Automated Deployment of Windows 7 and Application Software on Large Installations of Computers Having Similar Hardware

Gopesh Tiwari and Narendra S. Rathore

***Abstract*—This paper describes an automated procedure to deploy a large number of identical computers with Windows 7 (preferably OEM version) along with a wide variety of application software. The normal Windows Deployment Service (WDS) has been tweaked with, to get a methodology which is suitable for a customized deployment. A quick overview of the setup, the scripts involved and overall methodology has been presented. The advantages over deployment method using conventional WDS and Windows Automated Installation Kit (WAIK) have been explained. Finally, the possibility to harness Intel vPro technology to remotely deploy the computers has been discussed.**

***Index Terms*—Deployment, operating system, WDS, WAIK, cloning, imaging, automated installation, PXE, VPro.**

I. Introduction

System administrators at an installation site having a large number of computers with similar hardware generally find it challenging to deploy OS and a long list of applications identically on all computers. Many times, they take up the longer way of installing OS and application software independently on each computer. There exists disk cloning software like Symantec Ghost, or Acronis Snap Deploy which can be used to do the initial disk cloning and thereafter to configure each computer separately. However, these approaches are not only time consuming but also prone to errors. Further this software requires lot of manual intervention in the post-cloning operation to complete the job.

Computers today have a PXE boot support. There exist several solutions for computer deployment with Linux or Windows based on PXE boot. These solutions harness the PXE boot capability of the computer to boot it and then use a third party tool like Symantec Ghost or Acronis True Image to restore a preconfigured image on the client computers. But these approaches also lack total automation and simplicity. Post-cloning configurations are still necessitated.

Generally, at such installation sites, it is required to have a system which can be used to deploy/redeploy the computers with minimum or no manual intervention by even a poor computer skilled person after a one time customization of the system.

Manuscript received November 5, 2013; revised January 27, 2014.

The authors are with the Computer Centre, Indian Institute of Technology Kanpur, Kanpur, India (e-mail: gopesh@iitk.ac.in, narensr@iitk.ac.in).

A. Challenges in Developing an Automated OS and Application(Re)Deployment System

A typical deployment procedure for deploying Windows OS and applications involves many steps. To start with, a reference image is created on a reference computer. The target computers are then booted via some boot image - through local DVD drive or through network. The reference image is restored to all target computers. After the restore the target computers are booted through local hard disk to log in with administrative privileges. The target computers either get the IP address through DHCP or they are given an IP address manually. A computer name is assigned to each computer and then they are joined to an Active Directory (AD) domain. Application software licensing is an added challenge. Software using network licensing pose no problem. But software using volume licensing, like the OS itself and MS Office, for example, make it necessary to include further scripting to manage their licenses. Once the target computers become members of AD domain they are ready for use.

Very often one may need to redeploy the computers because of some malfunctioning software, or the requirement to update/add new software. Thus there is a need to create a system which not only automates all these steps involved in the deployment, but also provides a way to do the operation remotely without visiting the individual computers. In a typical scenario, size of the system image is large enough to consume a lot of time to deploy a single computer. This has motivated to design a procedure to support multiple servers for hosting the system image so that the operation can be done in parallel to reduce the overall total deployment time. Further, it helps not only to minimize the need of third party tools but also make the system easy to be operated by a person having no in-depth knowledge of the tools involved.

II. Preparing for the Setup

Windows Deployment Service (WDS) [1] is installed and configured in the Active Directory. A DHCP server is essential for WDS. WAIK [2] is installed to add boot image for Windows PE as default option to the WDS boot images list. Further, it may be required to add the network driver for the target computer to the Windows PE boot image [3]. Using dism or imagex tool (components of WDS) the file "statrecovery.bat" [Listing1] is inserted in the WinPE boot image in the Windows\System32 folder. To do this, the

WinPE WIM image is mounted to any directory using dism tool and then "startrecovery.bat" file is copied to the Windows\system32 folder available under the mounted directory. Now the WIM image is un-mounted using dism tool with commit switch.

The target computers are tested to boot into Windows PE from the WDS using the PXE boot option. These computers must be configured for using the Intel vPro technology (optional) in case remote deployment option is required.

