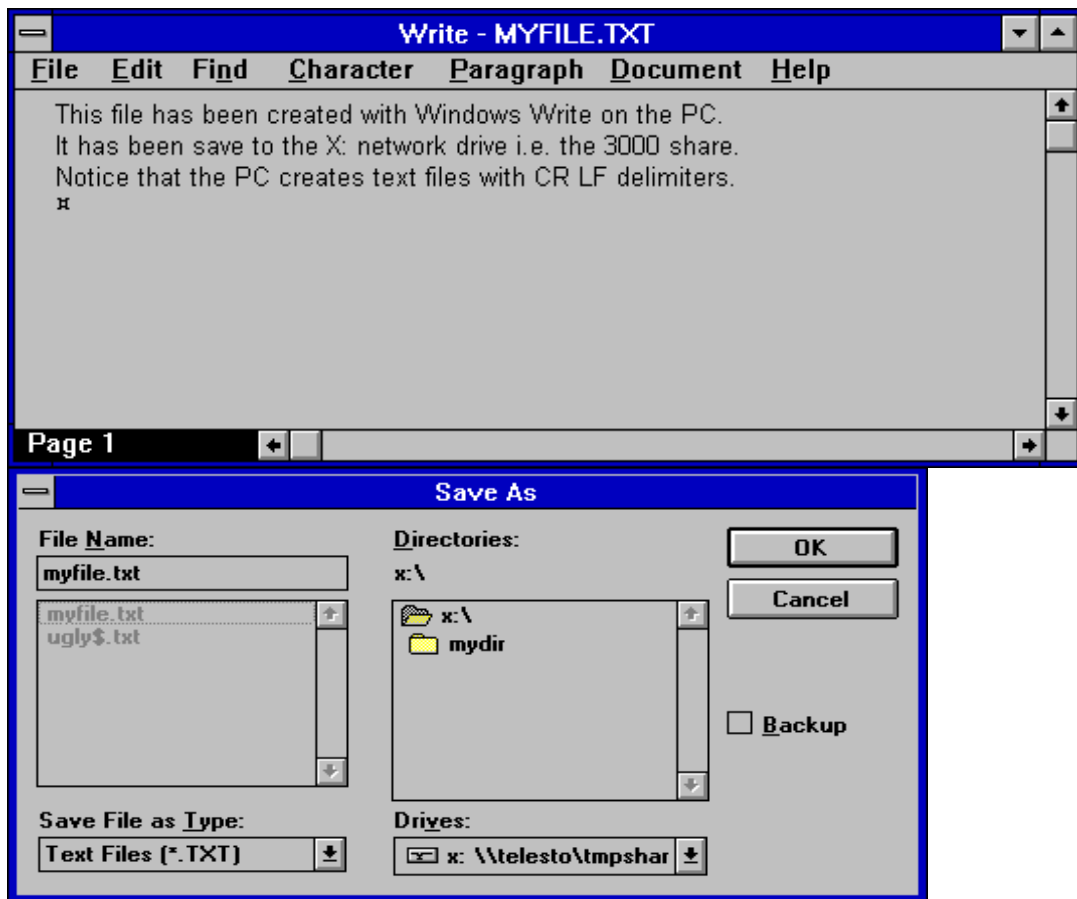
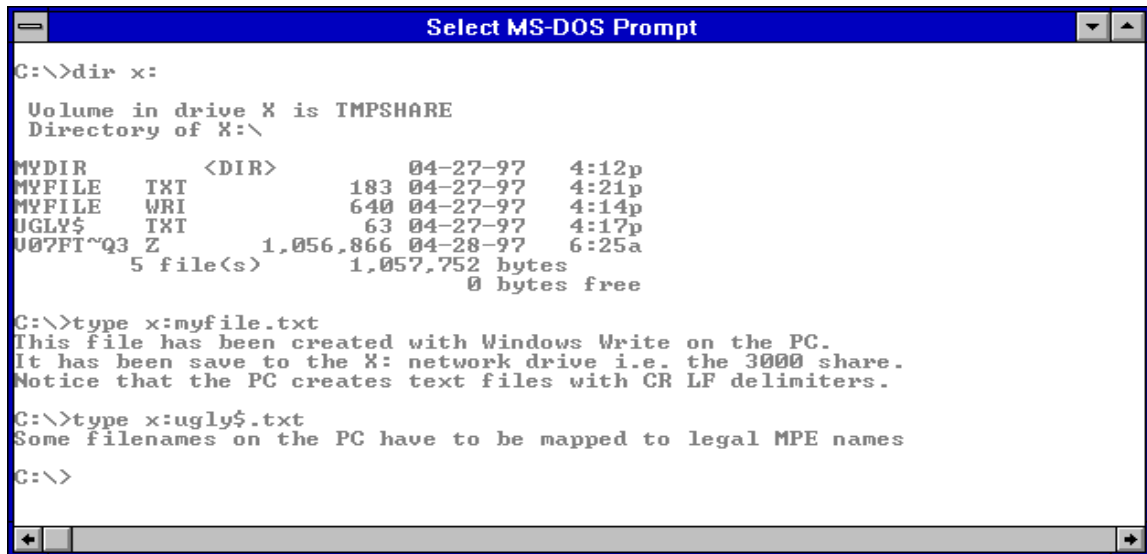


