VeriTest®

April 2003
www.veritest.com • info@veritest.com

Microsoft: Windows Server 2003 Study

Test report prepared under contract from Microsoft Corporation

Executive Summary

Microsoft Corporation commissioned VeriTest to conduct a reliability study comparing Windows Standard Server 2003 to various versions of Red Hat Linux. VeriTest found Microsoft Windows Server 2003 and Red Hat Linux to have equal reliability, or uptime, over a period of 61 days.

Key findings

- ❑ VeriTest found Microsoft Windows Server 2003 and Red Hat Linux to have equal reliability, or uptime, over a period of 61 days.
- ❑ In addition to a reliability comparison, VeriTest conducted two pilot studies comparing Red Hat Linux to Windows Server 2003: an "ease of administration study" and a "speed of recovery" study. Preliminary results from these studies indicate that Windows 2003 is significantly faster to set up and returns to a stable state in roughly 30% less time than Red Hat Linux.
- ❑ Follow-up testing is planned to explore preliminary results to better quantify the ease of management and speed of recovery of Windows Server 2003.

In addition to the reliability comparison, VeriTest also conducted two pilot studies that measured the differences between Windows Server 2003 and Red Hat Linux in ease of administration and speed of recoverability from an unexpected power failure. VeriTest measured the amount of time required to perform similar tasks on the two operating systems. While the two pilot studies need more data and further exploration, preliminary results showed that Windows Server 2003 proved superior to Red Hat Linux in timing measurements of ease of administration and speed of recoverability. To follow up on pilot results, more testing is scheduled to take place in the following months.

Based on the preliminary results from the pilot studies, compared to Red Hat Linux 8.0, Windows Server 2003 needed fewer minutes of IT administrator time to setup and configure, and was on average 30% faster to boot up after an unexpected power loss. Additionally, VeriTest found that Server 2003 had equal reliability to that of Red Hat Linux 7.2. These findings suggest that Windows Server 2003 offers strong value to small and medium businesses that need a reliable, easy-to-manage server that can recover quickly and completely after a crash.

Table of Contents

Reliability Study

Study Methodology

For the reliability study, VeriTest compared Windows Standard Server 2003 (RC1, build 3663) on three Dell PowerEdge 1500SC servers to Red Hat 7.2 Server (at the time of the study, Red Hat 7.2 was the latest version of the operating system) on three identical servers. (See appendix A for complete details on the server hardware that was used.) Each operating system ran on three servers for a total of three Windows 2003 test servers, and three Red Hat Linux test servers. All servers were "exercised" against clients running test scripts for a testing period of 61days of real time and monitored for problems that could cause them to become unreliable. The client test scripts that were used emulated file, print and email functions against the test servers. The testing period ran for 61 days, or a theoretical total of 183 days, or 61 days per server uptime times 3 servers.

To realistically emulate the typical IT environment found in small or medium businesses, we networked 20 clients to each of the six servers, for a total of 120 clients. The clients that ran the test scripts against the servers were Dell Optiplex 100s with 500MHz Celerons with 128MB RAM running Windows XP Professional SP1. For each client script, we used Rational VisualTest 6.5 to create, design, and automate the email, file, and print scripts used against both Windows Standard Server 2003 and Red Hat Linux 7.2. We designed the client scripts for the server operating systems being compared to perform the same tasks in order to ensure equality of testing between the two systems. (See Appendices C, D, and E for the client scripts used in the reliability test.)

The test clients attached to each individual server were assigned either an email, file or print script to run continuously for 61 days against the server as described in Figure 1 below.

Windows Server and Client Configuration	Linux Server and Client Configuration
Server #1 – Windows Standard Server 2003 • 7 email activity clients • 7 file activity clients • 6 print clients	Server #1 - Red Hat Linux 7.2 • 7 email activity clients • 7 file activity clients • 6 print clients
Server #2 - Windows Standard Server 2003 • 10 file activity clients • 10 print clients	Server #2 - Red Hat Linux 7.2 • 10 file activity clients • 10 print clients
Server #3 - Windows Standard Server 2003 • 10 file activity clients • 10 print clients	Server #3 - Red Hat Linux 7.2 • 10 file activity clients • 10 print clients

Figure 1: Windows and Linux Server and Client Configurations

The test servers were configured to provide email, file and print services to the attached clients. To ensure that printer hardware issues would not affect server reliability, print servers on all six servers (both Windows and Linux) were configured to print to a NULL printer. Email and file capabilities were set up as described in the following lists below.

The following steps were performed to set up the test Red Hat Linux 7.2 systems for email and file capabilities:

1. Assign the server name to Server*x*, when *x* is 1, 2 or 3.
2. Set domain name to linux*x*.com, when *x* is 1, 2 or 3.
3. Add to /etc/hosts "200.110.1.250 server*x* server*x*.linux*x*.com, when *x* is 1, 2 or 3"
4. Verify xinetd is working by performing a tail on the file /var/log/messages
5. Install POP server called GNO RPM on Dell CD

6. Drag the file /mnt/cdrom/redhat/rpms/imap-2000c-15.i386.ram to the RPM manager window
7. Install the imap package.
8. Ensure that “verify disable = no” for /etc/xinetd.d/ipop3
9. Comment out the line beginning with “daemon port options…” in the file /etc/sendmail.cf.
10. Create users 20 users using the user management user interface.
11. Enable SAMBA by selecting the SMB process in the process manager user interface.
12. Configure the /etc/samba/smb.conf file as listed in Appendix D.
13. Create a new printer called “Epson” using the Print manager user interface.
14. Change the print.conf file to print to /dev/null.

The following steps were necessary to set up the Windows Standard Server 2003 systems for email and file capabilities:

1. Go to “Manage Your Server” user interface.
2. Add Active Directory and following directions in Wizard.
3. Add File Server and following directions in Wizard.
4. Add Print Server and following directions in Wizard.
5. Add POP Server and following directions in Wizard.
6. Go to “Start Menu -> Settings -> Printers” and add a NULL printer
7. Create users and mailboxes in the “Manage Your Server” user interface.
8. Create user folders in the user share folders.
9. Set the unique static IP address for each server.

Figure 2 shows the complete test bed configuration that was used.

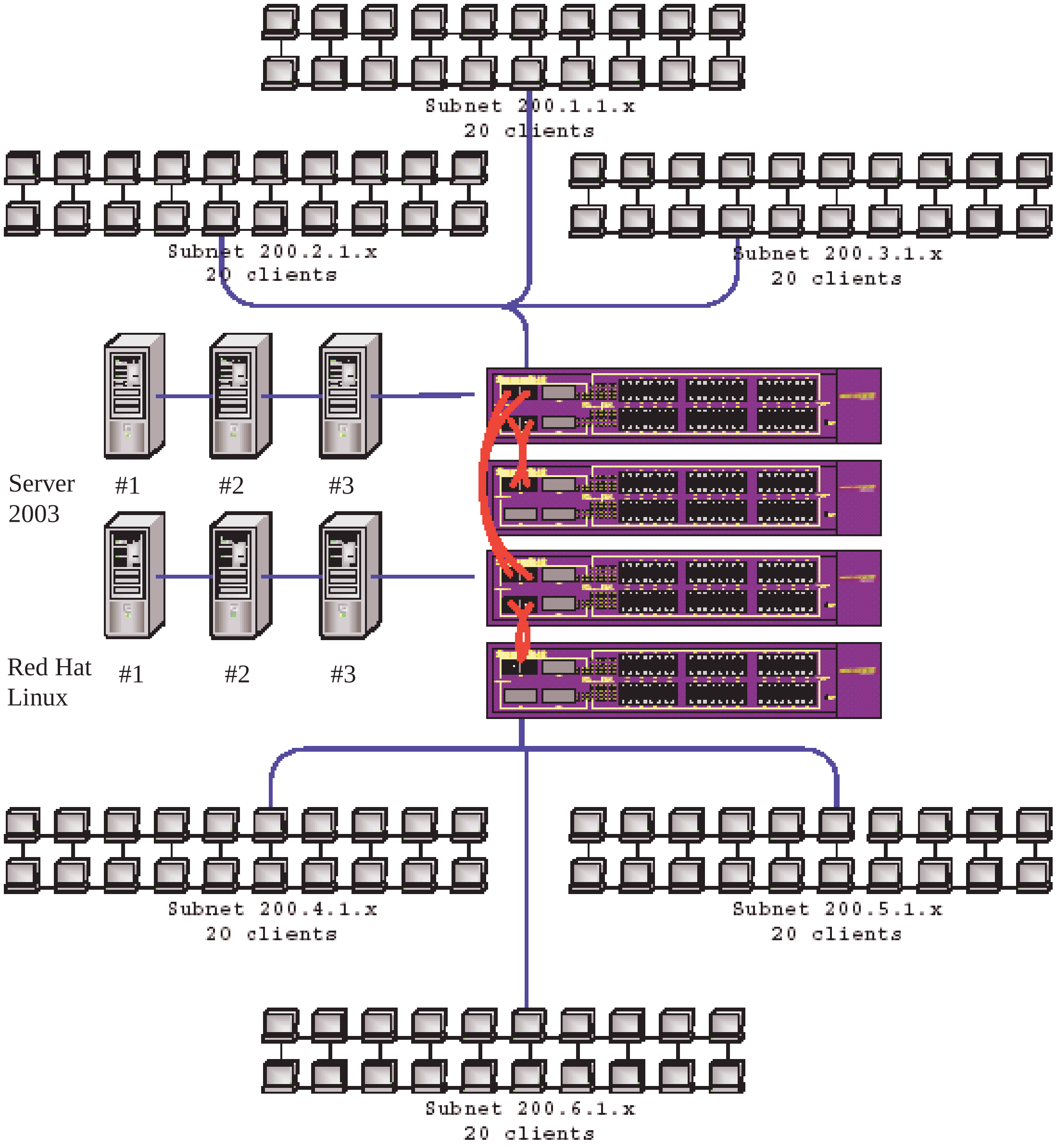


Figure 2: Reliability test bed configuration

Since the client scripts were designed to mimic typical user workloads on the servers, we introduced a random amount of sleep time from one to ten minutes between each iteration of the script. For the duration of the 61-day test, clients were checked twice a day during business hours to assure that the script programs were running correctly. A visual check was performed on all 120 clients and hung clients were rebooted.

To prevent short-term power supply fluctuations from adversely affecting the test, each server was connected to an independent uninterruptible power supply (UPS). The UPS used was a diesel-powered generator that could deliver no less than eight hours of continuous power.