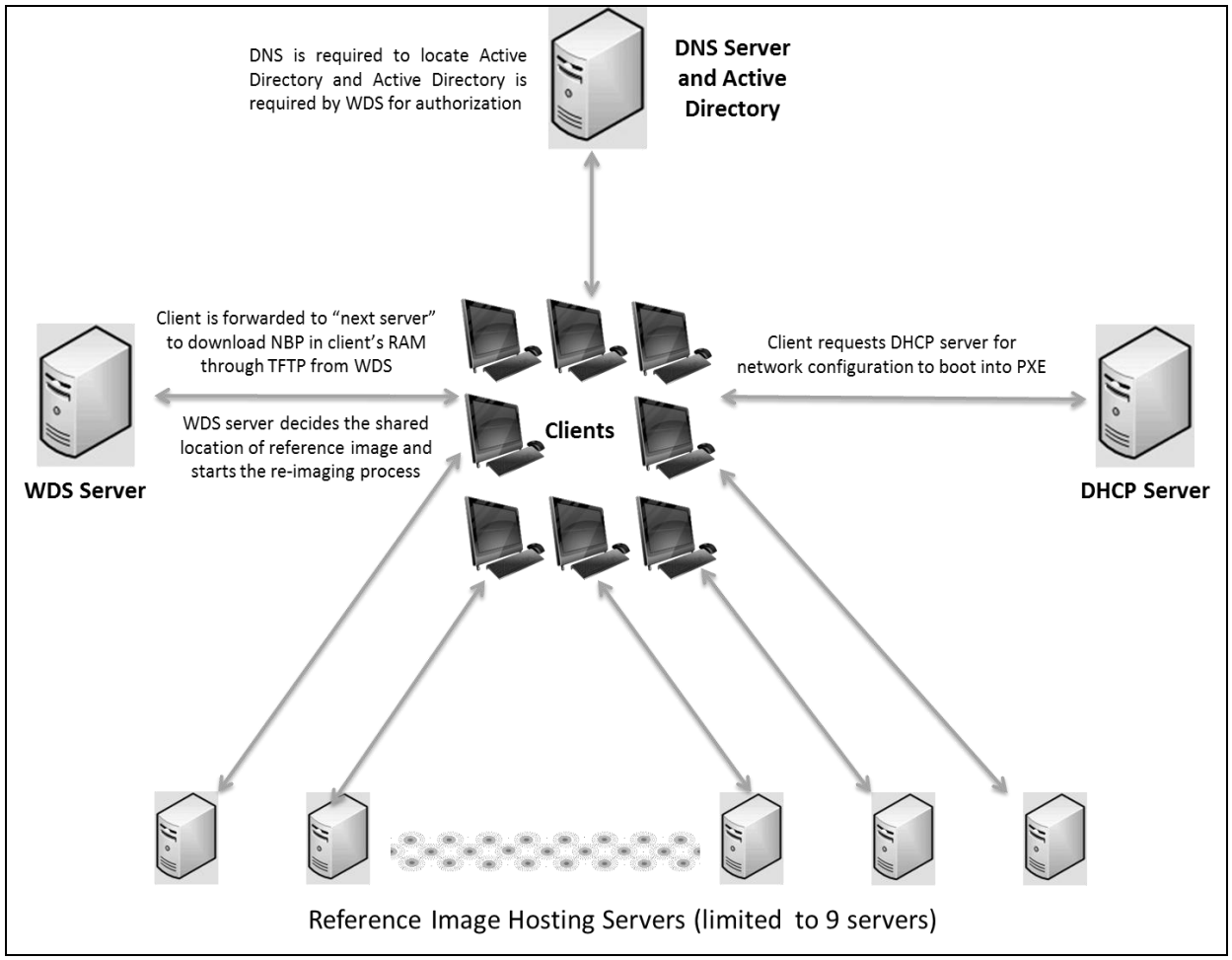

Fig. 1. Setup showing all components involved.

III. Outline of the Automation Scripts

A bundle of nine batch files, ini files, and vbs scripts carry out the deployment process in an automated way. The function of each file is discussed below.