Figure 3 - Client Side, Windows Write

Figure 4 shows a few DOS commands executed on the network drive. The DIR command shows that there is a directory as well as a number of files, including our myfile.txt and the type command displays

the contents of myfile.txt once more. All the files and directories on such a network drive reside on the 3000 in the MPE file system (typically using the HFS name space).



```
C:\>dir x:

Volume in drive X is TMP SHARE
Directory of X:\

MYDIR          <DIR>              04-27-97    4:12p
MYFILE.TXT     183 04-27-97    4:21p
MYFILE.WRI     640 04-27-97    4:14p
UGLY$.TXT      63 04-27-97    4:17p
U07FT~Q3.Z     1,056,866 04-28-97    6:25a
               5 file(s)          1,057,752 bytes
                                   0 bytes free

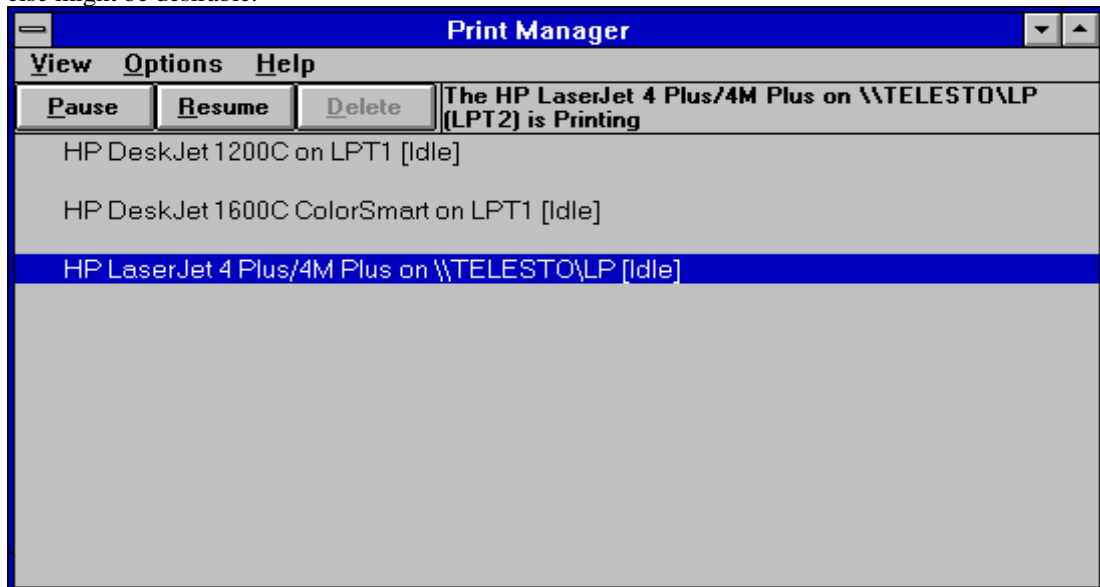
C:\>type x:myfile.txt
This file has been created with Windows Write on the PC.
It has been save to the X: network drive i.e. the 3000 share.
Notice that the PC creates text files with CR LF delimiters.

C:\>type x:ugly$.txt
Some filenames on the PC have to be mapped to legal MPE names

C:\>
```

Figure 4 - Client Side, DOS Prompt

Figure 5 shows that you can also access server based printers using Samba. You connect or disconnect with a dialog box similar to the one seen on the previous slide. Printer sharing does basically work in such a way that the client creates a file on the server directory associated with the printer and then lets the server process trigger a configurable command to push that file into the MPE spooler or whatever else might be desirable.



