



BY GUY C. YOST

Streamlining Management of WinFrame Servers: Part II

Routine tasks on both NT/WinFrame and NetWare platforms can be automated using the techniques presented in this month's article.

LAST month, Part I of this series looked at strategic designs for implementing large-scale WinFrame installations as well as integrating NDS using non-conventional means. I examined how to

- ◆ leverage the WinFrame multi-user OS as a network client
- ◆ use a "reverse" gateway approach to allow NDS users to authenticate to the WinFrame desktop
- ◆ manage WinFrame Session drive mappings
- ◆ make distribution of files across multiple WinFrame servers easy

This article will explore techniques to automate routine tasks on NT/WinFrame and NetWare platforms, and the concluding article will examine how to streamline the creation of WinFrame servers for initial roll-out, maintenance or disaster recovery. The strategies presented can also be applied to the Citrix MetaFrame environment, which uses NT 4 as the base OS instead of NT 3.51.

TASK AUTOMATION

In any network there will be routine tasks required to keep the system running smoothly. UNIX administrators have long relied on CRON, the clock daemon that spends most of its time asleep but wakes up once a minute to see if there is any work to be done as directed by the CRONTAB file. NT and NetWare have their versions of CRON, as described in the following sections.

Automating Tasks on NT or WinFrame

NT's version of CRON is called AT.EXE, a command-line utility that schedules repetitive tasks weekly, daily, hourly, or by

the minute. AT relies on NT's schedule service to operate, and you may need to adjust how the schedule service interacts with the system for unattended operations. Specifically, the schedule service runs as the NT security entity "SYSTEM", which might not have all the rights required to perform your automated tasks, especially operations that involve file transfer and manipulation. To adjust the security for the schedule service, go to NT's Control Panel, start the Services applet, then highlight and double-click on the Schedule service (listed alphabetically). The service startup configuration screen shown in Figure 1 will be displayed.

Simply enter the user information for the service so it can operate as a security equivalent of that user. You'll need to stop and restart the service in order for the changes to take effect. To configure AT, you can get help from the command line by typing AT /?. However, there is a utility available called WINAT.EXE that presents a GUI interface for the AT command and is much easier to use. You can obtain WINAT.EXE from the NT Resource Kit in the I386\CONFIG\ directory or from www.microsoft.com. Using the WINAT interface, you can add numerous entries as shown in Figure 2.

Note that the example in Figure 2 is calling a batch file, SHDOWNP.BAT. In a NetWare or NT network environment, it is also possible to launch remote (or centrally stored) batch files by using a UNC path in the command field. For example, the command

```
\\NWFS1\SYS\PUBLIC\CLEANUP.BAT
```

could be used in the command line to launch CLEANUP.BAT stored on the NetWare server named NWFS1. When

accessing remote commands be sure that the schedule service is running as a (NetWare or NT) user with appropriate rights known to the remote system that is hosting the command files.

SHDOWN.BAT was created to automatically shutdown and reboot a list of WinFrame servers at off hours or during an emergency shutdown incident. We discovered that user session performance improved when the WinFrame servers were restarted regularly as opposed to leaving them up for weeks on end. The NT resource kit comes with two utilities called SHUTCMD.EXE and SHUTGUI.EXE (found in the \i386\netadmin directory), but we discovered a piece of freeware software called Remote Shutdown by Jon Roch-Berry that for some reason worked more reliably with WinFrame. You can download Remote Shutdown from <http://dialspace.dial.pipex.com/town/estate/py22/Software.html> or from www.shareware.com.

Using “old-fashioned” network management practices in new environments can pay off — after all, older and proven management techniques withstand the times for a reason.

REMOTESH.EXE operates with a Windows GUI if no parameters are given, or you can shut down systems from the command line as demonstrated by SHDOWNP.BAT, which is presented in Figure 3. Note that this batch file logs all command activity output to C:\REMSH\SHDOWN.LOG and uses the utility’s seven return codes so you can verify when it ran and whether or not any problems were encountered. The key to leveraging the FOR loop is having sequentially named servers such as wfserver01, wfserver02, wfserver03, and so on. Although this example shows how to use a particular utility with the AT service, keep in mind that any batch file or utility can be automated in the same manner.