1) Start.reg
2) Start.bat
3) Ip2static.bat
4) Insert.reg
5) IPv4_pcName.vbs
6) JoinDomain.bat
7) Delete.reg
8) DiskOption.ini
9) JoinDomain.vbs

"Start.reg" and "Start.bat" are the initialization scripts which start the automation process and call the other scripts in order of execution. "Start.reg" adds a value to the "Run" key:

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
"AutoDep"="C:\Windows\Automate\Start.bat"
```

"Start.bat" calls the remaining scripts in order of execution

"ip2static.bat" first checks for availability of DHCP server. If DHCP response is not received, it shuts down the computer after popping up an error message. Otherwise, it retrieves the IP address of computer provided by DHCP server and converts the same IP address to static IP address with predefined subnet, default gateway primary DNS server.

"Insert.reg" inserts a value to "Run" key. This is used once the computer reboots after executing the "IPv4_pcName.vbs" file (called by "ip2static.bat"):

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]"AutoDep"="c:\\Windows\\Automate\\JoinDomain.Bat"
```

"IPv4_pcName.vbs" retrieves the IP address to generate a computer name with the help of its last bit. The script issues command to open system properties dialog box. The function objShell.SendKeys is used to send key strokes to the open system properties dialog box. The computer is rebooted automatically after this script ends.

```
:: Filename startrecovery.bat to be integrated in the WinPE boot image
@ECHO OFF
Title Windows Recovery Tool For IIT Kanpur
Color 7c
net use y: \\<WDS Server>\Lab <password> /User:Administrator
cls
CALL \\<WDS Server>\Lab\server_assign.cmd

:: Call the 'server_assign.cmd' file to select the backup server and start recovery process
:: File name server_assign.cmd
:: This file can be put on any network share on the WDS server with proper permissions so that users cannot access it. This file is not put in the WinPE image so that it can be easily edited as required for changing backup server details.

@Echo Off
SET token=
for /F "delims=" %%i in (\\<WDS Server>\Lab\tokenPass.txt) do set token=%%i
IF %token%==1 GOTO SERVER1
IF %token%==2 GOTO SERVER2
.
.
IF %token%==9 GOTO SERVER9

:SERVER1
Color 0E
net use z: /delete /yes
net use z: \\<BackupServerIP>\Lab <PASSWORD> /User:Administrator
cls
Echo.
Echo.Windows 7 Backup on Server 1
Echo.
ping -n 2 127.0.0.1 > nul

:: Put a delay of 2 seconds

SET tmp=
Wbadmin get versions -backupTarget:\\<BackupServerIP>\Lab |FIND "Version" > %temp%\tmpVersion.txt

:: Find the backup version from the Backup Sever where backup is kept and
:: store the Version value to 'tmpVersion.txt'

For /F "delims=" %%i in (%temp%\tmpVersion.txt) do set tmp=%%i

:: Extract the content of 'tmpVersion.txt' file and store it to a variable 'tmp'

del %temp%\tmpVersion.txt
Echo Backup Version = "%tmp:~20,16%"
Echo.

:: Display the Backup Version

Echo 2 > \\<WDS Server>\Lab\tokenPass.txt

:: Write the token value for using next backup server
wbadmin start sysrecovery -recreateDisks -restoreallvolumes -version:%tmp:~20,16% -backuptarget:\\<BackupServerIP>\Lab -quiet

:: Start the system recovery process with 'WBADMIN' tool for full system recovery

goto:QUIT

:SERVER2
```

```
:: ----------Same code with different backup server address----------

goto:QUIT

:: Similarly populate support for more servers

:SERVER9

:: ----------Same code with different backup server address----------

Echo 1 > \\<WDS Server>\Lab\tokenPass.txt

:: Write the token value 1 for selecting the first server again as this is the 9th server.

goto:QUIT

:QUIT
Wpeutil Reboot
```

Listing 1. Contents of StartRecovery.bat and associated files.