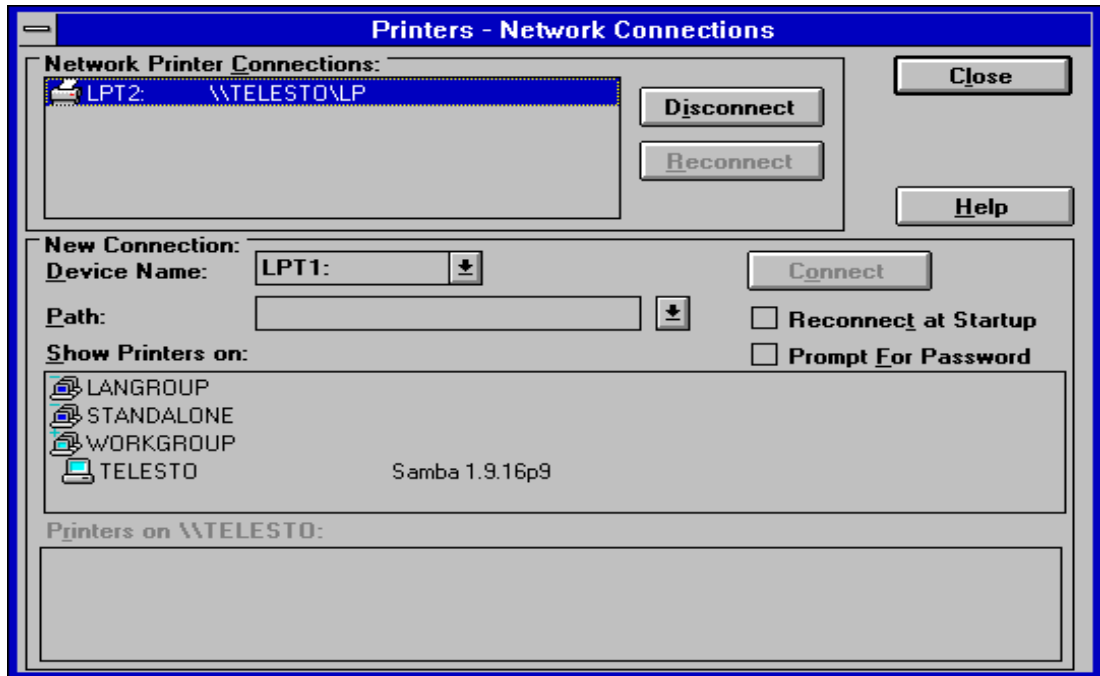


Figure 5 - Windows Print Manager

### Current Trends in Application Development

With the explosion of the Internet, Microsoft has embraced the Internet. In addition, the vendors that provide integration tools have embraced both the Internet and NT. Consequently, it is very difficult to separate a discussion on NT interoperability from a discussion on Internet application development. The next few sections will discuss tools that provide NT interoperability while at the same time providing web access.

New technologies such as NT and the Internet evolve in current business environments. Figure 6 describes this evolution to the Internet which involves three distinct phases. These phases are similar to the evolution that occurs with most new technologies. These phases include:

- The “Facelift” phase  
This initial phase provides business benefit with very little investment. This phase involves providing a new user interface with little or no code changes to existing applications. You can try out new technology with very little investment and hopefully provide business benefit and productivity improvements at the same time. This new interface allows you to get results to your customers quickly and to preview new technology, while still leveraging the reliability and stability of existing applications.
- The “Migration” phase  
In this phase, the gradual migration to new technologies starts to occur, and the foundation of a new infrastructure which is needed for re-architecting is laid. Initial migration occurs in modules that are straightforward to implement, make business sense, and provide additional value. This phase also involves creating new applications using the new evolving infrastructure.
- The “Re-architecting” phase  
In the final phase, the application infrastructure has been re-architectured and the last set of modules are being migrated. These modules are usually the most complex and considered to be mission-critical applications.

## Evolution in Software Development and the Internet/Intranet

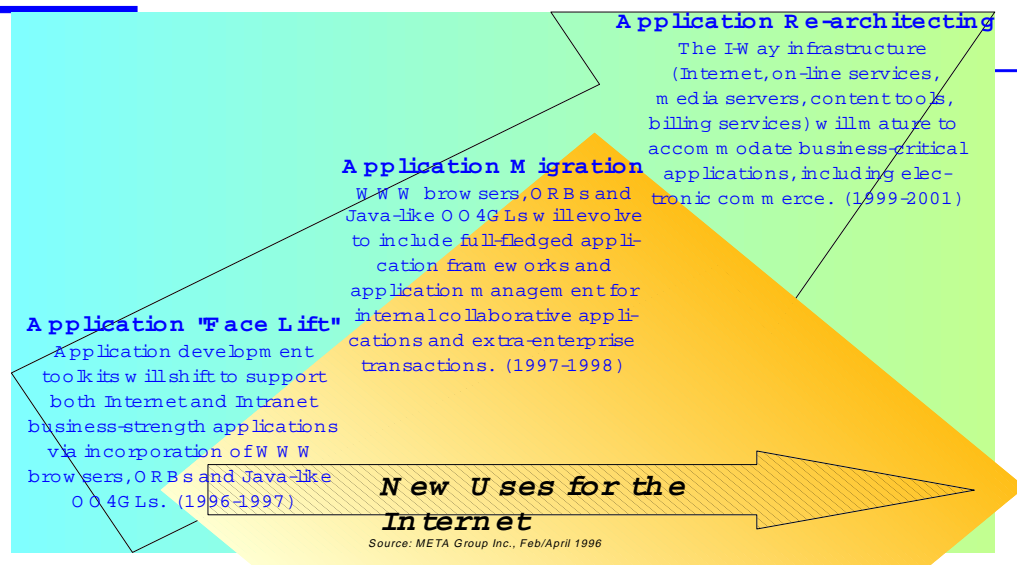


Figure 6

### GUI Front Ends

In the case of the HP 3000, the “Face Lift” phase usually consists of leveraging your existing application by putting a GUI front end on top of an existing VPLUS application. VPLUS is a proprietary screen and forms management package on the HP 3000. Because it is proprietary, VPLUS is often seen as a roadblock to interoperability. Several third-party tools exist to help provide interoperability between VPLUS applications and new interfaces. Figure 7 is a pictorial view of how these front end tools work with the HP 3000.