Study Results

Over the test period of 61 days, we repeatedly ran the email, file and print client scripts against the Windows and Linux test servers. For each server workload, we charted the number of times each client script was run against the server in the tables below.

To compare the reliability of Windows to Linux servers in the email server workload, we designed the email activity clients to run a script that created and sent three emails to all of the other mailboxes on the server under test. One email message was a small text message, one email message had a medium (32KB) attachment, and one email message had a large (1MB) email attachment. After each client sent all of its email, it performed a “send/receive” action and copied all of the emails in its server mailbox to the local system. It then deleted all email messages. (See Appendix B for a complete listing of the Ration VisualTest 6.5 email activity script.)

As shown in Figure 3, we performed 114,980 email server tasks on the Linux server and 115,032 on the Windows Server. After 61 days of continual uptime, both Windows and Linux email servers had perfect uptime.

	Num of small files/messages	Num of medium files/messages	Num of large files/messages	Num of Send/Receive	Total count
Linux Email Server #1	28,759	28,747	28,740	28,734	114,980
Windows Email Server #1	28,782	28,768	28,752	28,730	115,032

Figure 3: Linux email server activity

To compare the reliability of Windows to Linux servers in the file sharing workload, we designed the file activity clients to run a script that copied three files (a small 32KB file, a medium 1MB file, and a large 10MB file) from the file server to the local client. The script then moved the files back to the server in a different folder location. The script finally deleted the new copy from the server. (See Appendix D for a complete listing of the Ration VisualTest 6.5 file activity script.)

As shown in Figure 4, we performed 405,479 file server tasks across the three Linux file servers, and 420,578 across the three Windows file servers.

	Num of small files/messages	Num of medium files/messages	Num of large files/messages	Total count
Linux File Server #1	34,970	34,961	34,953	104,884
Linux File Server #2	49,515	49,493	49,480	148,488
Linux File Server #3	50,711	50,702	50,694	152,107
Linux Total	**135,196**	**135,156**	**135,127**	**405,479**
Windows File Server #1	35,568	35,565	35,554	106,687
Windows File Server #2	50,277	50,267	50,257	150,801
Windows File Server #3	51,039	51,029	51,022	153,090
Windows Total	**136,884**	**136,861**	**136,833**	**410,578**

Figure 4: Linux file server activity

Finally, the print activity clients ran a script against the test servers that submitted three text files (a small 1KB file, a medium 24KB file, and a large 67KB file) for printing on the print server. (See Appendix E for a complete listing of the Ration VisualTest 6.5 print activity script.)

As shown in Figure 5, we performed 396,304 print server tasks across the three Linux servers and 356,931 across the three Windows print servers.

	Num of small files/messages	Num of medium files/messages	Num of large files/messages	Total count
Linux Print Server #1	30,811	30,803	30,799	92,413
Linux Print Server #2	49,466	49,454	49,441	148,361
Linux Print Server #3	51,849	51,843	51,838	155,530
Total	**132,126**	**132,100**	**132,078**	**396,304**
Windows Print Server #1	28,057	28,034	28,002	84,093
Windows Print Server #2	45,963	45,921	45,849	137,733
Windows Print Server #3	45,092	45,038	44,975	135,105
Total	**119,112**	**118,993**	**118,826**	**356,931**

Figure 5: Linux print server activity

We found no reliability problems on any of the six servers in our reliability test bed. Based on these findings, we found Windows Server 2003 (RC1, build 3663) to be equally reliable as Red Hat Linux 7.2.

Ease of Administration Study

After the reliability study was completed, Microsoft asked us to create an independent test that would measure ease of administration of both Windows and Linux servers. The ease of administration, in this study, was defined as *the amount of time needed by an IT administrator to configure a single server for email, file, and print services attached to a single client.* This test was called the "Ease of Administration" study and like the reliability study, was intended to emulate a typical single-server environment found in many small and medium businesses. For the hardware to conduct the test, Microsoft supplied six identical Dell PowerEdge 1500SC server class systems. (See Appendix A for more details on the server.)

To run the Ease of Administration study, we selected two skilled IT administrators to create and configure a Windows Standard Server 2003 RC2 and a Red Hat Linux 8.0 server as an email server, a file server, and as a print server. One of the IT administrators had a strong Windows server software administration background and a modest Linux server software administration background. The other IT administrator had a strong Linux server software administration background and a modest Windows server software administration background.

To conduct the study, both IT administrators were given the instruction to configure and set up both Windows and Linux servers in a single-server environment, so no domain controllers or additional servers were assumed to be present in the test bed environment. Both IT administers were instructed to perform clean installations, and to setup the email/file/print server configurations for both Windows Standard Server 2003 (Release Candidate 2) and Red Hat Linux 8.0 Server test servers. As each IT administrator was performing the server setup tasks, we timed the number of minutes for each task, and each IT admin recorded his installation experience in a journal. (See Appendix G for complete journals).

Two of the three servers for both Windows and Linux were configured as single workload file and print servers. To set up the single workload file and print servers for the Windows Standard Server 2003 servers, both IT administrators used the "Manage Your Server" interface to enable and configure file and print services. For the Red Hat Linux 7.2 servers, both IT administrators installed and configured SAMBA (http://www.samba.org) to manage both file and print serving. The third server in each operating system group was configured to carry all three server workloads of email, file, and print. For the third multi-workload

Windows server, both IT administrators again used the “Manage Your Server” interface to enable and configure email services. For the Red Hat Linux 7.2 server, both IT administrators installed and configured the imap-2000c-15.i386.ram package to set up the server to handle email, file and print services.

As shown in Figure 6, both of the IT administrators were able to setup, install, and configure Windows 2003 Server in significantly less time than Red Hat Linux 8.0. On average, the IT administrators were able to complete the Windows 2003 Server installations in an average of 79.5 minutes, while it took them on average 530.9 minutes to perform the same tasks using Red Hat Linux 8.0.

	IT Administrator 1 On Windows 2003 Server	IT Administrator 2 On Windows 2003 Server	IT Administrator 1 On Red Hat 8.0	IT Administrator 2 On Red Hat 8.0
Total time in minutes	**44.6**	**114.3**	**517.2**	**544.6**

Figure 6: Overall system setup and configuration times (minutes)

Due to the small sample size, more testing is planned to conduct the study with a larger sample size of IT administrators and more control points. Our preliminary results in this study suggest that faster setup time of Windows Server 2003 was due to an administration interface, setup wizards, and integrated software platform that was easier to use. While testing is needed to compare the ease of use of Windows Server to Linux, preliminary results suggest that Windows Server requires significantly fewer minutes of administrator time than does Red Hat Linux.

Recoverability Study

The third study that we conducted proved another strong capability of Windows Server 2003: its ability to recover quickly after an unexpected power loss. To compare the speed of recoverability of Windows to Linux, we set up a test environment of a single server (either Windows or Linux) attached to a client running Windows XP. We ran the same email, file and print scripts on the server/client configuration for both operating systems. To measure the speed of recovery after a power loss, we cut off the power three times for each operating system and measured the time required to restore the server to its pre-crash state.

Figure 7 displays the average time in seconds for both Windows and Linux servers to recover. On average, Windows Server 2003 recovered about 30% faster than Red Hat Linux.

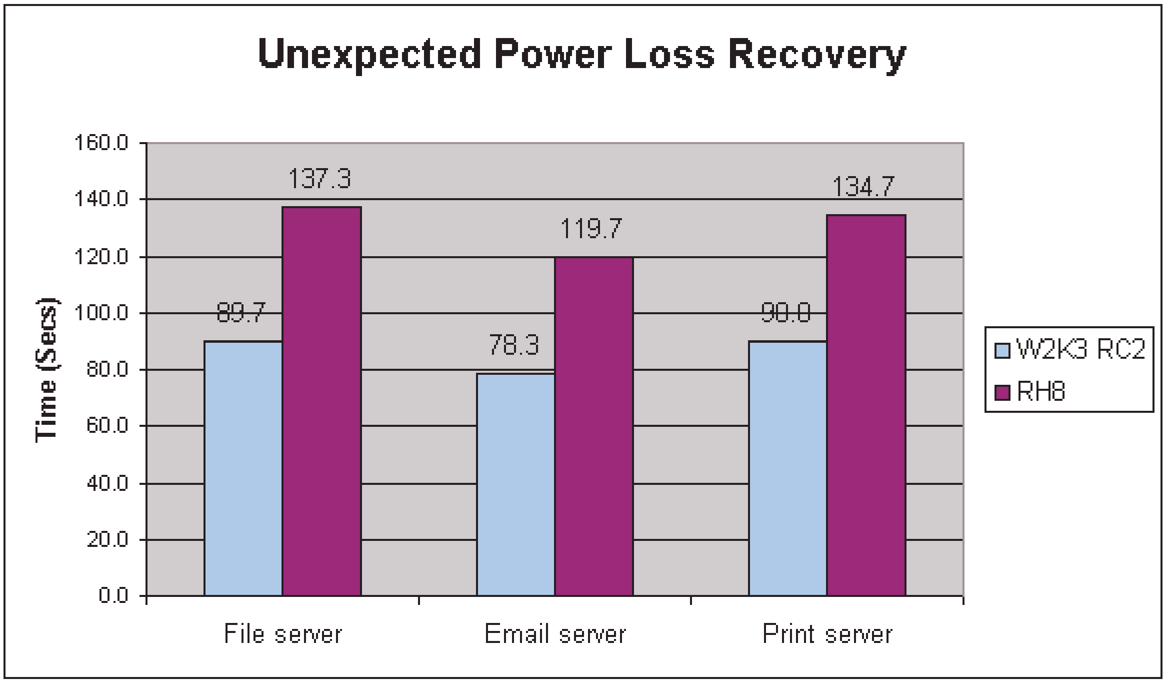


Figure 7: Average recovery times in seconds. (Smaller is better.)

The preliminary Speed of Recoverability test results suggest that Windows Server 2003 recovers more quickly from an unexpected power loss and returns to its pre-crash state in less time that Red Hat Linux 8.0. Follow-up testing is planned to explore preliminary results and to better quantify the speed of recoverability of Windows Server 2003.