"JoinDomain.bat" first checks whether a domain controller is responding. It shuts down the computer if it does not get a response from any domain controller after displaying error message. Otherwise, the registry entry previously created by "Insert.reg" is deleted. Further, any pending software activations are taken care of at this stage. In our case the process of activation of Microsoft Office Standard 2010 executes by defining the KMS Server IP address and port number of KMS Server. In this batch file, a disk utility is also used for removing the drive letter of additional partition(s) with the help of DiskOption.ini file. At the end of this file "JoinDomain.vbs" is initiated for Domain joining process, which is mostly a collection of commands using the function objShell.SendKeys to send appropriate key strokes to dialog boxes, called by the script. A segment of the code explains it:

```
'* Set System Performance Options (best
performance)

objShell.Run "control sysdm.cpl"
Do While objShell.AppActivate("System
Properties") = FALSE
WScript.Sleep 500 '* loop sleep until
window is available
Loop
objShell.SendKeys "{home}"

'* ensure general tab focus (default
focus on invoke)
objShell.SendKeys "{right 3}"

'* set focus on Advanced tab
objShell.SendKeys "%c"

'* Performance section, settings button
objShell.SendKeys "{TAB}"
objShell.SendKeys "%d"
objShell.SendKeys "{TAB}"
objShell.SendKeys "Your Domain"
WScript.Sleep 500
objShell.SendKeys "{TAB}"
objShell.SendKeys "{ENTER}"
```

IV. Outline of Overall Procedure

One of the target computers is chosen as the reference computer. Application software as required by the site are installed on this computer which has Windows 7 OEM version pre-installed. After joining it to active directory domain, the settings of Windows and application software are customized as per preference of the site environment. Exhaustive testing of the settings is done for a non-privileged user. Once satisfied, the reference computer is removed from the domain.

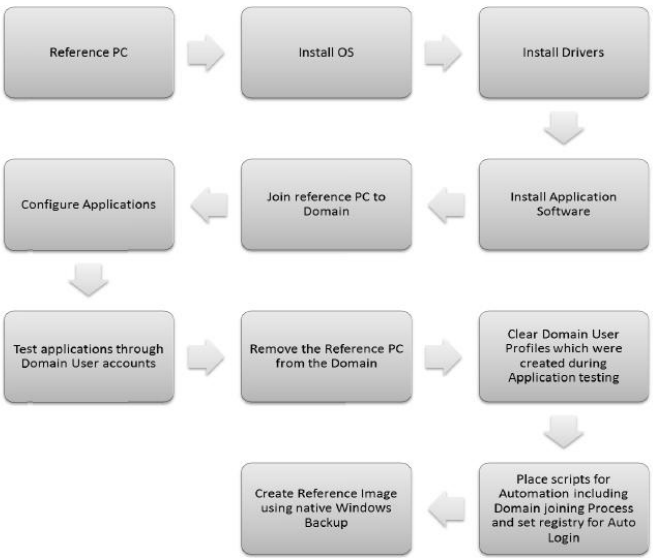

Fig. 2. Process outline for preparing reference image.

Automation scripts and registry settings are added to the appropriate path so that on reboot the computer would do all required post-imaging tasks after auto logon with administrative privileges. It gets an IP address from the DHCP server. The IP address is based on MAC reservation so that a particular computer always gets the same IP address. Though this would have worked, we further assign

the same IP address as static to the computer with predefined other network settings. We do this for two reasons. One is to avoid network outage in the lab in case of unavailability of DHCP server. Another reason is that there may be a mix of OS platforms. There is a common DHCP server for all platforms. In certain cases, for example, in conducting a computer based examination, it is required to segregate the lab computers from the overall network so that only the segment of the lab where the examination is to be conducted is on restricted network. The computers are locally connected to a server, which is normally a laptop, hosting the examination system. In this scenario, keeping the static IPs is very helpful. Moreover, the load on DHCP server is considerably reduced as it is only required when the computers are being (re)deployed.

A. Naming the Computers

We use a naming convention like WinPCXXX for the computer name where XXX is the last octet of the IP address. For example, the computer with IP 172.31.5.204 could be assigned the name WinPC204. Using this convention, the script assigns a computer name to the computer. After this step the computer is joined to the domain.

The naming convention limits the total number of computers deployable in a group to 254. However, this does not limit the total number of computers deployable in an institution. It is advisable to divide the computers into groups based on department or building or any other parameter, each group containing less than 255 computers. A different prefix based on the name of the group can then be used for the naming convention. In fact, different deployment system can be put for different groups.