### HP 3000 GUI Front End Interfaces

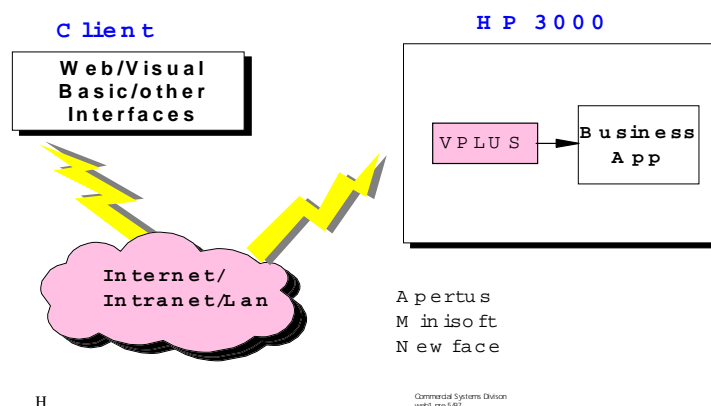


Figure 7

Apertus Technologies offers Enterprise/Access, a tool for building server-based solutions to access applications on traditional systems like the HP 3000 without modification to the host systems or applications. This allows you to put a Visual Basic or Web front-end on top of your existing applications. The Enterprise/Access product does not intercept VPLUS calls. It has less dependence on

VPLUS. Instead it works with what is displayed to the terminal screen instead. Enterprise/Access requires an HP 9000 to access HP 3000 VPLUS transactions.

Minisoft's FrontMan automatically converts applications based on VPLUS to a Windows-based client/server environment or a web front end. The HP 3000 code remains untouched while FrontMan intercepts the VPLUS calls and handles all requirements of the user interface. FrontMan also allows you to combine data from widely different environments into a single, coherent interface. It allows you to access relational databases, VPLUS applications, and HP IMAGE databases using a single user interface.

### Database Interoperability

In the "Migration" phase, most people wrestle with problems beyond the user interface and start to deal with database interoperability. Because many organizations may already have both MPE/iX and NT systems, evaluating how to access and share data across platforms is often a pressing concern. Relational database management systems or specific interoperability subsystems have the functionality necessary to access data across platforms. Interoperability features free users and developers from managing the complexity of establishing connections, handling the communication, reporting any errors, translating data, and disconnecting from remote databases.

### ODBC Interfaces

An example of a common database interoperability method with NT is ODBC. Because ODBC is widely used throughout the industry to communicate with different database management systems, developers are saved from having to learn multiple query dialects or APIs and complex networking requirements. ODBC is Microsoft's implementation of one part of the XOPEN standard for database access, SQL Access. ODBC is Microsoft's C language SQL API based on ANSI SQL. ODBC has become so popular that virtually every windows-based product works with it. Using ODBC allows you to take advantage of myriad of simple, easy-to-use GUI development environments. A new 32-bit ODBC driver will soon be available and will enhance the tools offering. We at HP continue to work with a variety of third-party vendors to provide tools that work with HP IMAGE database. Figure 8 is a pictorial view of how ODBC works with the HP 3000.

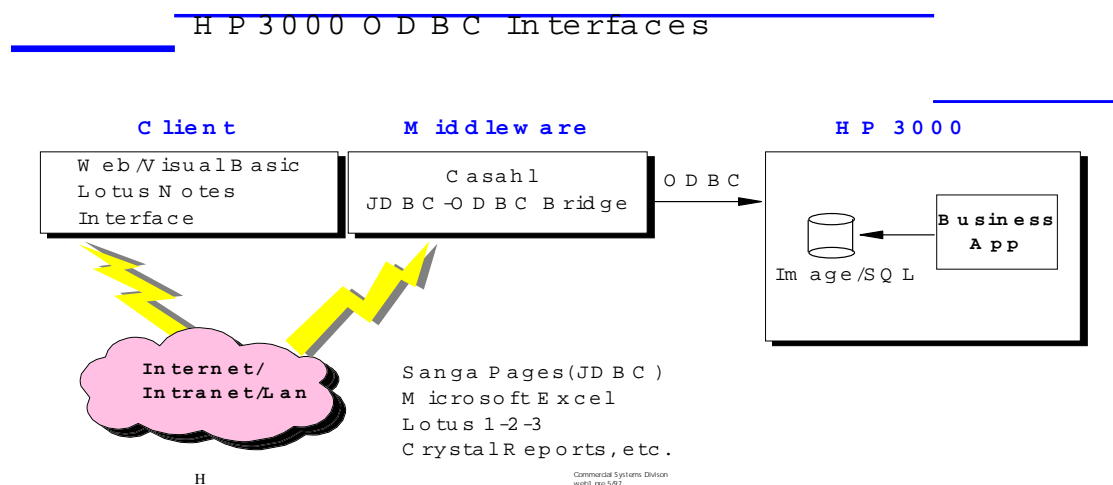


Figure 8

When working with ODBC there are several bridges/middleware tools that help in specific types of applications. For JAVA based tools, the JDBC-ODBC bridge available from Javasoft allows ODBC-enabled databases to be used by JAVA development tools. An example of a tool that support JAVA