Appendices

Appendix A. Reliability Study Dell Server Specifications

Complete specifications can be found at:
http://www.dell.com/us/en/esg/topics/esg_pedge_sc_towermain_servers_pedge_1500sc.htm

PowerEdge 1500SC - Up to 2 Intel® Pentium® III processors at 1.13GHz, 1.26GHz, and 1.4GHz

Front Side Bus - 133MHz front side bus allows for faster data throughput compared to legacy 100MHz front side bus speeds

Cache - 32KB Level 1 cache (16KB instruction cache and 16KB two-way write-back data cache); 512KB Level 2 cache (full processor speed)

Chipset - ServerWorks High End-SL (HE-SL) chipset utilizes memory interleaving/setcombining and a tri-peer
PCI bus architecture for uncompromising I/O throughput

Memory- 128MB - 4GB 133MHz ECC SDRAM; Features Two-Way Memory Interleaving (requires DIMMs to be added in pairs of equal capacity) 4 DIMM sockets on system board can be set combined

Expansion Slots - 4 x 64-bit/66MHz PCI (supports 3.3V cards only); 2 x 32-bit/33MHz PCI (support 5V or Universal cards)

Drive Controller - Embedded Adaptec® AIC-7899 dual channel Ultra3 (Ultra160)/LVD SCSI for one of the fastest high-performance SCSI technologies available

RAID Controller - Optional embedded PowerEdge Expandable RAID Controller, Version 3, Single Channel (PERC3/SC)

Drive Bays - Hot-pluggable hard drive bays: six 1.0" hard drive bays

Peripheral bays - Three 5.25" half-height bays **-** one occupied by combination CD-ROM/floppy, one available for tape, other media; Up to 48X max. IDE CD-ROM standard; 3.5" 1.44MB diskette drive standard

Hard Drives - 18GB1, 36GB and 73GB (1") 10,000 RPM Hot Plug Ultra3 (Ultra160) SCSI hard drives ;18GB and 36GB (1") 15,000 RPM Hot Plug Ultra3 (Ultra160) SCSI hard drives

Maximum Internal Storage - 438GB Maximum SCSI HDD Storage Capacity (6 x 73GB 10K RPM SCSI)

Tape Backup Options - Optional internal PowerVault 100T DDS4 tape drive (20GB/40GB); Optional internal PowerVault 110T VS80 Tape Drive (40GB/80GB)

Communications - Embedded Intel PRO/1000 XT Network Connection (auto-switching to 10/100/1000 Mbps); For Additional Availability and Fail Over NIC support: Intel PRO/100+ Dual-Port Server Adapter; Intel PRO/100S Server Adapter (with IP SEC Encryption); Broadcom® NetXtreme 10/100/1000 (Cat-5 Copper Cabling); Intel PRO/1000 XT56 K2 modem (internal)

Input Devices - Standard Windows keyboard; Deluxe Windows keyboard; Dell PS/2-style mouse

USB Ports - 2 USB ports

Power - Standard 350 Watt Non-Redundant supply; Optional 2 x 350W Redundant Supplies for availability to keep the server up and running; Auto-switching 110/220 Volts

Chassis - Tower Chassis Only (with feet): 17.5" H x 10.25" W x 23.5" D (add .25" to the depth for models with redundant power supplies to account for the locking mechanism); Weight 66 lb. (fully loaded)

Graphics - Integrated ATI-Rage XL controller w/ 4MB of SDRAM (not upgradeable)

Management

- Fault monitoring of voltage, fan, and thermal conditions to help ensure notification in case of
- potential problems
- Tracks memory errors that have been corrected by the ECC memory
- Automatic Server Recovery will reboot and restart the server if the OS hangs without user intervention
- Email or paging through Dell OpenManageTM keeps administrators informed of potential server
- problems before they become critical
- Asset management features enable customers to inventory server configuration, CPU, memory and disk information, helping keep track of systems and keep them up-to-date
- User definable OS thresholds can be set, allowing administrators to tune systems and help reduce bottlenecks to performance
- Dell OpenManage Server Setup CD included with every server to get your PowerEdge up, running, and contributing to your infrastructure quickly

Software - Optional Software

- Novell® NetWare® Version 5.1 (validated and supported, no factory install)
- Microsoft® Windows NT® Server, Version 4.0
- Microsoft BackOffice Small Business Server 2000
- Microsoft Windows® 2000 Server
- Red Hat® Linux 7.2 and 7.3
- Dell Tape Backup Software by Veritas® Backup

Appendix B: Reliability Study Steps to Configuring Clients

Email activity clients

1) We copied an Outlook Express shortcut from the Start menu to the desktop.
2) We defined the email username as "user(x)" ["x" is a variable to be replaced].
3) If connecting to Windows Standard Server 2003 email servers, we used the email address form: "user(x)@veritestwin(x).com ["x" is unique for each client].
4) If connecting to Linux email servers, we used the email address form: "user(x)@server(x).linux(x).com ["x" is unique for each client].
5) We assigned the incoming and outgoing mail server to be the IP address of attached server.
6) If connecting to Windows Standard Server 2003 email servers, we defined the password to be "user(x) ["x" is a variable to be replaced].
7) If connecting to Linux email servers, we defined the password to be "user(x)user(x)" ["x" is a variable to be replaced].
8) If connecting to Windows Standard Server 2003 email servers, we checked the "SPA" box
9) If connecting to Linux email servers, we did not check the "SPA" box
10) We made sure that "remember password" check box was checked.
11) When done with the Wizard setup, we reopened Outlook Express and navigated to: Tools->Options.
    a) Under the "Send" tab, we deselected the "Save copy of sent messages in the 'Sent Items' folder."
    b) Under the "Receipts" tab, we selected the "Request a read receipt for all sent messages" check box.
    c) Under the "Receipts" tab, we selected the "Always send a read receipt" radio button.

File activity clients

1) We created a file called "clientname.txt" in the root folder on the "C:" drive and inserted the specific user ID ("userxx") into this file.
2) We mapped a network drive to a share on the attached server and copied a shortcut to the desktop.
3) We copied the file serving script to client desktop.
4) Using Ration VisualTest, we opened and compiled the script.
5) We made sure that the "My Documents" folder was not maximized when it was opened.

Print activity clients

1) We created a file called "clientname.txt" in the root folder on the "C:" drive and inserted the specific user ID ("userxx") into this file.
2) We copied the files large.txt, medium.txt, and small.txt to the "My Documents" folder on client desktop from the attached server share \\veritest-winx\share.
3) We copied the print script from the share on attached server to the desktop.
4) Using Ration VisualTest, we opened and compiled the script.
5) We added the assigned printer on the attached server.

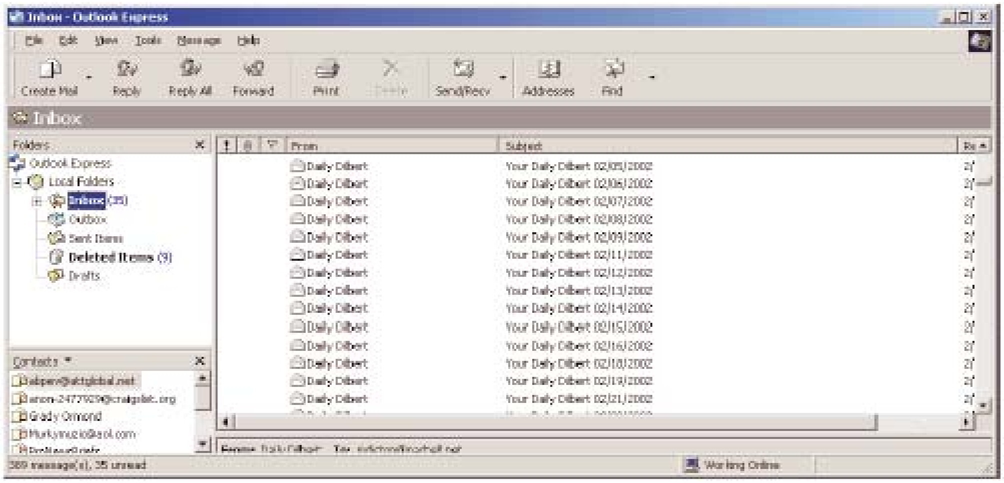
We used the information in C:\clientname.txt file to determine the specific client name. All client activity was logged in a file located at “C:\”. The format we used was “assignedserver_userID_assignedactivity.log”. For example, “Net1_user1_email.log” collected information regarding the email activity on client one that was connected to Windows Standard Server 2003 server number one.

We used the following procedures for checking the email, file, and print clients.

Checking email clients

1) If one of the email clients stopped, we stopped all other email clients also associated with the same server and cleared out all messages from all of the mailboxes. Once we deleted all email, we restarted the email scripts of these clients.
2) With Microsoft Outlook Express open, we navigated to: “Tools -> Send and Receive” and selected “Receive all.” This retrieved all emails from the server.
3) At the Outlook Express main window, we navigated to the “Inbox” in the left pane. We selected all emails and deleted them.
4) To start the script for a particular client, we executed the script application located on the desktop.

Note: Microsoft Outlook Express must be in a specific state for the script to execute correctly. We required that the Outlook Express interface be visible in four panes for the script to operate correctly, as shown in Figure 8. If we did not see this, we forced this state by selecting the “Inbox” folder located in the upper-left pane. We also ensured that the Outlook Express window was not maximized.



From	Subject
Daily Dilbert	Your Daily Dilbert 02/05/2002
Daily Dilbert	Your Daily Dilbert 02/06/2002
Daily Dilbert	Your Daily Dilbert 02/07/2002
Daily Dilbert	Your Daily Dilbert 02/08/2002
Daily Dilbert	Your Daily Dilbert 02/09/2002
Daily Dilbert	Your Daily Dilbert 02/11/2002
Daily Dilbert	Your Daily Dilbert 02/12/2002
Daily Dilbert	Your Daily Dilbert 02/13/2002
Daily Dilbert	Your Daily Dilbert 02/14/2002
Daily Dilbert	Your Daily Dilbert 02/15/2002
Daily Dilbert	Your Daily Dilbert 02/16/2002
Daily Dilbert	Your Daily Dilbert 02/18/2002
Daily Dilbert	Your Daily Dilbert 02/19/2002
Daily Dilbert	Your Daily Dilbert 02/21/2002

Figure 8: Outlook Express main window

Checking file clients

1) If one of the file clients stopped, we navigated to the \\Network share to associated server\user folder (i.e., \\veritestwin1\share\userxx).
2) Once we navigated to the client’s user folder, we deleted any files that had remained from previous runs.
3) We also made sure that files from previous runs were not on the client desktop. These files can be identified by the following names:
   a) 1mbb.exe
   b) 4mbb.exe
   c) 10mbb.exe
   d) 32kBb.exe
   e) 64kBb.exe
   f) 256kBb.exe
4) We then closed all of the windows and executed the file script located on the desktop.