Automating Tasks on NetWare

Recall from last month’s article that there are technical issues with using IntranetWare’s NT client version 4.11a to authenticate WinFrame users into an NDS environment because the client was designed for single-user access. Another caveat using the IntranetWare client was discovered when users disconnected from the WinFrame servers. Specifically, when users disconnect from the WinFrame servers using any of the several available methods — “logoff” from the File menu or Program Manager’s upper left corner control menu or “Close” from the Session manager’s upper left control menu — they officially disconnect and log out from the WinFrame session but do not log out of NDS. Even using SPX watchdog to close inactive connections doesn’t work because the NetWare server “sees” the MAC address of the WinFrame server as still being alive and therefore, does not clear the connection. From the NetWare Monitor console, the dead sessions appear as unlicensed connections, denoted by an asterisk next to the connection name. NetWare’s CLEAR STATION ALL command releases all pending sessions, but we had to issue the command at night when the system was not being used.

Figure 1: Configuring WinFrame’s Schedule Service



Figure 2: Using WINAT to Schedule Routine Tasks



Although server task automation doesn’t come with NetWare, Novell makes a utility called CRON.NLM available on their FTP site, and like its UNIX model, it uses a simple text file called CRONTAB to instruct its actions. Once a minute it reads SYS:\ETC\CRONTAB. Any server console commands scheduled in CRONTAB matching the current date and time are executed. CRON also records all actions in the log file SYS:\ETC\CRONLOG so you can track its history of success or failure. CRON can be downloaded from Novell’s FTP site at <ftp://ftp.novell.com/pub/updates/nwos/inw411/cron5.exe>. CRONTAB’s syntax is simple and identical to its UNIX predecessor. Each line entry has six fields, each separated by tabs or spaces:

- ◆ minute
- ◆ hour
- ◆ day-of-month
- ◆ month
- ◆ day-of-week
- ◆ command

Each line is checked in the order listed, and any entry matching the current time is executed. The entry “*” matches anything.

Figure 3: Automating Your Server Shutdown, SHUTDOWNP.BAT

REM shdwnp.bat, written by Guy C. Yost, Redstone Consulting 7/15/98 to automatically REM shutdown and reboot series of WinFrame servers for operations and maintenance

```
@echo off
REM Protect against accidental use by requiring explicit parameter before executing
command
if not %1==.ifyouinsist. goto out
REM Use Environment variables to specify server set and log file to record success or
failures
REM of each instance of this routine's execution.
SET SERVERSET="01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16"
SET SHLOG="C:\REMSH\SHDOWN.LOG"
echo ++++++%SHLOG%
REM the program resides in C:\REMSH
C:
CD C:\REMSH
REM The /r and /f parameters are used to force closure of applications and initiate reboot
of REM the server.
for %%c in (%SERVERSET%) do remotesh.exe /remote:wfserver%%c /r /f>>%SHLOG%
if errorlevel 6 goto E6
if errorlevel 5 goto E5
if errorlevel 4 goto E4
if errorlevel 3 goto E3
if errorlevel 2 goto E2
if errorlevel 1 goto E1
REM REMOTESH has 7 levels of return codes for troubleshooting
echo Success (no error)>>%SHLOG%
goto out
:E1
echo Error 1 - NOT_NT>>%SHLOG%
goto out
:E2
echo Error 2 - BAD_PARAMS>>%SHLOG%
goto out
:E3
echo Error 3 - NO_PRIVS>>%SHLOG%
goto out
:E4
echo Error 4 - BAD_MACHINE>>%SHLOG%
goto out
:E5
echo Error 5 - NOT_AVAILABLE>>%SHLOG%
goto out
:E6
echo Error 6 - ACTION_FAILED>>%SHLOG%
goto out
:out
```

Figure 4: CRONTAB

```
#Min Hr Dat Mo Day Command
#clear connections at 02:30 am
30 2 * * * clear station all
#Run NRS services between 2:31 and 6:30
31 2 * * * load NRS
30 6 * * * unload NRS
#reboot server after Service Pack is applied on 9/15
00 3 15 9 * ipl.ncf
```

Figure 5: The Logout Routine

```
@ECHO OFF
P:\MAP ROOT K:=NWFS01\DATA:LOGS>NUL
P:\FDATE /Ff /Ot1>>K:\LOGOUT.LOG
Echo %DATE%,%TIME%,%USERNAME%,%COMPUTERNAME%,%CLIENTNAME%,%WINSTATIONNAME%>>K:\LOGOUT.LOG
MAP DEL K:>NUL
LOGOUT
LOGOFF
```

A pound sign (#) is a comment. Valid values are:

- ◆ minute (0-59)
- ◆ hour (0-23)
- ◆ day-of-month (1-31)
- ◆ month (1-12)
- ◆ day-of-week (0-6) 0 = Sunday

We set up the CRONTAB file shown in Figure 4 to execute a few commands nightly. The first entry clears out the dead user connections, while the second and third entries load and then unload NetWare Replication Services. Due to the heavy processor overhead associated with NRS v1.2, we chose to only run NRS at night. The fourth entry performs a one-time “as needed” server reboot after hours when we apply packs during business hours. The two commands reside in the IPL.NCF file:

```
DOWN
RESTART SERVER
```

To run CRON, simply save your CRONTAB text file to SYS:ETC, copy CRON.NLM to the SYS:SYSTEM directory and CRON.MSG to the NLS directory off of SYS:SYSTEM, and type ‘LOAD CRON’ from the server console.