and JDBC is Sanga Pages from Sanga International. JAVA and JDBC will be discussed in the next section.

For Lotus Notes, Casahl Technology's Replic-Action product provides the ability to define and access HP IMAGE and ALLBASE databases in Lotus Notes using an ODBC interface.

### **JAVA and JDBC**

JAVA is another emerging method for database interoperability with NT. JAVA is being popular because of its platform independence. The following two paragraphs, taken from Javasoft's web page, describes in a nutshell what JDBC is:

*The JDBC API defines JAVA classes to represent database connections, SQL statements, result sets, database metadata, etc. It allows a JAVA programmer to issue SQL statements and process the results. JDBC is the primary API for database access in JAVA . . .*

*The JDBC API is implemented via a driver manager that can support multiple drivers connecting to different databases. JDBC drivers can be written in JAVA so that they can be downloaded as part of an applet, or they can be implemented using native methods to bridge to existing database access libraries.*

Although the HP 3000 has the JAVA virtual machine, there is currently no native JDBC driver for HP IMAGE and ALLBASE written completely in JAVA. Today the way to access IMAGE and ALLBASE data from a JAVA program is to connect to ODBC data-sources via the JDBC-ODBC bridge, which provides a translation from JDBC to ODBC.

The JDBC to ODBC bridge is downloadable from <http://splash.javasoft.com> and the SDK is available at <http://www.javasoft.com>.

For Native IMAGE intrinsic calls, Advanced Network Systems, Inc.'s ADBC Developers' Kit links browser applets and server applications written for the MPE/iX Java Virtual Machine directly to TurboIMAGE.

### **Database Gateways**

In the final "Re-architecting" phase, you may have a major module which is on NT which now needs to read and write to HP IMAGE and ALLBASE databases. In this scenario HP IMAGE and ALLBASE databases need to be able to share data, run reports, and handle queries with other databases such as SQL Server, Oracle, Sybase, and others. Various gateways are provided by database vendors to allow for this type of interoperability.

#### **Connecting with Oracle Databases**

If you use Oracle's RDBMS on your HP 3000 and if you need to interoperate with Oracle databases on other platforms, you can use Oracle's SQL\*NET.

SQL\*NET makes remote Oracle databases appear to be a local Oracle database. Data residing on your HP 3000 can be joined with a variety of other databases on your network using Oracle's distributed database technology.

Oracle provides an SQL-based read/write gateway to IMAGE/SQL, called the Oracle Transparent Gateway for IMAGE/SQL. It works with SQL\*NET to offer SQL-based access to IMAGE/SQL on your HP 3000 from remote systems where Oracle is installed. This allows you to transparently read and write to an IMAGE/SQL database from more than 85 platforms where Oracle is supported. Users can view IMAGE/SQL and Oracle data on the network as a single distributed database without needing to know where the data or server resides.

#### **Connecting with Sybase Databases**



Sybase is another widely used relational database. Open Client/Open Server is Sybase's database interoperability solution for the HP 3000 and enables HP 3000 users to read and write data to IMAGE/SQL and ALLBASE/SQL from the Sybase environment or vice versa via APIs. Later this year, Sybase will release a SQL-based transparent gateway to access IMAGE/SQL data.

### Middleware

In the final re-architecting phase, you may decide to re-architect applications that reside on your HP 3000. In this situation, you will be looking for tools that allows for data interoperability not only between NT and the HP 3000 but also between a variety of additional platforms. This situation requires going beyond relying on ODBC or a single database vendor's gateway, but instead to using more generalized tools called middleware. Middleware is the glue that pulls heterogeneous data together in a seamless fashion and does not burden the user with knowing the access methods for each platform in order to do their job.

Figure 9 gives you a picture of a typical role that middleware plays in your enterprise. In this figure, middleware is the glue that allows your environment to communicate seamlessly between your front end, the application server, and the database server which maybe an HP 3000, HP 9000, or a Net Server, as well as a variety of other vendors' platforms. Examples of middleware products that provide this capability are DCE, DISC's OMNIDEX for the Web, Minisoft's Middleman, and Speedware's Autobahn, and StarVision's Client Server Software Development Tools.