Checking print clients

1) If one of the print clients stopped, we closed all windows and executed the print script located on the desktop.
2) If the “Access denied” error appeared on a client, we navigated to: “Start Menu-> Settings-> Printer and Fax” and deleted the printer.
3) We then navigated to “Start menu-> Run” and typed in “\\Server “ (replace “server”). This opened a window that showed the shared resources of that server.
4) We right-clicked on the printer and selected “Connect.” This added the printer to that client.
5) Before restarting the script, we made sure that all of the windows were closed.

We included sleep times within the script that caused the script to appear to have stopped. Before resetting any script, we made sure that the client was actually stopped and not in a sleep mode. We used the “Ctrl-q” key combination to cancel the script.

Appendix C: Reliability Study Email Activity Script

```
'$INCLUDE 'RECORDER.INC'

SetDefaultWaitTimeout(Timeout)

SUB StopTheTest (vtNotifyData AS VARIANT )
        STOP
End Sub

'Pressing <CNTL>-q halts execution cleanly.
ON KeyPress ("q",FCONTROL) CALL StopTheTest

DIM number_of_users, number_of_actions
number_of_users = 7
number_of_actions = 4

DIM     user(number_of_users) AS STRING
user(0) = "user1"
user(1) = "user2"
user(2) = "user3"
user(3) = "user4"
user(4) = "user5"
user(5) = "user6"
user(6) = "user7"

'DIM    filenames(2) AS STRING
'filenames(0) = "1mbb"
'filenames(1) = "32KBb"

DIM i, j
i = 0
j = 0

DIM rndtime, rndtimestr$

DIM sleep_length
sleep_length = 2.0

DIM Astring$, logname$
'Astring = "1user"

dim hwnd as long

Scenario "email_server"

Open "C:\clientname.txt" FOR INPUT AS #2
LINE INPUT #2, Astring

logname = "C:\NET1_" + Astring + "_email.log"
Open logname FOR APPEND AS #1
PRINT #1, " "
PRINT #1, "Start of Test: " + DATETIME$
PRINT #1, "Server name: NET1"
PRINT #1, "Client name: " + Astring
```

```
'Minimize the Visual Test window.
If GetHandle(GH_HWNDCLIENT) Then WMinWnd(GetHandle(GH_HWNDCLIENT))

' Check the resolution that this script was recorded on.
CheckResolution (800, 600)

    WDesktopClk(129, 202, VK_LBUTTON)
    Play "outlook{ENTER}"
    Sleep(2.0)

    Play "^(m)"
    Sleep(sleep_length)
    Play "^(y)"
    Sleep(sleep_length)
    Play "{RIGHT}{RIGHT}"
    Sleep(sleep_length)
    Sleep(sleep_length)
    Play "{ENTER}"
    Sleep(sleep_length)

RANDOMIZE TIMER

DO WHILE TRUE

    i = 0
    DO WHILE i < number_of_users

        'We don't want to send email to ourselves
        IF user(i) = Astring THEN
            i = i + 1
            IF i >= number_of_users THEN
                i = 0
            END IF
        END IF

        j = 0
        DO WHILE j < number_of_actions

            'Sleep random time between 1 minute and 10 minutes
            rndtime = RND * 10 * 60 'Number of seconds to sleep
            if (rndtime < 60) THEN
                rndtime = rndtime + 60
            END IF
            PRINT #1, "Wait: " + STR$(rndtime) + " secs at: " + DATETIME$

            DO WHILE rndtime > 0
                rndtimestr = "Time to next action: " + STR$(rndtime) + " secs"
                rndtime = rndtime - 1
                StatusBox rndtimestr , , , , , TRUE
                Sleep(1.0)
                StatusBox Close
            LOOP


            IF (j = 0) THEN
                Play "^(n)"
                CurrentWindow = WFndWndC("New Message", "ATH_Note",
FINDWINDOWFLAGS_IF, Timeout)
                WSetWndPosSiz(CurrentWindow, 200, 90, 400, 420)

                Play user(i) + "@veritestwin1.com"
                Play "{TAB}{TAB}text - "
                Play user(i)
                Sleep(sleep_length)

                Play "{TAB}This is a short message. Hello."
                Sleep(sleep_length)
                WToolbarButtonClk("@1", "Send")

            ELSEIF (j = 1) THEN
                Play "^(n)"
                CurrentWindow = WFndWndC("New Message", "ATH_Note",
FINDWINDOWFLAGS_IF, Timeout)
                WSetWndPosSiz(CurrentWindow, 200, 90, 400, 420)

                Play user(i) + "@veritestwin1.com"
                Play "{TAB}{TAB}small attach - "
                Play user(i)
                Play "{TAB}"
                Sleep(sleep_length)

                WToolbarButtonClk("@2", "&Insert")
                Sleep(sleep_length)
                WMenuSelect("File &Attachment...")
                Sleep(sleep_length)

                Play "32kb_attach.exe"
                Sleep(sleep_length)
                Play "{ENTER}"
                Sleep(sleep_length)

                CurrentWindow = WFndWndC("small attach", "ATH_Note",
FINDWINDOWFLAGS, Timeout)
                WSetWndPosSiz(CurrentWindow, 115, 78, 508, 420)
                WToolbarButtonClk("@1", "Send")

            ELSEIF (j = 2) THEN
                Play "^(n)"
                CurrentWindow = WFndWndC("New Message", "ATH_Note",
FINDWINDOWFLAGS_IF, Timeout)
                WSetWndPosSiz(CurrentWindow, 200, 90, 400, 420)

                Play user(i) + "@veritestwin1.com"
                Play "{TAB}{TAB}large attach - "
                Play user(i)
                Play "{TAB}"
                Sleep(sleep_length)

                WToolbarButtonClk("@2", "&Insert")
                Sleep(sleep_length)
                WMenuSelect("File &Attachment...")
                Sleep(sleep_length)
```

```
                                Play "1mb_attach.exe"
                                Sleep(sleep_length)
                                Play "{ENTER}"
                                Sleep(sleep_length)

                                CurrentWindow = WFndWndC("large attach", "ATH_Note",
FINDWINDOWFLAGS, Timeout)
                                WSetWndPosSiz(CurrentWindow, 115, 78, 508, 420)
                                WToolbarButtonClk("@1", "Send")
                                Sleep(sleep_length)

                        ELSEIF (j = 3) THEN
                                Play "^(m)"
                                Sleep(sleep_length)
                                'Play "^(y)"
                                'Sleep(sleep_length)
                                'Play "{RIGHT}{RIGHT}"
                                'Sleep(sleep_length)
                                'Play "{ENTER}"
                                'Sleep(sleep_length)
                                Play "{Click 468, 251, Left}"

                                'Play "{TAB}{TAB}{TAB}{TAB}"
                                Sleep(sleep_length)
                                Play "^(a)"
                                Sleep(sleep_length)
                                Play "^(d)"
                                Sleep(sleep_length)
                                WTreeItemClk("@1", "Deleted Items", VK_LBUTTON)
                                Sleep(sleep_length)
                                Play "{Click 477, 266, Left}"
                                Sleep(sleep_length)
                                Play "{CTRL DOWN}{a DOWN}"
                                Sleep(sleep_length)
                                Play "{a UP}"
                                Sleep(sleep_length)
                                Play "d"
                                Sleep(sleep_length)
                                Play "{CTRL UP}"
                                Sleep(sleep_length)
                                Play "{LEFT}{ENTER}"

                                CurrentWindow = WFndWndC("Deleted Items - Outlook Express", "Outlook
Express Browser Class", FINDWINDOWFLAGS, Timeout)
                                WSetWndPosSiz(CurrentWindow, 164, 65, 600, 404)
                                WTreeItemClk("@1", "Inbox", VK_LBUTTON)
                                Sleep(sleep_length)

                        END IF

                        PRINT #1, "Completed run with action: " + str(j) + ": user: " + user(i) + ": " +
DATETIME$


                        IF j < (number_of_actions) THEN
                                j = j + 1
                        ELSE
                                j = 0
```

```
            END IF

        LOOP

        IF i < number_of_users THEN
            i = i + 1
        ELSE
            i = 0
        END IF
    LOOP

LOOP

End Scenario
```