A system image of the reference computer is made using natively installed windows backup. This is done just after removing the computer from the domain and adding the automation scripts. Multiple copies of the system image are kept on a number of predefined shared locations. These shared locations are as defined in the file "server_chooser.cmd" [Listing 1]

B. Initiating the (Re)Deployment

The lab computers are configured to boot through network. By default, they boot with the local hard disk which is set as the default boot option in the PXE boot menu. For carrying out the deployment one uses the arrow keys to select the "recovery" boot loader which boots the system with Windows PE preconfigured to get the image from the predefined location and to start restoring to the drive of the computer. This is the only manual intervention required in the whole setup. Once the restore is complete, the system reboots and the automation scripts take over. A single hosting of the system image can be used to deploy up to two computers simultaneously without any significant loss in the rate of transfer of the image file.

When more application software or some new settings are required to be put on all computers, the reference image is again deployed on one of the computers so that it is the reference computer. New applications settings are applied, tested and then the procedure for creating the image is followed. The old image file(s) on the shared location(s) is/are replaced with the new one and all the computers are redeployed with the new image.

Users have no data on the local computers. Profiles and home directories are server based. Hence there is no issue even if a user herself redeploys the computer should she find any irregularity with the working of the computer.

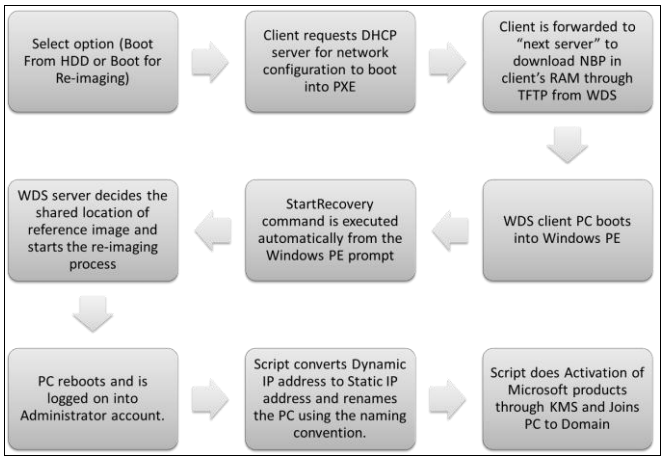

Fig. 3. Process outline for deploying the copies of reference image.

The default boot option received from the PXE is reconfigurable. Hence this fact is exploited to remotely order a redeployment of any computer.

The servers hosting the system image is predefined in the deployment setup so a new server is automatically chosen in round robin fashion so as to load balance the image hosting servers.

V. WIM Based Deployment vs. System Image Based Deployment

The normal setup as supported by Microsoft for WDS is through the use of WAIK to create WinPE environment. From WinPE environment, Imagex tool is used to make WIM image of the reference computer. This mode offers advantages for cases where dissimilar hardware is frequent, or in places where the hardware is changed very often. However, the creation of WIM image is time consuming. This is a serious disadvantage as compared to system image based deployment for similar hardware. The creation of system image is much easier and consumes much lesser time. In a typical scenario, generally, there is a need to update the reference image and redeploy all computers at least once in three-four months. This could be due to a variety of reasons – need to add new software, need to provide OS and application updates, need to add new preference customizations etc. To add to the list, support for having several copies of WIM image at different hosted locations is lacking. Thus, when similar hardware is involved, the choice of system image based deployment is much more justified. Though system image based deployment is primarily suitable for similar hardware configurations, it can be easily adapted to make it suitable for installations with groups of computers having dissimilar hardware, if these groups each have considerable number of similar computers. Each group can then have a separate reference image.

VI. Some Statistics and Methods to Increase Total Deployment Speed

Though no benchmark tests have been performed, we

have collected some statistics. For a system image size of 70 GB, a single target computer is made ready to be used by any Active Directory user in about 20-25 minutes, with 1Gbps LAN. If the image is hosted on a server with SCSI/SAS based disks, the same copy of system image can be used to deploy up to two target computers simultaneously. In Computer Centre at Indian Institute of Technology Kanpur, we have four Windows labs having 161 computers. Two of the labs are in a different building. In this case, one lab of 48 computers is successfully deployed with a system image consisting of Windows 7 Professional 64-bit, and a long list of applications in duration of 4 hours. The applications in the category of Word processing, CAD/CAM, Statistics, Graphical presentation and other miscellaneous software make the image size a little over 70 GB. We have used five SATA disk based locations for hosting the system image. The maximum number of image hosting servers is limited to nine in a single setup.