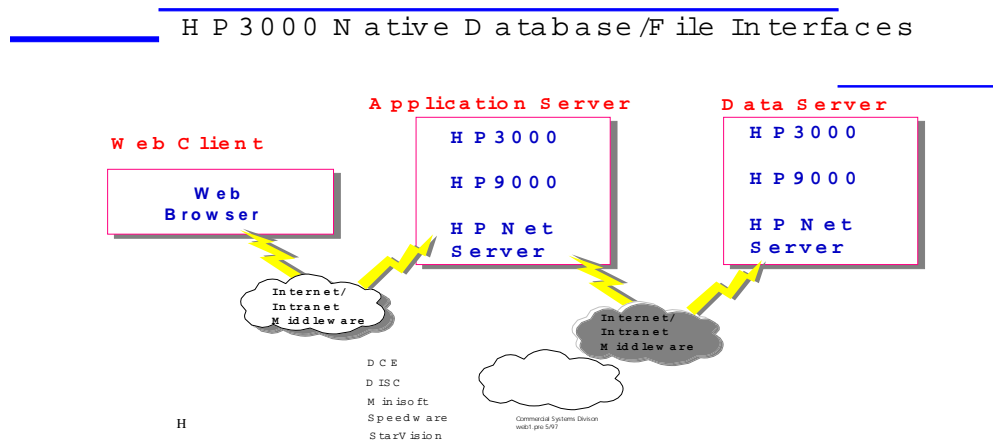


Figure 9

In summary, the mixed HP 3000 and Windows NT environment is being used by customers today, and this environment is becoming more and more common. This paper has outlined various methods that are available to interoperate your HP 3000 with NT. During the past several years, HP has invested in many enhancements to provide interoperability between MPE and UNIX. Now we are adding products and enhancements for interoperability between MPE and NT. We will continue to monitor activities in these rapidly changing environment and update you on new information from time to time.

### Appendix - Additional Technical Information on Samba

Figure 10 shows how a PC client gains access to server shares. The NET LOGON command teaches the PC which username and password to include in connection requests to servers. Samba for MPE uses an MPE style user.account as well as a password pair using the syntax known from ftp i.e. userpass,acctpass. This user identification is only passed to the server for validation purposes, there is no remote session established like with NS/VT.

The NET VIEW command retrieves information about available shares from the server and displays it to the user. You might notice the shares named WEBDOC and TMP SHARE that were shown in the config example in Figure 2. You will also notice share types of Disk and Printer; we'll cover the latter

a little bit later. The NET USE command connects to a server share and maps it to a disk drive letter. Here we map two shares and use drive letters X and Y.

```

C:\>net logon lars.appel
Type your password:
You were logged on, but have not been validated by a server.
Therefore, you may not have permission to use some network resources.

C:\>net view \\telesto
Shared resources at \\telesto
Samba 1.9.16p9

Sharename      Type            Used as  Comment
-----
docs           Disk            Samba doc files from /usr/local/samba/docs
html          Disk            Samba doc files from /usr/local/samba/html
LP            Print           HP 3000 System LP
tmpshare       Disk            serves /tmp (writable) for guest
webdoc        Disk            Easy access to /WEB/DOC for AUTHOR.WEB

The command completed successfully.

C:\>net use x: \\telesto\tmpshare
The command completed successfully.

C:\>net use y: \\telesto\docs
The command completed successfully.

C:\>

```

Figure 10 - PC Client, DOS Prompt

Figure 11 shows similar commands executed on the server side, here using the POSIX Shell. You can see the directory mydir and files like myfile.txt in the output of the ls command on /tmp, which is the MPE directory associated with TMP SHARE according to the configuration shown earlier. You will notice that most of these files belong to the owner LARS.APPEL which is the username specified at the NET LOGON command on the PC. The server process uses calls to setuid() which is AIFCHANGELOGON to assume the appropriate user identity before creating or accessing files (if the client was able to pass a valid user and password specification).

```

*****
MPE/iX Shell and Utilities (A.50.02)
COPYRIGHT (c) Hewlett-Packard Company 1992, All Rights Reserved.
*****

shell/iX> ls -l /tmp
total 2070
drwxr-xr-x  2 LARS.APPEL    0      224 Apr 27 14:12 mydir
-rwxr--r--  1 LARS.APPEL    0      183 Apr 27 14:21 myfile.txt
-rwxr--r--  1 LARS.APPEL    0      640 Apr 27 14:14 myfile.uri
-rwxr--r--  1 LARS.APPEL    0       63 Apr 27 14:17 ugly_24_.txt
-rw-rw-rw-  1 MGR.SAMBA     0    1056866 Apr 28 04:25 v07f.tar.Z

shell/iX> cat -v /tmp/myfile.txt
This file has been created with Windows Write on the PC.^M
It has been save to the X: network drive i.e. the 3000 share.^M
Notice that the PC creates text files with CR LF delimiters.^M

shell/iX> cat -v /tmp/ugly_24_.txt
Some filenames on the PC have to be mapped to legal MPE names^M

shell/iX>

```

Figure 11 - Server Side, POSIX Shell

Looking at the file owned by MGR.SAMBA you will notice that it is possible to have multiple users share a server directory and still have some kind of isolation between them due to the file system access rights for varying file ownership. You may, of course, also configure separate server spaces for different users.