Appendix D: Reliability Study File Activity Script

```
'$INCLUDE 'RECORDER.INC'

SetDefaultWaitTimeout(Timeout)

SUB StopTheTest (vtNotifyData AS VARIANT )
        STOP
End Sub

'Pressing <CNTL>-q halts execution cleanly.
ON KeyPress ("q",FCONTROL) CALL StopTheTest


DIM     filenames(3) AS STRING
filenames(0) = "10mbb"
filenames(1) = "4mbb"
filenames(2) = "32KBb"

DIM i
i = 0
DIM rndtime, rndtimestr$

'Change this to match the user number
DIM Astring$, logname$
'Astring = "1user"


Scenario "file_server"

' Check the resolution that this script was recorded on.
CheckResolution (800, 600)

WDesktopClk(373, 386, VK_LBUTTON)
Sleep(2.0)
Play "share{ENTER}"
Sleep(4.0)


'WTaskbarStartClk(VK_LBUTTON)

'CurrentWindow = WFndWndC(NULL, "DV2ControlHost", FINDWINDOWFLAGS_IF, Timeout)
'WSetWndPosSiz(CurrentWindow, 0, 12, 380, 558)
'WViewItemClk("@2", "My Network Places", VK_LBUTTON)
'Sleep(2.0)

'CurrentWindow = WFndWndC("My Network Places", "CabinetWClass", FINDWINDOWFLAGS_IF, Timeout)
'WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
'WViewItemDblClk("@1", "Share on Veritest-win1", VK_LBUTTON)

Open "C:\clientname.txt" FOR INPUT AS #2
LINE INPUT #2, Astring

logname = "C:\NET1_" + Astring + "_file.log"
Open logname FOR APPEND AS #1
```

```
PRINT #1, " "
PRINT #1, "Start of Test: " + DATETIME$
PRINT #1, "Server name: NET1"
PRINT #1, "Client name: " + Astring


' Test cleanup before start
DO WHILE i < 3
        'StatusBox "Looking for file" , , , , , TRUE
        'Sleep(1.00)
        'StatusBox Close
        Sleep(1.0)

        IF EXISTS ( "C:\Documents and Settings\" + Astring + "\Desktop\" + filenames(i) + ".exe") THEN
                'StatusBox "Found file" , , , , , TRUE
                'Sleep(1.00)
                'StatusBox Close

                Sleep(1.0)
                WDesktopClk(130, 254, VK_LBUTTON)
                Sleep(1.0)
                WDesktopItemClk(filenames(i), VK_LBUTTON)
                Sleep(2.0)
                Play "{DELETE}"
                Sleep(1.0)
                CurrentWindow = WFndWndC("Confirm File Delete", "#32770", FINDWINDOWFLAGS_IF,
Timeout)
                WSetWndPosSiz(CurrentWindow, 190, 200, 419, 140)
                Play "{ENTER}"
                Sleep(1.0)
        END IF

        IF EXISTS ( "\\Veritest-win1\Share\" + Astring +"\" + filenames(i) + ".exe") THEN
                'StatusBox "Found file" , , , , , TRUE
                'Sleep(1.00)
                'StatusBox Close

                'Sleep(2.0)
                CurrentWindow = WFndWndC("Share on Veritest-win1", "CabinetWClass",
FINDWINDOWFLAGS_IF, Timeout)
                WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
                WViewClk("@1", 125, 118, VK_LBUTTON)
                Play "1user{ENTER}"
                Sleep(1.00)
                Play "^(a)"
                Sleep(2.00)
                Play "{DELETE}"
                Sleep(1.00)

                CurrentWindow = WFndWndC(Astring, "CabinetWClass", FINDWINDOWFLAGS_IF,
Timeout)
                WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
                Sleep(1.00)
                Play filenames(i)
                Sleep(1.00)
                Play "{DELETE}"
                Sleep(1.00)
                CurrentWindow = WFndWndC("Confirm File Delete", "#32770", FINDWINDOWFLAGS_IF,
Timeout)
                WSetWndPosSiz(CurrentWindow, 190, 200, 419, 140)
                Sleep(1.0)
                Play "{ENTER}"

                CurrentWindow = WFndWndC(Astring, "CabinetWClass", FINDWINDOWFLAGS_IF,
Timeout)
                WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
                Sleep(2.0)
                WToolbarButtonClk("@2", "Up")
                Sleep(1.0)
        END IF
        i = i + 1
LOOP
i = 0

RANDOMIZE TIMER

DO WHILE TRUE

        'Sleep random time between 1 minute and 10 minutes
        rndtime = RND * 10 * 60 'Number of seconds to sleep
        if (rndtime < 60) THEN
                rndtime = rndtime + 60
        END IF
        PRINT #1, "Wait: " + STR$(rndtime) + " secs at: " + DATETIME$
'rndtime = 0
        DO WHILE rndtime > 0
                rndtimestr = "Time to next action: " + STR$(rndtime) + " secs"
                rndtime = rndtime - 1
                StatusBox rndtimestr , , , , , TRUE
                Sleep(1.0)
                StatusBox Close
        LOOP

        CurrentWindow = WFndWndC("Share on Veritest-win1", "CabinetWClass",
FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
        Sleep(2.0)
        Play filenames(i)
        Sleep(2.0)
        Play "^(c)"
        Sleep(2.0)
        WDesktopClk(70, 265, VK_LBUTTON)
        Sleep(2.0)

        Play "^(v)"
        Sleep(2.0)

        CurrentWindow = WFndWndC("Share on Veritest-win1", "CabinetWClass",
FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
        Sleep(2.0)
        Play Astring
        Sleep(2.0)
        Play "{ENTER}"

        Sleep(2.0)
        'WDesktopItemClk(filenames(i), VK_LBUTTON)
        'WDesktopItemClk(filenames(i), VK_RBUTTON)
        'WMenuSelect("Cu&t")
        WDesktopClk(70, 265, VK_LBUTTON)
        Sleep(2.0)
        Play filenames(i)
        Sleep(2.0)
        Play "^(x)"
        Sleep(2.0)

        CurrentWindow = WFndWndC(Astring, "CabinetWClass", FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
        Play "^(v)"
        Sleep(4.0)

        CurrentWindow = WFndWndC(Astring, "CabinetWClass", FINDWINDOWFLAGS_IF, Timeout)
        'WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
        'WViewItemClk("@1", filenames(i), VK_LBUTTON)
        Sleep(2.0)
        Play filenames(i)
        Sleep(2.0)
        Play "{DELETE}"
        Sleep(2.0)

        CurrentWindow = WFndWndC("Confirm File Delete", "#32770", FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 190, 200, 419, 140)
        Sleep(2.0)
        Play "{ENTER}"
        Sleep(2.0)

        CurrentWindow = WFndWndC(Astring, "CabinetWClass", FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 155, 53, 600, 505)
        Sleep(2.0)
        WToolbarButtonClk("@2", "Up")
        Sleep(2.0)


        PRINT #1, "Completed: " + filenames(i) + " at: " + DATETIME$

        IF i < 2 THEN
                i = i + 1
        ELSE
                i = 0
        END IF

        SLEEP(2.0)

LOOP

End Scenario
```

Appendix E: Reliability Study Print Activity Script

```
'$INCLUDE 'RECORDER.INC'

SetDefaultWaitTimeout(Timeout)

SUB StopTheTest (vtNotifyData AS VARIANT )
        STOP
End Sub

'Pressing <CNTL>-q halts execution cleanly.
ON KeyPress ("q",FCONTROL) CALL StopTheTest

DIM sleep_length
sleep_length = 2.0

DIM rndtime, rndtimestr$, loopnum

DIM Astring$, logname$
'Astring = "1user"

Scenario "print_server"

'Minimize the Visual Test window.
If GetHandle(GH_HWNDCLIENT) Then WMinWnd(GetHandle(GH_HWNDCLIENT))

' Check the resolution that this script was recorded on.
CheckResolution (800, 600)

Open "C:\clientname.txt" FOR INPUT AS #2
LINE INPUT #2, Astring

logname = "C:\NET1_" + Astring + "_print.log"
Open logname FOR APPEND AS #1
PRINT #1, " "
PRINT #1, "Start of Test: " + DATETIME$
PRINT #1, "Server name: NET1"
PRINT #1, "Client name: " + Astring

WDesktopClk(0, 0, VK_LBUTTON)
Sleep(2.0)
Play "my docs{ENTER}"
Sleep(sleep_length)

loopnum = 0

RANDOMIZE TIMER

DO WHILE TRUE

        'Sleep random time between 1 minute and 10 minutes
        rndtime = RND * 10 * 60 'Number of seconds to sleep
        if (rndtime < 60) THEN
                rndtime = rndtime + 60
        END IF
```

```
    PRINT #1, "Wait: " + STR$(rndtime) + " secs at: " + DATETIME$
'rndtime = 10
    DO WHILE rndtime > 0
        rndtimestr = "Time to next action: " + STR$(rndtime) + " secs"
        rndtime = rndtime - 1
        StatusBox rndtimestr , , , , , TRUE
        Sleep(1.0)
        StatusBox Close
    LOOP

    IF (loopnum = 0) THEN
        CurrentWindow = WFndWndC("My Documents", "CabinetWClass",
FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 154, 98, 600, 404)
        Play "small{ENTER}"
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Small.txt - Notepad", "Notepad", FINDWINDOWFLAGS,
Timeout)
        WSetWndPosSiz(CurrentWindow, 66, 66, 600, 411)
        WMenuSelect("&File\&Print...")
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Print", "#32770", FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 82, 82, 464, 416)
        Sleep(sleep_length)
        Play "{ENTER}"
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Small.txt - Notepad", "Notepad", FINDWINDOWFLAGS,
Timeout)
        WSetWndPosSiz(CurrentWindow, 66, 66, 600, 411)
        WMenuSelect("&File\E&xit")
        Sleep(sleep_length)

        loopnum = 1

    ELSEIF (loopnum = 1) THEN
        CurrentWindow = WFndWndC("My Documents", "CabinetWClass",
FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 154, 98, 600, 404)
        Play "medium{ENTER}"
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Medium.txt - Notepad", "Notepad", FINDWINDOWFLAGS,
Timeout)
        WSetWndPosSiz(CurrentWindow, 66, 66, 600, 411)
        WMenuSelect("&File\&Print...")
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Print", "#32770", FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 82, 82, 464, 416)
        Play "{ENTER}"
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Medium.txt - Notepad", "Notepad", FINDWINDOWFLAGS,
Timeout)
```

```
        WSetWndPosSiz(CurrentWindow, 66, 66, 600, 411)
        WMenuSelect("&File\E&xit")
        Sleep(sleep_length)

        loopnum = 2

    ELSEIF (loopnum = 2) THEN
        CurrentWindow = WFndWndC("My Documents", "CabinetWClass",
FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 154, 98, 600, 404)
        Play "large{ENTER}"
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Large.txt - Notepad", "Notepad", FINDWINDOWFLAGS,
Timeout)
        WSetWndPosSiz(CurrentWindow, 66, 66, 600, 411)
        WMenuSelect("&File\&Print...")
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Print", "#32770", FINDWINDOWFLAGS_IF, Timeout)
        WSetWndPosSiz(CurrentWindow, 82, 82, 464, 416)
        Play "{ENTER}"
        Sleep(sleep_length)

        CurrentWindow = WFndWndC("Large.txt - Notepad", "Notepad", FINDWINDOWFLAGS,
Timeout)
        WSetWndPosSiz(CurrentWindow, 66, 66, 600, 411)
        WMenuSelect("&File\E&xit")
        Sleep(sleep_length)

        loopnum = 0

    END IF

    PRINT #1, "Completed: " + DATETIME$

LOOP

End Scenario
```

Appendix F: SAMBA configuration file

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentry and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command "testparm"
# to check that you have not made any basic syntactic errors.
#
#======================= Global Settings =====================================
[global]

# workgroup = NT-Domain-Name or Workgroup-Name
   workgroup = WORKGROUP

# server string is the equivalent of the NT Description field
   server string = Samba Server

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
   hosts allow = 200.4.1.

# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
   printcap name = /etc/printcap
   load printers = yes

# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
# bsd, sysv, plp, lprng, aix, hpux, qnx
   printing = bsd

# Uncomment this if you want a guest account, you must add this to /etc/passwd
# otherwise the user "nobody" is used
; guest account = pcguest

# this tells Samba to use a separate log file for each machine
# that connects
   log file = /var/log/samba/%m.log

# Put a capping on the size of the log files (in Kb).
   max log size = 0

# Security mode. Most people will want user level security. See
# security_level.txt for details.
   security = share
```

```
# Use password server option only with security = server
# The argument list may include:
#   password server = My_PDC_Name [My_BDC_Name] [My_Next_BDC_Name]
# or to auto-locate the domain controller/s
#   password server = *
;   password server = <NT-Server-Name>

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
;  password level = 8
;  username level = 8

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
;   encrypt passwords = yes
;   smb passwd file = /etc/samba/passwd

# The following is needed to keep smbclient from spouting spurious errors
# when Samba is built with support for SSL.
;   ssl CA certFile = /usr/share/ssl/certs/ca-bundle.crt

# The following are needed to allow password changing from Windows to
# update the Linux sytsem password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
#        the encrypted SMB passwords. They allow the Unix password
#        to be kept in sync with the SMB password.
;  unix password sync = Yes
;  passwd program = /usr/bin/passwd %u
;  passwd chat = *New*password* %n\n *Retype*new*password* %n\n
*passwd:*all*authentication*tokens*updated*successfully*

# Unix users can map to different SMB User names
;  username map = /etc/samba/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /etc/samba/smb.conf.%m

# This parameter will control whether or not Samba should obey PAM's
# account and session management directives. The default behavior is
# to use PAM for clear text authentication only and to ignore any
# account or session management. Note that Samba always ignores PAM
# for authentication in the case of encrypt passwords = yes

;  obey pam restrictions = yes

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# Configure Samba to use multiple interfaces
# If you have multiple network interfaces then you must list them
# here. See the man page for details.
   interfaces = 200.4.1.250
```

```
# Configure remote browse list synchronisation here
#  request announcement to, or browse list sync from:
#     a specific host or from / to a whole subnet (see below)
;   remote browse sync = 192.168.3.25 192.168.5.255
# Cause this host to announce itself to local subnets here
;   remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
;   local master = no

# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
# if you already have a Windows NT domain controller doing this job
;   domain master = yes

# Preferred Master causes Samba to force a local browser election on startup
# and gives it a slightly higher chance of winning the election
;   preferred master = yes

# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
;   domain logons = yes

# if you enable domain logons then you may want a per-machine or
# per user logon script
# run a specific logon batch file per workstation (machine)
;   logon script = %m.bat
# run a specific logon batch file per username
;   logon script = %U.bat

# Where to store roving profiles (only for Win95 and WinNT)
#        %L substitutes for this servers netbios name, %U is username
#        You must uncomment the [Profiles] share below
;   logon path = \\%L\Profiles\%U

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's WINS Server
;   wins support = yes

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
#     Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
;   wins server = w.x.y.z

# WINS Proxy - Tells Samba to answer name resolution queries on
# behalf of a non WINS capable client, for this to work there must be
# at least one    WINS Server on the network. The default is NO.
;   wins proxy = yes

# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.
   dns proxy = no
```

```
# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis
;  preserve case = no
;  short preserve case = no
# Default case is normally upper case for all DOS files
;  default case = lower
# Be very careful with case sensitivity - it can break things!
;  case sensitive = no

#============================ Share Definitions ==============================
[homes]
   comment = Home Directories
   browseable = yes
   writable = yes
   guest ok = yes
;   valid users = %S
;   create mode = 0664
;   directory mode = 0775
# If you want users samba doesn't recognize to be mapped to a guest user
; map to guest = bad user


# Un-comment the following and create the netlogon directory for Domain Logons
; [netlogon]
;   comment = Network Logon Service
;   path = /usr/local/samba/lib/netlogon
;   guest ok = yes
;   writable = no
;   share modes = no


# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory
;[Profiles]
;    path = /usr/local/samba/profiles
;    browseable = no
;    guest ok = yes


# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
   comment = All Printers
   security = server
   path = /var/spool/lpd
# Set public = yes to allow user 'guest account' to print
   guest ok = yes
   writable = yes
   printable = yes
   print ok = yes
   browseable = yes

[HP]
   security = server
   path = /var/spool/lpd/HP
   writable = yes
   public = yes
   printable = yes
   print ok = yes
   print command = lpr -r -h -P %p %s


# This one is useful for people to share files
[beaufort]
  comment = Temporary file space
   path = /home
   read only = no
   public = yes
   guest ok = yes
   browseable = yes
   writable = yes
   share modes = yes
   only guest = yes

# A publicly accessible directory, but read only, except for people in
# the "staff" group
[public]
;   comment = Public Stuff
;   path = /home/user1
;   public = yes
;   writable = yes
;   printable = no
;   write list = @staff

# Other examples.
#
# A private printer, usable only by fred. Spool data will be placed in fred's
# home directory. Note that fred must have write access to the spool directory,
# wherever it is.
;[fredsprn]
;   comment = Fred's Printer
;   valid users = fred
;   path = /home/fred
;   printer = freds_printer
;   public = no
;   writable = no
;   printable = yes

# A private directory, usable only by fred. Note that fred requires write
# access to the directory.
;[fredsdir]
;   comment = Fred's Service
;   path = /usr/somewhere/private
;   valid users = fred
;   public = no
;   writable = yes
;   printable = no

# a service which has a different directory for each machine that connects
# this allows you to tailor configurations to incoming machines. You could
# also use the %U option to tailor it by user name.
# The %m gets replaced with the machine name that is connecting.
;[pchome]
;  comment = PC Directories
;  path = /usr/local/pc/%m
;  public = no
;  writable = yes

# A publicly accessible directory, read/write to all users. Note that all files
# created in the directory by users will be owned by the default user, so
# any user with access can delete any other user's files. Obviously this
# directory must be writable by the default user. Another user could of course
# be specified, in which case all files would be owned by that user instead.
[public]
;  path = /usr/tmp
;   public = yes
;   only guest = yes
;   writable = yes
;   printable = no

# The following two entries demonstrate how to share a directory so that two
# users can place files there that will be owned by the specific users. In this
# setup, the directory should be writable by both users and should have the
# sticky bit set on it to prevent abuse. Obviously this could be extended to
# as many users as required.
;[myshare]
;   comment = Mary's and Fred's stuff
;   path = /usr/somewhere/shared
;   valid users = mary fred
;   public = no
;   writable = yes
;   printable = no
;   create mask = 0765
```

Appendix G: Ease of Management IT Admin Journals

IT Administrator 1 (Experienced in Windows server software)

Windows Standard Server 2003 ease of administration journal

Preparing the hard drive

The hard drive had Linux installed on it so I deleted all of the Linux partitions during the installation. After deleting the partitions, I formatted the entire hard drive using the NTFS quick format. It took only 26 seconds to format the 17GB drive.

Installing the operating system with default settings

I installed the operating system on one large 17GB partition. I used all of the default settings during the installation except for the network setup. For the network portion of the installation, I chose the custom setup and gave the system a static IP address. Other than that, the installation was a default setup.

The entire installation finished in 31 minutes and 11 seconds. The time represented here is from when the system started to boot off of the operating system CD until it came to the Ctrl-Alt-Delete logon screen. In between that time, it installed all components and configured them accordingly.

Setting up email server

I used the “Manage Your Server” window to create the file server. I clicked on the “add or remove a role” within the window to add the file server. I used the custom configuration and added only the file server role.

I set the mail domain up as “mail.com”. After the installation was completed, I used the “Manage this Mail Server” to manage the server and create users. I created two users: “User1” and “User2”.

I opened Outlook Express on the client and configured the mail account to connect with the mail server. After I configured the client’s email account I tried to send and receive. The application would not send and receive properly. It kept asking for a proper account logon. I put the account information of “User1” and the password but it would not take it. I thought the issue may be because Active Directory was required so I updated the server to be a Domain Controller.

After promoting the system to a Domain Controller, I rebooted the system. Upon system boot, I received the error that one or more service failed, and to check the Event Viewer for more information. The service that failed during the boot process was the POP3 service. So to correct the error I tried removing the POP3 Server and then reinstalled. This corrected the problem with the POP3 service not starting.

After promoting the server to a Domain Controller and reinstalling the POP3 Server, I tried to send and receive with the client computer. I found that I received the same logon message. At this point, I realized that the logon account was User1@mail.com instead of just “User1”. Once I entered the correct user logon, everything worked fine. In all it took about an hour to updated the server to a Domain Controller, remove the POP3 Server, and then reinstall the POP3 Server.

I went back and verified that the POP3 Server could be configured without Active Directory installed on the system. When I verified this, I had the server configured as a stand-alone server in a workgroup called “workgroup”.

Getting a successful email connection via Outlook on the XP client

As noted in the section above, I could not get communication with the mail client the first time. I found out later that this was due to me using the wrong logon information. The logon account was User1@mail.com and not “User1”. After correcting this, everything worked fine.

I created the Email account information so that I logged on as User1. While logged on as User1 I sent email messages to another email client, User2. After I sent the messages to the User2, I changed my account information so that I was now logged on as User2. Then I performed a send and receive to verify that I receive the messages I just sent from “User1”. I received the messages without any problems.

Setting up file server

I used the “Manage Your Server” window to create the file server. I clicked on the “add or remove a role” within the window to add the file server. I used the custom configuration and added only the file server role.

At the time I did the above installation I had Active Directory in place and the client system was part of the network. I did verify after this that the client does not have to set the server as a domain controller and have Active Directory running to operate this file server.

Successfully copying a file from the server to the client

I was able to browse to the share through “My Network Places” and see the share. I could copy a file to it and edit an existing file without any problems. Note: I had the permissions set so that the user could copy the files. By default, the permissions are set to read only.

Setting up print server

Like the file server setup, I used the “Manage Your Server” window to create the print server. I clicked on the “add or remove a role” within the window to add the print server. I used the custom configuration and added only the print server role.

Since we did not have a printer installed on the system I just picked the “HP Color LaserJet 4600 PS” printer for the driver to install. I installed it as a local printer on lpt1. I then installed the printer as a network printer on the client.

Like the file server, I performed the installation above with the server setup as a Domain Controller with Active Directory running. However, I did go back and verify that the server could be configured as a print server without being set as a Domain Controller or having Active Directory running.

Successfully submitting a print job to the print queue on the server

After the printer was installed, I created a simple document with WordPad and then clicked print. The document went to the printer without problems. I verified this by looking in the print cue on the server. The document was there and ready to print.

Red Hat Linux 8.0 ease of administration journal

Preparing the hard drive

During the installation, I chose to run Disk Druid manually. I set up the hard drive with a 100MB boot, 1000MB swap file, and set root to fill the remainder of the disk space (16230MB). I used the EXT3 file system. It took less than a minute to format the hard drive.

Installing the operating system with default settings

During the installation, I chose to server installation option. I set the IP address manually as well as the host name. I accepted the default firewall settings of medium. At the screen to add additional services, I added KDE desktop, X Window System, and Gnome desktop. In addition, I also added mail server with sendmail and imap. The following were added by default:

- Windows file server,
- server configuration tools, and
- printing support.

Setting up email server

For the email server setup, I used the sendmail program that Red Hat installed. I configured the sendmail server as the “mail.com” domain. For the client, I used the Outlook Express program as a POP3 connection. I used the steps at the following web page to configure the sendmail server and the POP3 server. http://www.siliconvalleyccie.com/sendmail.htm

Getting a successful email connection via Outlook on the XP client

After the server was configured, I sent messages to one of the users I created. The Outlook client performed a send and receive without problems. However, I did have a delay in the message delivery. For some reason the messages delivered to the correct address but it took a while. There is apparently some configuration file that is telling all messages to be held in the queue. I can read “/var/spool/mqueue” and see the messages, but for some reason they are not being delivered immediately.

Setting up file server

To allow the Windows XP client computer to see the Linux server, I modified the smb.conf file. I left the default settings for workgroup, but changed the server name to “server”. I also created a user account in the “Red Hat user manager” and then added this users home directory in the “smb.conf”. I entered the following command into the configuration file:

```
[user’s home]
      path = /home/user
      valid users = user
      admin users = user
      public = yes
      writable = yes
      printable = yes
      create mask = 0765
```

After editing this file, I could see the server from my Windows XP client; however, when I tried to click on the server icon within the network browser, I received an error that I did not have permissions to access this server. I tried to edit the smb.conf file different ways to correct this issue, but none of the edits corrected the problem.

One of the edits that I made was regarding the password encryption. According to the Linux documentation, the user has to either edit the smb.conf file so that it accepts password encryption, or you have to set the Windows system to use plain text passwords. I set the Windows client to use plain text passwords. The steps below are the steps that I took:

To enable "PlainTextPassword" connections:
Windows XP Professional has menu options (below) that should be used to enable plain text passwords for SMB servers. The Windows XP Home edition requires manual addition of this "xp.reg" registry key.

1. Start -> Programs -> Administrative Tools
   We have seen some Windows XP systems that did not have Administrative Tools on the Programs menu. If it is not there do:

- Start -> Settings -> Control Panel -> Administrative Tools

or

- Start -> Control Panel -> Performance and Maintenance -> Administrative Tools

2. On the Administrative Tools Folder, double-click Local Security Policy.
3. On the Security Setting folder, click the plus sign next to Local Policies to expand it.
4. Double-click Security Options.
5. Scroll down to near the bottom of the list.
6. Double-click -> Microsoft network client: Send unencrypted passwords to connect to third-party SMB servers
7. Click the Enabled radio button.
8. Click OK
9. Close the Local Security Settings Window
10. Shut down Windows XP and reboot.
11. After rebooting, use the above procedure to check that Microsoft network client: Send unencrypted passwords to connect to third-party SMB servers shows the Local Security Setting set to Enabled.

Finally, I found that I was being blocked by the firewall that I installed with the default setting was blocking my client’s access. I changed the security level configuration to “No firewall” and then saved this setting. This overrode the current firewall setting and allowed me to connect to the server from the client.

Successfully copying a file from the server to the client

After getting past the issue with the client not being able to connect to the server, I was able to copy a file without any problems.

Setting up print server

To set up the printer, I used the “Red Hat Printer Config” program. I clicked on “New” and then choose “Windows Printer” from the Queue type. I entered the Share, Host IP, and Workgroup as the information from the server. For the user name and password on configuration screen I used the same user name that I did when I configured the smb.conf.

For the printer, I used the HP color laserjet 4500 postscript printer. It configured without any problems on the Linux server. After I finished installing the printer, I clicked on the “apply” button in the Red Hat Printer Config. This restarted the lpd service on the server.

Successfully submitting a print job to the print queue on the server

After I set up the printer on the server, I chose “Add a Printer” from within the Printer and Faxes window on the Windows XP client. During the installation process, I selected to add a network printer. The program found the printer on the Linux server and installed the printer. However, it requested that I install the appropriate Windows drivers, so I did. After the installation, it appeared that everything worked. However, the status of the printer in the Printer and Faxes window states “Access denied, unable to connect”.

I typed up a short document with Microsoft WordPad on the client system. Then I clicked on the print option. The document did not appear in the print Queue or give any error messages. I did not have a print device hooked up so I could not verify that the printout actually worked or not. According to all the Linux documentation these are the correct steps to install a network printer on the Linux server and Windows client.

IT Administrator 2 (Experienced in Linux server software)

Windows Standard Server 2003 ease of administration journal

Preparing the hard drive

I used the 'configure your server wizard' custom configuration option to set up the POP server. The 'configure POP3 service' screen sets the default authentication method as local windows accounts. I left the default. Because this system was not set up as a DNS server or Active Directory server, I was unsure what to put for 'E-mail domain name'. The system had no domain, only a workgroup 'etlnet'. So I created an email address of Compaq@etlnet. I don't suspect it will work because it is not a FQDN. I received an error message, so I placed Compaq@localhost.localdomain. I received another error message stating I could not use the '@' symbol. So, I placed 'etlnet' as the local domain, even though it was my workgroup. I selected 'next' and it appeared to take my entry. Next, I selected 'view the next steps for this role'. The help file stated that I created mailboxes. I exited the help file and configured the mailboxes manually. I returned to the 'manager your server' wizard and selected 'manage your mail server'. I selected the 'add mailbox' link in the POP3 service windows. I created Compaq@etlnet and user1@etlnet.

Configure XP outlook client to connect to POP3 server

I verified that I could ping the IP address of the server and the name of the server 'dell70296'. I clicked 'email' of the Start menu. It started the 'new connection wizard'. I selected 'next' and on the second screen, I selected 'set up a home or small office network' radio button. I then selected 'next' and 'finish'. I then went onto the 'network setup wizard' and selected 'next'.
At the 'select a connection method', I choose 'this computer connects to the internet through another computer on my network' radio button. I then selected 'next'. I kept the computer name and entered 'etlnet' for workgroup name. I then selected 'next' until the wizard was completed. I choose to complete the wizard and reboot system. After rebooting, I selected 'outlook express' again. It started 'internet connection wizard'.
For display name: I typed 'compaq'.
For email address: I typed 'compaq@etlnet'
For incoming mail: I typed 'dell70296' and for outgoing mail I typed 'dell70296'.

For internet mail logon account name: Compaq, password: password. I did not select 'log on using SPA'. The connection failed. After the system rebooted it changed my IP address (192.168.1.2) from static to dhcp client. I changed it back to the IP address I had before I ran the wizards. I again opened Outlook Express. The username showed as 'compaq' with the correct password. I was unable to log in. I changed the username to Compaq@etlnet and I could get in. I went to tools>account and created the user1 account. I noticed the 'internet mail logon' did not say anything about the appropriate syntax of the account name. With the Compaq account, I typed in the account name as 'Compaq' and was unable log in into the email server.

I choose to type in the account name as user1@etlnet, instead of user1.
I verified user1 and Compaq could email back and forth. It worked fine.

Trying to debug the username problem caused me roughly one hour of troubleshooting time. I recommend that Microsoft clearly state the syntax of the user name for logging onto a POP3 email server as username@domain or username@domain.com.

Red Hat Linux 8.0 ease of administration journal

I booted to Red Hat Linux 8.0 from the CD. The Red Hat CD loaded the correct Adaptec SCSI driver (aic7xxx), video driver (ATI Mach64-4mb) and network driver prior to displaying installation screen. It did not auto detect my 3-button Microsoft mouse. I choose the default generic 3-button PS2 mouse. The "online help" screen to the left of the installation screen detailed what mouse to choose if I had a Microsoft 3-button 'scroll' mouse under the "tip" section.

I had the following options to choose from: Personal desktop, Workstations, Server and Custom. I selected the 'server' installation. I then selected the default 'automatically partition' option. I manually configured the network connection. I choose the 'medium' security level setting for the firewall. I customized the Firewall setting to allow for the following incoming traffic WWW, FTP, and Mail. I added the NFS server port 2049 under 'other ports'. The default medium firewall setting Red Hat chooses blocks all NFS, WWW, FTP, and Mail.

Under package group selection, I choose to install the services I knew I would need to set up a mail, file and print server. Red Hat chose the following packages by default:

- Server configuration tools,
- Web server Windows file server,
- Printer support,
- no desktop,
- no text editor,
- no mail server,
- no FTP server,
- no DNS server,
- no development tools (in order to recompile the kernel for adding drivers), and
- no system tools for using Samba to perform filing sharing with Windows clients.

I chose to add the following services:

- KDE desktop environment,
- Gnome Desktop environment,
- Text editor for .conf files,
- Mail server,
- FTP server,
- Development tools for recompiling the kernel, and
- system tools.

I did not choose to select each 'individual package', but chose to go with Red Hat's defaults for each package group. Red Hat auto-detected the video card (ATI Mach64-4MB). The Red Hat default desktop is Red Hat Gnome v1.4.

Installation problems: Having to individually select certain server packages (Mail, FTP, etc…) and their related services after having selected a server installation was not very user friendly. If the administrator has selected a server installation, I believe all server services should be selected by default. In addition, having to choose to install the kernel source code manually after selecting a server installation is a potential problem. If there is any trouble with network drivers or anything other drivers after the installation, an administrator could not update the system using .rpm files without the kernel source code being installed. For most IT administrators, it would require some investigation, only then to discover that a reinstallation would be required to install the source code along with the operating system.

Email Server Setup

My first step to set up an email server was the 'start here' icon on the desktop. I then selected the most logical place 'server settings' icon. I found no reference to setting up an email server. I found only DNS services, HTTP server, NFS server and services. I selected the services icon. I scrolled down the services menu and selected the following ‘imap, imaps, ipop2, ipop3, and pop3s. Each daemon description stated it was dependent on the ‘xinetd’ service. I also selected ‘servers’ and ‘services’ which would enable me to view active servers via the all encompassing network daemon, ‘xinetd’. Next, I restarted the xinetd services and saved the configuration.

I then went to a command terminal to verify the services were up and running by typing the command ‘top’ and ‘ps –ef |more’. Both commands did not show any of the email server daemons were running.

I tried to the ‘rpcinfo’ command to make a RPC call to the RPC server. Again, I found no information regarding any of the email services I started. I reviewed the help utility from the start menu. I found this utility to be no help in setting up a email server. There was no information on any email services, such as pop or imap. I rebooted the server by typing ‘init 6’ in the command prompt.

I checked the /etc folder where most of the service configuration are kept. I selected the postfix folder and the postfix.cf text file. I read through the short comments regarding each configuration. Not knowing the exact configuration syntax, I decided to go with the Red Hat documentation CD. I loaded the documents into the default location /usr/share/doc. After looking through the HTML configuration doc files, I found /rhl-cg-en-8.0/ch-switchmeal.html. These documents were very difficult to read. This document described how to configure an MTA (mail transport agent or email server using sendmail or postfix). Sendmail is the default email server service.

I selected to stop the postfix service and start the sendmail via the command terminal from / directory with these commands /sbin/service postfix stop and /sbin/service sendmail start. The document referenced another more comprehensive document labeled: /rhl-rg-en-8.0. I browsed to /usr/share/doc/rhl-rg-en-8.0 and found ch-email.html.

I read through Sendmail help file and found the information to be generic with rough examples of how to set up an email server. There wasn’t anything very helpful for configuring the sendmail.cf file. After further reading, I saw the email reference guide referred to postfix as another email server option that loads by default. It referred to another document “Red Hat Linux Customization guide” to find more information on postfix.

I followed the steps to change the default mail service from Sendmail to Postfix. To do this, I selected the Start Menu > system settings > mail transport agent switcher. I used the wizard steps to complete this task. Once I completed this, I stopped the Sendmail service by typing /sbin/service/sendmail stop and started the postfix service by typing /sbin/service/postfix.
I did not find any documentation on the Red Hat documentation CD on setting up the configuration for postfix. I selected a howto document from www.postfix.org.

I made the following changes to the main.cf file. I changed the myorigin=$mydomain. I changed the mail server to be a domain-wide mail server by typing my destination=$myhostname localhost.$mydomain $mydomain. Lastly, I set up the server to trust anyone on the same subnet by typing: mynetwork_style=subnet.

I then realized that I needed the system to be on a DNS domain for the mail server to work correctly. I then read how to configure BIND from Red Hat 8.0: The Official Red Hat Linux Customization Guide. I returned to the Start menu > Server settings > Domain name service utility. I selected to ‘Add’ the master zone domain ‘Mail.com’. I set the following file name ‘mail.com.zone’ contact root@localhost, Primary Nameserver (SOA) DELL1500SC.mail.com. I added a nameserver for this domain through nameserver properties. Hostname ‘DELL1500SC.mail.com’ server by: ‘192.168.1.1’. I selected ‘OK’.

In the records section in the ‘Name to IP translations’ window, I highlighted ‘mail.com’ and selected edit. I added a mailexchangr. Host name: ‘DELL1500SC’ and set the priority to 1. I started the domain service demon ‘named’ through the

services menu and saved the configuration. I brought the server down to runlevel 4 by typing ‘init4’ and brought it back to runlevel 5 ‘init5’.
I typed in the command ‘dnsdomainname –v’ to see if my newly created domain existed and it did not. I only saw localhost. I continued to try to get the DNS service to read correctly. It did not.

I went to the following web address to find more information abound Postfix configuration of the main.cf file:
http://www.redhat.com/support/resources/howto/RH-postfix-HOWTO/x118.html. I found the howto to be the most user-friendly of all the howto documents that I found. The document clearly states that a domain is required. I followed the Postfix howto and uncommented and edited all required lines. I deleted all domain records from the Domain Name Service. I added mail.com and defined the host name and IP address. When I ran dnsdomainname –v I got the correct read.
I rebooted the server. The named service started successfully. I found an error with Sendmail and the local loopback adapter that I could not resolve. I choose to reinstall Red Hat.

After I reinstalled the software, I followed a Sendmail howto found here:
www.siliconvalleyccie.com/sendmail.htm.

I began configuring Sendmail again. I made only two changes to the sendmail.mc file. I selected the service to listen for incoming connections on all network adapters instead of the default loopback adapter. I added the stand-alone server hostname and domain name to the /etc/hosts file.

I edited the /etc/mail/access file to set up mail relaying setting. I added the client’s IP address to the access file with the RELAY message. I added mail.com to the /etc/mail/local-host-names file to help Sendmail define what domains it is responsible for. Next, I added a user to the server machine named ‘user1’ and add him to the adm and mail group. I went to /etc/mail/virtusertable/file. I added user1@mail.com in the first column and set the path for received mail for this user. I assumed it defaulted to /home/user1 on the mail server. I converted the text file ‘virtusertable’ to a .db file with the make command. Next, I added an alias address to the /etc/aliases file. I added webmaster@mail.com for the root email. I ran the ‘newaliases’ command to update the aliases.db file.

I configured the POP mail server with the Sendmail service. I started the imap service and restarted the network daemon xinetd. I configured the pop mail server in the /etc/xinetd.d/ipop3 configuration file. After following these steps, I emailed successfully. The mail was still not getting forwarded client to client through the server.

I typed the ‘mailq’ command to discover that all mail in /var/spool/mqueue was queuing up due to a DNS lookup error for mail.com. We did not set up a Linux box as a DNS server. We verified this typing ‘dnsdomainname –v’ and saw the system was localhost.localdomain. We sent email from user1@localhost.localdomain to troy@localhost.localdomain and it worked.

I continued to set the system up with the mail.com domain via the domain name services wizard. I defined the name sever and the hostname for the mail exchanger. I defined a reverse master zone. I edited the /etc/hosts. I changed 127.0.0.1 from linny.localhost.localdomain to linny.mail.com. with an alias as localhost. I edited the /etc/named.conf file. I changed zone “localhost” to refer to “mail.com.zone”, instead of “localhost.localdomain.zone”. This fixed the problem and mail could be freely exchanged between user1@mail.com and troy@mail.com.

XP Pro connection to email server

I opened Outlook Express > tools > accounts. Under server I selected POP3 server, Incoming mail (POP3): 192.168.1.1 and Outgoing mail (SMTP): 192.168.1.1. I used the user1 account, I created on the Linux server. During my attempts to log unto the Postfix-pop3 server, I received the following errors:

> For POP3 connection: Unable to logon to the server using Secure Password Authentication. Account: 'user-account', Server: 'server-name', Protocol:

POP3, Server Response: '.', Port: 110, Secure(SSL): No, Error Number: 0x800CCC18.

I disabled the encryption password on the Windows client in an attempt to get around the error. Administrative Tools > Local Security Policy -> Local policies folder > security options > ‘Microsoft network client: send unencrypted password to third-party SMB servers’. I right-clicked and selected properties. I changed the setting from disable to enable. Once the domain information was setup, the client could send and receive mail without errors.

Setting up file server

The first step was starting the smb services using the ‘services’ utility and saving the setup. The Samba configuration can either be done through the text-based smb.conf or you can use the SWAT samba web based utility. Red Hat 8.0 does not include the SWAT samba configuration utility. I made the following changes to the smb.conf- [global] workgroup=etlnet, uncommented guest account= pcguest, commented encrypt passwords=yes, smb passwd file= /etc/samba/smbpasswd, uncommented wins support =yes,

Copy files from file server to client and client to server

I disabled the default encryption password setting on client. I went to Administrative Tools > Local Security Policy. Go under Local policies folder > security options > ‘Microsoft network client: send unencrypted password to third-party SMB servers’. I right-clicked and selected properties. I changed settings from disable to enable. I selected for the Samba server to not use encryption for passwords. I was able to log on into /home/user1 and copy to and from Samba server. I tested the connection with encryption enabled on client and server and I could not access the server.

Setting up print server Success submission of print job to print queue on server

I added a HP Deskjet 815C printer to Red Hat Linux server via Start > System Settings > Printing.
I made sure the follow text was added the following setting to the smb.conf file