The use of multiple image hosting servers is a unique concept of this deployment system setup. Increasing the number of hosting locations increases the initial preparation time because the image has to be copied to each hosting location, but speeds up the overall deployment process.

VII. Preparing for Remote Deployment Using Intel vPro

The computer must support Intel vPro technology to carry out the deployment remotely. The computer's BIOS settings have to be configured [4]. After setting up the computer for Intel vPro and AMT, the computer can be fully monitored or controlled in the same way as one would do on physically visiting the computer. The only condition is that the computer should be receiving power and connected to network.

For controlling the computers, Intel provides a software called Manageability Commander Tool. This is for testing and developing software that supports the Intel vPro technology. This software is available for development purpose only and not for commercial use. It contains all Intel vPro controlling features including remote commands to shutdown, reboot, redirect to network boot, redirect local CD/DVD or ISO image, lock remote keyboard & mouse etc. This Software contains two main remote control modules - Manageability Terminal Tool and Remote KVM. The Remote KVM is a Remote GUI tool, which shows a large logo of Real VNC. To get rid of this logo one has to purchase real VNC.

Remote deployment option is more suitable in case of a few computers being reported as malfunctioning. The administrator can remotely start the redeployment process for the computer without physically visiting it.

VIII. Conclusion and Scope for Future Improvements

The procedure to install and restore a computer presented in this paper allows administrators of small and large sites to automate the deployment of Windows 7 and application software. After the initial configuration and customization, the system is very useful in maintaining large installations. In only a few years the investment on time and additional hardware is easily harvested.

Support for including any changes in hardware of the target computers is to be explored. A concept of building a universal image which would work on computers of almost all makes is in the pipeline. Though such a system has been reported to be in use with Windows XP [5], it needs to be worked out and tested for the current setup.

Though possibility of multicast deployment has been explored [6], it has been found to deteriorate the overall network performance. It is advisable to segregate the section of network where the multicast deployment is being carried out. The system is yet to be extended to support ipv6.

Acknowledgment

We acknowledge system administrators, faculty members and students of Indian Institute of Technology Kanpur to help in testing out the deployment setup and give timely feedbacks.

References

[1] Microsoft. (July 2013). On Windows Deployment Services Getting Started Guide. *Microsoft Technet* [Online]. Available: http://technet.microsoft.com/en-us/library/7d837d88-6d8e-420c-b68f-a5b4baeb5248.aspx

[2] Microsoft. (June 2013). On the Windows Automated Installation Kit (AIK) for Windows 7. *Microsoft Download Centre*. [Online]. Available: http://www.microsoft.com/en-us/download/details.aspx?id=5753

[3] Microsoft. (July 2013). On How to Add Windows Device Drivers to a Boot Image. *Microsoft Technet.* [Online]. Available: http://technet.microsoft.com/en-us/library/bb680705.aspx

[4] Intel. (June 2013). On Intel vPro Technology. *Intel Website.* [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/vpro/vpro-technology-general.html

[5] S. G. Lewis and Sara K. Rodgers, "Universal imaging: revolutionizing desktop support," presented at the 33rd annual ACM SIGUCCS fall conference, Monterey, CA, November 6-9, 2005.

[6] Microsoft. (June 2013). On Performing Multicast Deployments. *Microsoft Technet.* [Online]. Available: http://technet.microsoft.com/en-us/library/dd637994(WS.10).aspx

Gopesh Tiwari received B.Tech. degree in electrical engineering from G.B. Pant University, Pantnagar, India, in 1992 and M.Tech. degree in electrical engineering from Institute of Technology, Banaras Hindu University, Varanasi, India, in 1995.

He is currently a senior computer engineer at Indian Institute of Technology Kanpur, Kanpur, India. He was born in 1967 at Allahabad, India. His research interests are in the areas of automated large scale deployment of computers, virtualization, computer networking, remote monitoring of servers and lab computers. He may be contacted on his email: gopesh@iitk.ac.in.

Narendra S. Rathore is appearing for MCA from Lovely Professional University, Jalandhar, India. He is currently working as a technical associate at Indian Institute of Technology Kanpur, Kanpur, India. He was born in 1981 at Kanpur, India. His field of interest is deployment and troubleshooting of Windows operating system and scripting in Visual Basic.