If you look at the output of myfile.txt using the cat -v command, you will notice that every line shows a trailing Ctrl-M character, i.e., CR control code. This is because PC's use CR LF as line delimiter whereas UNIX as well as POSIX systems use LF only. This is not specific to MPE but has to be kept in mind.

You may also notice that some PC filenames are not legal on MPE because they contain invalid special characters like ugly\$.txt does. As of version 0.7 of Samba/iX this issue is being addressed with some MPE specific file name mapping. Notice the filename ugly\_24\_.txt on the MPE side. The 24 is basically the ASCII code of \$ character in hex.

Looking at the file v07f.tar.Z on the server side you can also see another filename transformation, this time not MPE specific. The so-called "name mangling" converts UNIX filenames that would not be legal on the PC side (at least with certain DOS/Windows versions) to some unique but sometimes strange-looking 8.3 style name (8 char filename, 3 char extension) which can be examined on the previous slide. This "name mangling" is not needed with Windows 95 or Windows NT clients as these are able to handle long filenames.

Figure 12 gives you an idea how the server is actually implemented. The listener process PIN 74 has two children created here. This is because two clients are connected at this moment. Each client may be connected to one or more shares with all the connections handled by the respective server process for the very client. The smbstatus output for example shows that the user LARS.APPEL from the client machine pc1lars has mapped two shares (tmpshare and docs) and both are handled by the server process id (pid) 917551.

```

Reflection 1 - SETTINGS.R1W
File Edit Terminal Connection Options Window Help

:showproc job=#j4
QPRI  CPUTIME  STATE  JOBNUM  PIN  (PROGRAM) STEP
D202  0:00.113  WAIT   J4      75   :XEQ smbd.prv.samba -D
D202  0:00.423  WAIT   J4      74   [SMBD.PRIV.SAMBA] -D
D202  0:00.530  WAIT   J4      47   [SMBD.PRIV.SAMBA]
D202  0:00.511  WAIT   J4      24   [SMBD.PRIV.SAMBA]

:xseq /usr/local/samba/bin/smbstatus

Samba version 1.9.16p9
Service      uid          gid          pid          machine
-----
tmpshare     LARS.APPEL  APPEL       917551       pc1lars [15.xxx.xxx.xxx] Tue Apr 29 1
webdoc       AUTHOR.WEB  WEB         393240       theresa [15.xxx.x.xx] Tue Apr 29 15:4
docs         LARS.APPEL  APPEL       917551       pc1lars [15.xxx.xxx.xxx] Tue Apr 29 1

Locked files:
Pid  DenyMode  R/W  Name
-----
917551  DENY_READ  RDWR  ./myfile.txt  Tue Apr 29 15:47:17 1997

:
f1 f2 f3 f4 f5 f6 f7 f8
24.2 HP70094 -- telesto (vt-mgr) ? Enter Insert Num Caps Stop

```

Figure 12 - Server Status and Processes

At the bottom you can also see that the Samba server has to handle PC shared access modes to files, which are very different from what is usual on UNIX or MPE. Here the file myfile.txt is opened for read/write access with so-called DenyMode being Deny\_Read. For the PC clients this is a kind of exclusive access but the server side may handle this still somewhat differently when e.g. trying to print the file myfile.txt from the POSIX shell or CI.

HP 3000 playing the "client role" using Samba/iX

The client functionality is mainly provided by the SMBCLIENT utility. It provides access to SMB servers e.g. an NT server or a Windows for Workgroups PC with an ftp-like user interface. This is more like file transfer, it does not allow mounting of remote directories.

On Figure 13, you can see the -L option of SMBCLIENT which works like the NET VIEW command on the PC i.e. shows the available shares for a server (I'm accessing my own machine TELESTO here). The next command connects to the TMP SHARE on the server TELESTO, specifying username and passwords and performs a DIR command which is included in the command line with the -c option of SMBCLIENT. Notice the double backslashes in the share specification which are needed due to the fact that the backslash character has a special meaning to the shell.