```
[global]
     printing = lprng
     printcap name = /etc/printcap
     load printers = yes
     log file = /var/log/samba-log.%m
     lock directory = /var/lock/samba

  [printers]
     comment = All Printers
     security = user
     path = /var/spool/lpd/hp815C
     browseable = yes
     printable = yes
     public = yes
     writable = no
     create mode = 0700
```

When I browsed from the client to hp815C on the samba server, I received the following error message:

‘hp815C on linny Access denied, unable to connect’.

I attempted to print a .txt document anyway and received the following pop up error message ‘A StartDocPrinter call was not issued’. I decided to add the following lines per a ‘SMB-how to’ linux help document from www.lpd.org.

```
[hp815C]
     security = user
     path = /var/spool/lpd/lp
     printer name = lp
     writable = yes
     public = yes
     printable = yes
     print command = lpr -r -h -P %p %s
```

I received the same error as before. I right-clicked on the ‘hp815C’ printer and got the error: ‘The server linny does accept remote request’. I went to check my firewall setting. My firewall was on ‘high’ setting by default. I changed it to ‘no firewall’ and I lost being able to browse the samba server.

VeriTest (www.veritest.com), the testing division of Lionbridge Technologies, Inc., provides outsourced testing solutions that maximize revenue and reduce costs for our clients. For companies who use high-tech products as well as those who produce them, smoothly functioning technology is essential to business success. VeriTest helps our clients identify and correct technology problems in their products and in their line of business applications by providing the widest range of testing services available.

VeriTest created the suite of industry-standard benchmark software that includes WebBench, NetBench, Winstone, and WinBench. We've distributed over 20 million copies of these tools, which are in use at every one of the 2001 Fortune 100 companies. Our Internet BenchMark service provides the definitive ratings for Internet Service Providers in the US, Canada, and the UK.

Under our former names of ZD Labs and eTesting Labs, and as part of VeriTest since July of 2002, we have delivered rigorous, objective, independent testing and analysis for over a decade. With the most knowledgeable staff in the business, testing facilities around the world, and almost 1,600 dedicated network PCs, VeriTest offers our clients the expertise and equipment necessary to meet all their testing needs.

For more information email us at info@veritest.com or call us at 919-380-2800.

Disclaimer of Warranties; Limitation of Liability:

VERITEST HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, VERITEST SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT VERITEST, ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL VERITEST BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL VERITEST'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH VERITEST'S TESTING. CUSTOMER’S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.