In general, any NetWare server console command, NCF or NLM file that can be loaded at the server prompt can be automatically executed using the CRON.NLM.

To help alleviate the “dead session” problem during business hours, we also created a logout routine that was put on the desktop, and stressed the importance of using our LOGOUT icon to end users’ sessions. The logout routine is shown in Figure 5. The FDATE utility used in the example is another freeware jewel that performs several functions; in this case, writing the current date and time to a central log file so we could track when users logged out. (FDATE can be downloaded from www.shareware.com.) The environment variables %DATE% and %TIME% were set in the NetWare login script indicating the user’s login time. We import the contents of LOGOUT.LOG into a spreadsheet program and produce reports showing peak usage times and average session duration — a very valuable reporting tool for forecasting future capacity needs. Although WinFrame comes with some “WinStation session” usage parameter hooks in the Performance Monitor

Figure 6: Clearing Out Defunct Print Jobs Across Servers, CLRSPPOOL.BAT

```
CLRSPPOOL.BAT:

D:
CD D:\WINFRAME
SET SHLOG=D:\WINFRAME\CLRSPPOOL.LOG
SET SERVERSET="01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 19 20 21 22"
echo +++++>>>SHLOG%
for %%c in (%SERVERSET%) do sc.exe \\wfserver%%c stop spooler>>>SHLOG%
for %%c in (%SERVERSET%) do CALL purgsp1.bat wfserver%%c
for %%c in (%SERVERSET%) do sc.exe \\wfserver%%c start spooler>>>SHLOG%

PURGSL.BAT:

net use k: \\%1\d$>>>SHLOG%
del k:\winframe\system32\spool\printers\*.spl>>>SHLOG%
del k:\winframe\system32\spool\printers\*.shd>>>SHLOG%
net use k: /delete>>>SHLOG%
```

utility, they only show real-time system usage and can't automatically (unattended) create an ongoing history of concurrent usage.

SPOOL MANAGEMENT

We found that often the spool directories on the servers would contain defunct print jobs caused by printing to an invalid LPT port or using a mismatched driver. Because each server hosts its own printer spools, it became tiresome logging into dozens of servers to clear out their dead print jobs regularly. I looked at and evaluated several products that claimed to have "Enterprise" management abilities for distributed printer control but came up short looking for a simple tool that would just clear out all the servers' print spools in unattended mode.

Going back to good old batch files, I created the routine in Figure 6 to do the job at night, as invoked by the WINAT service. The first batch file, CLRSPPOOL.BAT, calls a second batch file, PURGSPL.BAT, in

between stopping and restarting the NT spooler service. This was necessary because when the spooler was active, it held the current print job file open and would not allow it to be manually deleted. Like the other examples, a log file (CLRSPPOOL.LOG) is used to track the success and failure of the automated routine, and the main batch file is initiated from a central server that runs the utility in a distributed manner.

The command line utility SC.EXE is a Microsoft developer utility (available from <ftp://ftp.microsoft.com/bussys/winnt/winnt-public/>). It retrieves information about services, and also allows you to stop and start services from the command line.

SUMMARY

WinFrame servers are definitely taking a strong foothold in the corporate networking environment, yet the technology comes with its own set of issues and caveats that are only magnified when dealing with large numbers

of servers. Using "old-fashioned" network management practices in new environments can pay off — after all, older and proven management techniques withstand the times for a reason. At the same time, new practices can open the doors to greater productivity. The trick is to always insist on and search for the ideal solution, regardless of whether it's based on a new or old idea.

Using the techniques introduced, you can save a lot of time and avoid frustrating repetitive tasks that are prone to errors. The concluding article in this series will examine how to streamline the creation of WinFrame servers for initial server farm roll-out, maintenance and disaster recovery. 



NaSPA member Guy C. Yost is the owner of Redstone Consulting, an IT management consulting firm in New York. He has authored several books on networking for Que Publishing, including Learning NetWare 4.1, and NetWare 4.1 SmartStart, and contributes to Technical Support magazine as an author, columnist and technical editor. Guy also develops and conducts seminars on networking with Windows NT, UNIX, NetWare and Internet/intranet technologies across the United States and Canada. He can be reached at (518) 674-5606 or gyost@logical.net.

©1998 Technical Enterprises, Inc. For reprints of this document contact sales@naspa.net.