The screenshot shows a terminal window titled "Reflection 1 - SETTINGS.R1W". The menu bar includes File, Edit, Terminal, Connection, Options, Window, and Help. The terminal output shows the following commands and results:

```
shell/iX> smbclient -L telesto
Server=[telesto] User=[mgr.samba] Workgroup=[WORKGROUP] Domain=[WORKGROUP]

Sharename      Type      Comment
-----
docs            Disk      Samba doc files from /usr/local/samba/docs
html            Disk      Samba doc files from /usr/local/samba/html
IPC$            IPC       IPC Service (Samba 1.9.16p9)
LP              Printer   HP 3000 System LP
tmpshare        Disk      serves /tmp (writable) for guest
webdoc          Disk      Easy access to /WEB/DOC for AUTHOR.WEB

shell/iX> smbclient \\\telesto\\tmpshare userpw,acctpw -U lars.appel -c dir

mydir           D         0      Sun Apr 27 16:12:43 1997
myfile.txt      A         183   Sun Apr 27 16:21:31 1997
myfile.wri      A         640   Sun Apr 27 16:14:54 1997
ugly$.txt       A          63   Sun Apr 27 16:17:16 1997
v07f.tar.Z      A        1056866 Mon Apr 28 06:25:27 1997

0 blocks of size 0. 0 blocks available

shell/iX>
```

At the bottom of the terminal window, there are function key buttons (f1 through f8) and a status bar showing "24, 11" and "HP70094 -- telesto (vt-mgr)".

Figure 13 - Samba Client for MPE/iX

Figure 14 shows the online help screen of SMBCLIENT which looks pretty much like ftp commands (get, put, cd, pwd, dir etc.) except for the printing related commands, which are used for printer share access. The remaining commands on this slide shows how to get a file from a remote system and examine the contents of the local copy then.

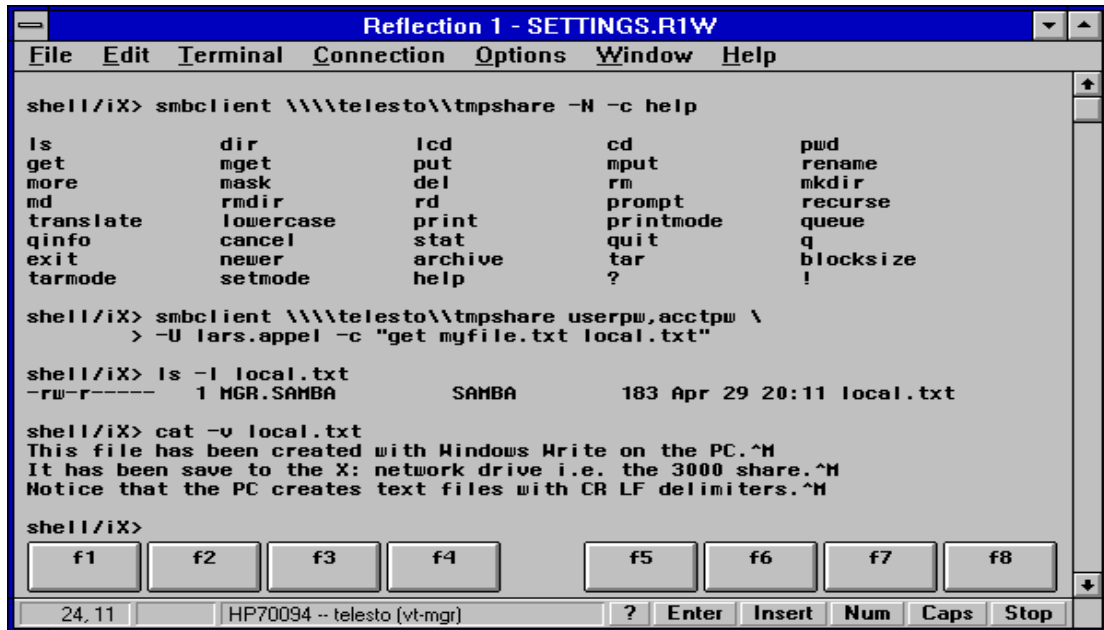


Figure 14 - Samba Client for MPE/iX

Notice that all SMBCLIENT examples used the -c option to specify the command on the command line. The SMBCLIENT program also has an interactive mode which looks much more like FTP i.e. you connect to a server and then get a prompt to execute a sequence of command, terminated by an exit or quit command. Due to a limitation of the select() function on MPE/iX, the interactive mode does not yet work properly. It can be worked around by using the -c option of SMBCLIENT at present.

The SMBCLIENT utility also provides an option -M that allows to send a WinPopup message to the remote computer (if that computer supports this kind of messages and has receiving enabled). This does not require a PC to be logged on to the 3000 or have a network drive mapped.

#### Current Limitations

The main thing to keep in mind in this area is the fact that Samba is basically a UNIX implementation of a PC networking protocol and that these two platforms are not very "MPE aware". PC and UNIX file systems only know about bytestream files i.e. they consider files as just a bunch of bytes and have no concept of file types like text/binary or record-oriented access like FREADBYKEY or FUPDATE for KSAM files.

You can find various details about current or even general limitations of the Samba implementation for MPE/iX in the ReadMe file that is on the web and updated with every new version (sometimes even in between).

In many cases it might be possible to create MPE specific extensions of the original Samba code to provide better MPE integration. But sometimes this might also fail due to the fact that the PC side lack appropriate tools or concepts (and that Microsoft software cannot be changed quite easily to better work with MPE/iX).